

## Лекція 2. ПРЕДСТАВЛЕННЯ ЗНАНЬ У СШІ

- знання та моделі представлення знань у СШІ
- продукційні моделі представлення знань
- управління пошуком рішень у продукційних системах
- переваги і недоліки продукційних систем
- приклади продукційних систем

### 1. Знання та моделі представлення знань у СШІ

#### 1.1. Дані і знання

При вивченні систем штучного інтелекту традиційно виникає питання що ж таке знання і чим вони відрізняються від звичайних даних, що десятиліттями обробляються ЕОМ. Слід відзначити, що поняття *знання* є центральним поняттям в *ІС*. Можна запропонувати декілька робочих визначень, в рамках яких це стає очевидним.

*Дані* – це окремі факти, що характеризують об'єкти, процеси і явища предметної області, а також їх властивості.

Знання засновані на даних, отриманих емпіричним шляхом. Вони є результатом розумової діяльності людини, направленої на узагальнення її досвіду, отриманого внаслідок практичної діяльності. Дамо декілька визначень *знань*:

1. *Знання* є результат, отриманий пізнанням навколишнього світу і його об'єктів.

2. *Знання* – система думок з принциповою і єдиною організацією, заснована на об'єктивній закономірності.

3. *Знання* – це формалізована інформація, на яку посилаються або яку використовують в процесі логічного виводу (рис. 1).

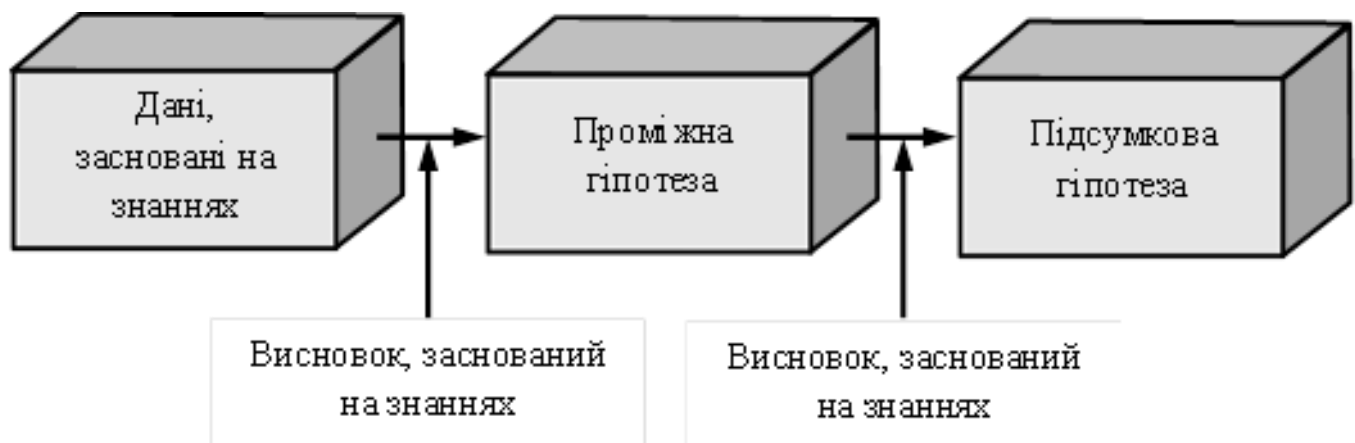


Рис.1. Процес логічного виводу в ІС

**4.** Під **знаннями** розумітимемо сукупність фактів і правил. Поняття правила, що представляє фрагмент *знань*, має вигляд:

*якщо <умова> тоді <дія>*

Наприклад, якщо Х істинно і Y істинно, тоді Z істинно з достовірністю P.

Визначення 1 і 2 є достатньо загальними філософськими визначеннями. У ІС прийнято використовувати визначення 3 для визначення *знань*. Визначення 4 є окремий випадок визначення 3.

Під статичними *знаннями* розумітимемо *знання*, введені в ІС на етапі проектування. Під динамічними *знаннями* (досвідом) розумітимемо *знання*, отримані ІС в процесі функціонування або експлуатації в реальному масштабі часу.

*Знання* можна розділити на факти і правила. Під фактами маються на увазі *знання* типу «А це А», вони характерні для баз даних. Під правилами (продукціями) розуміються *знання* виду «ЯКЩО-ТО». Окрім цих *знань* існують так звані метазнання (*знання про знання*). Створення продукційних систем для представлення *знань* дозволило розділити *знання* і управління в комп'ютерній програмі, забезпечити модульність продукційних правил, тобто відсутність синтаксичної взаємодії між правилами.

**Знання** – це закономірності предметної області (принципи, зв'язки, закони), отримані внаслідок практичної діяльності і професійного досвіду, що дозволяють фахівцям ставити і вирішувати задачі в цій області.

Часто використовується таке визначення *знань*:

**Знання** – це добре структуровані дані, або дані про дані, або метадані.

Для зберігання даних використовуються бази даних (для них характерні великий об'єм і відносно невелика питома вартість інформації), а для зберігання *знань* – бази *знань* (невеликого об'єму, але виключно дорогі інформаційні масиви).

**База знань** – основа будь-якої інтелектуальної системи. Знання можуть бути класифіковані за наступними категоріями:

- Поверхневі – знання про видимі взаємозв'язки між окремими подіями і фактами в предметній області.
- Глибинні – абстракції, аналогії, схеми, що відображають структуру і природу процесів, які протікають в предметній області. Ці знання пояснюють явища і можуть використовуватися для прогнозування поведінки об'єктів.

### **Приклад.**

#### ***Поверхневі знання:***

“Якщо натиснути на кнопку дзвінка, роздасться звук. Якщо болить голова, то потрібно прийняти аспірин”.

#### ***Глибинні знання:***

“Принципова електрична схема дзвінка і проводки. Знання фізіологів і лікарів високої кваліфікації про причини, види головних болів і методи їх лікування”.

Сучасні СШІ працюють в основному з поверхневими знаннями. Це пов’язано з тим, що на даний момент немає універсальних методик, що дозволяють виявляти глибинні структури знань і працювати з ними.

Знання характеризуються послідовністю ***властивостей***, що відрізняють їх від традиційних моделей даних:

- ***Внутрішня інтерпретованість***. При збереженні знань у пам’яті СШІ, поряд із традиційними елементами даних, зберігаються й інформаційні структури; що дозволяють інтерпретувати вміст відповідних комірок пам’яті.
- ***Структурованість***. Знання складаються з окремих інформаційних одиниць, між якими можна встановити класифікаційні стосунки: рід – вид, клас – елемент, тип – підтип, частина – ціле і т.п.
- ***Зв’язність***. Між інформаційними одиницями передбачаються зв’язки різного типу: причина – наслідок, одночасно, бути поруч і ін. Дані зв’язки визначають семантику і прагматику предметної області.
- ***Семантична метрика***. На множини інформаційних одиниць, збережених у пам’яті, вводяться деякі шкали, що дозволяють оцінити їхню семантичну близькість. Це дозволяє знаходити в інформаційній базі знання, близькі до вже знайденого.
- ***Активність***. За допомогою даної властивості підкреслюється принципова відмінність знань від даних. Виконання тих чи інших дій у СШІ ініціюється станом бази знань; при цьому передбачається, що поява нових фактів і зв’язків може активізувати систему.

Знання в базі знань подаються в певному вигляді, тобто в певних інформаційних одиницях знань і зв’язках між ними. Форма подання знань істотно впливає на

характеристики і властивості інформаційних систем, тому це є однією з найважливіших проблем, характерних для систем, орієнтованих на знання.

Оскільки логічні висновки і дії над знаннями в інформаційних інтелектуальних системах проводяться програмовано, то знання не можуть бути подані безпосередньо в звичайному вигляді, тобто в якому вони використовуються людьми. У зв'язку з цим розробляються формальні моделі подання знань.

При розробленні специфічних моделей подання знань намагаються дотримуватися таких вимог:

1. Подання знань має бути однорідним (одноманітним).
2. Подання знань має бути зрозумілим експертам і користувачам системи.

Мова, що використовується для розроблення систем на основі цих моделей, називається мовою подання знань. Відомі такі **мови подання знань**:

- логічна мова подання знань, в основу якої покладено числення предикатів першого порядку;
- продукційна мова, основними одиницями якої є продукції (правила);
- фреймова мова, у котрій для подання і маніпулювання знаннями використовується фреймова модель подання знань.

## ***1.2. Моделі представлення знань у СШІ***

Центральним питанням побудови систем, заснованих на знаннях, є вибір форми представлення знань. Представлення знань – це спосіб формального вираження знань про предметну область у комп'ютерно-інтерпретованій формі. Відповідні формалізми, що забезпечують зазначене представлення, називають **моделями представлення знань**.

Моделі представлення знань можна умовно розділити на **декларативні** і **процедурні**.

**У декларативних моделях** знання представляються у вигляді описів об'єктів і відносин між об'єктами без вказівки в явному вигляді, як ці знання обробляти. Такі моделі припускають відділення описів (декларацій) інформаційних структур від механізму вивід в, що оперує цими структурами.

**У процедурних моделях** значення представляються алгоритмами (процедурами), що містять необхідний опис інформаційних елементів і одночасно визначають способи їх обробки.

Конкретні моделі, що застосовуються на практиці, зазвичай є комбінацією декларативних і процедурних представлень. Найбільш розповсюдженими є наступні моделі представлення знань:

- логічні моделі;
- продукційні моделі;
- мережні моделі;
- фреймові моделі.

**Логічні моделі** реалізуються засобами логіки предикатів. У цьому випадку знання про предметну область представляються у вигляді сукупності логічних формул. Тотожні перетворення формул дозволяють одержувати нові знання. Перевагами логічних моделей представлення знань є наявність чіткого синтаксису і широко прийнятої формальної семантики, а також теоретично обґрунтованих процедур автоматичного виводу. Основним недоліком даних моделей є неможливість одержання висновків в областях, де вимагаються правдоподібні висновки, коли результат виходить з визначеною оцінкою впевненості в його істинності. Крім цього, такі моделі характеризуються монотонним характером виводу, тобто в базу знань додаються тільки істинні твердження, що виключає можливість протиріч. На практиці часто зустрічаються немонотонні міркування, що важко реалізувати в рамках логічної моделі.

Проте, логічні моделі виступають у якості теоретичної основи опису самої системи представлення знань і поступово розширюють свої можливості. Тому надалі цим моделям приділяється значна увага.

У **продукційних моделях** знання представляються набором правил виду “Якщо А, то В”, де умова правила А є твердженням про вміст бази фактів, а наслідок В говорить про те, що треба робити, коли дане продукційне правило активізоване.

Продукційні моделі представлення знань завдяки природній модульності правил, наочності і простоті їхнього створення широко застосовуються в інтелектуальних системах.

**Мережні моделі** представлення знань розрізняються між собою типами зв'язків (відносин), що використовуються. Якщо в мережі використовуються ієрархічні зв'язки (клас-підклас, рід-вид і т.п.), то мережа називається класифікованою. Якщо зв'язки між інформаційними елементами представляються функціональними відносинами, що

дозволяють обчислювати значення одних інформаційних елементів за значеннями інших, то мережі називають функціональними (обчислювальними). Якщо в мережі допускаються зв'язки різного типу, то її називають семантичною мережею.

*Семантичні мережі* є винятковим випадком мережних моделей представлення знань. Формально мережні моделі задаються у вигляді:

$$H = \langle I, C_1, C_2, \dots, C_n, Q \rangle,$$

де  $I$  – множина інформаційних елементів, що зберігаються у вузлах мережі;  $C_1, C_2, \dots, C_n$  – типи зв'язків між інформаційними елементами;  $Q$  – відображення, що встановлює відповідність між множиною типів зв'язків і множиною інформаційних елементів мережі.

Семантична мережа є спрямованим графом, у якому вершинам відповідають об'єкти (сутності) предметної області, а дугам – відносини, у яких знаходяться ці об'єкти. Вивід у семантичних мережах може виконуватися на основі алгоритмів співставлення, шляхом виділення підграфів з визначеними властивостями.

До переваг семантичних мереж відносять: велику виразну здатність, наочність графічного представлення, схожість структури мережі до семантичної структури фраз природної мови. Недоліком представлення знань у вигляді семантичних мереж є відсутність єдиної термінології. Дана модель представлення знань знаходить різне втілення в різних дослідників.

**Фреймові моделі** представлення знань використовують теорію організації пам'яті, розуміння і навчання. Фрейм (від англ. frame – рамка, каркас, кістяк) – структура даних, призначена для представлення стереотипних ситуацій. Фрейм складається зі слотів (slot – гніздо, щілінка, паз). Значенням слота можуть бути числа, вираження, тексти, програми, посилання на інші фрейми. Сукупності фреймів утворюють ієрархічні структури, побудовані на родовидових ознаках, що дозволяє успадковувати значення слотів. Така властивість фреймів забезпечує ощадливе розміщення бази знань у пам'яті. Крім цього, значення слотів можуть обчислюватися за допомогою різних процедур, тобто фрейми комбінують у собі декларативні і процедурні представлення знань. Фреймові моделі можна розуміти як мережні моделі уявлення знань, коли фрагмент мережі представляється фреймом з відповідними слотами і значеннями. З фреймовими моделями зв'язані моделі представлення знань на основі сценаріїв і об'єктів.

## 2. Продукційні моделі представлення знань

Продукційні системи вперше винайдені Постом в 1941г. Продукція в системі Поста має наступну схему

$$\frac{t_1, t_2 \dots t_n}{t}, \quad (1)$$

де  $t_1, t_2 \dots, t_n$  називаються посилками, а  $t$  висновком продукції.

Застосування схеми (1) ґрунтується на підстановці ланцюжків знаків замість всіх змінних, причому замість входжень однієї і тієї ж змінної підставляється один і той же ланцюжок.

Продукційна система – це модель обчислень, яка грає особливо важливу роль для створення алгоритмів пошуку і для моделювання вирішення завдань людиною. Продукційна система забезпечує управління процесом вирішення завдання за зразком і складається з набору продукційних правил, робочої пам'яті і циклу управління "розпізнавання - дія". Продукційну систему можна визначити на основі наступних категорій.

**Набір продукційних правил.** Їх часто просто називають продукціями. Продукція – це пара "умова - дія", яка визначає одну порцію знань, необхідних для вирішення задачі. Умовна частина правила – це зразок (шаблон), який визначає, коли це правило може бути застосовано для вирішення якогось етапу завдання. Частина дії визначає відповідний крок у вирішенні завдання.

**Робоча пам'ять** містить опис поточного стану світу в процесі міркувань. Це опис є зразком, який зіставляється з умовною частиною продукції з метою вибору відповідних дій при вирішенні задачі. Якщо умова деякого правила відповідає вмісту робочої пам'яті, то може виконуватися дія, пов'язана з цією умовою. Дії продукційних правил призначені для зміни змісту робочої пам'яті.

**Цикл "розпізнавання - дія".** Керуюча структура виробничої системи проста: робоча пам'ять ініціалізується початковим описом завдання. Поточний стан вирішення завдання представляється набором зразків в робочій пам'яті. Ці зразки зіставляються з умовами продукційних правил; що породжує підмножину правил виводу, зване конфліктною множиною. Умови цих правил узгоджені із зразками в робочій пам'яті. Продукції, що містяться в конфліктній множині, називають допустимими. Вибирається і активізується одна з продукцій конфліктної множини (дозвіл конфлікту). Активізація правила означає виконання його дії. При цьому змінюється зміст робочої пам'яті. Після того, як вибране

правило спрацювало, цикл управління повторюється для модифікованої робочої пам'яті. Процес закінчується, якщо вміст робочої пам'яті не відповідає жодним умовам.

У процесі вирішення конфліктів вибирається для виконання правило з конфліктної множини. Стратегії вирішення конфліктів можуть бути досить простими, наприклад, вибір першого правила, умова якого відповідає стану світу. Можна для такого вибору використовувати складну евристику. Слід підкреслити, що продукційна система допускає використання додаткових евристик для управління алгоритмом пошуку.

Чиста продукційна модель не має ніякого механізму виходу з тупикових станів у процесі пошуку; вона просто продовжує працювати до того часу, поки не будуть вичерпані всі допустимі продукції. Багато практичних реалізацій продукційних систем містять механізми повернення в попередній стан робочої пам'яті.

Простий приклад роботи продукційної системи представлений на рис. 2.

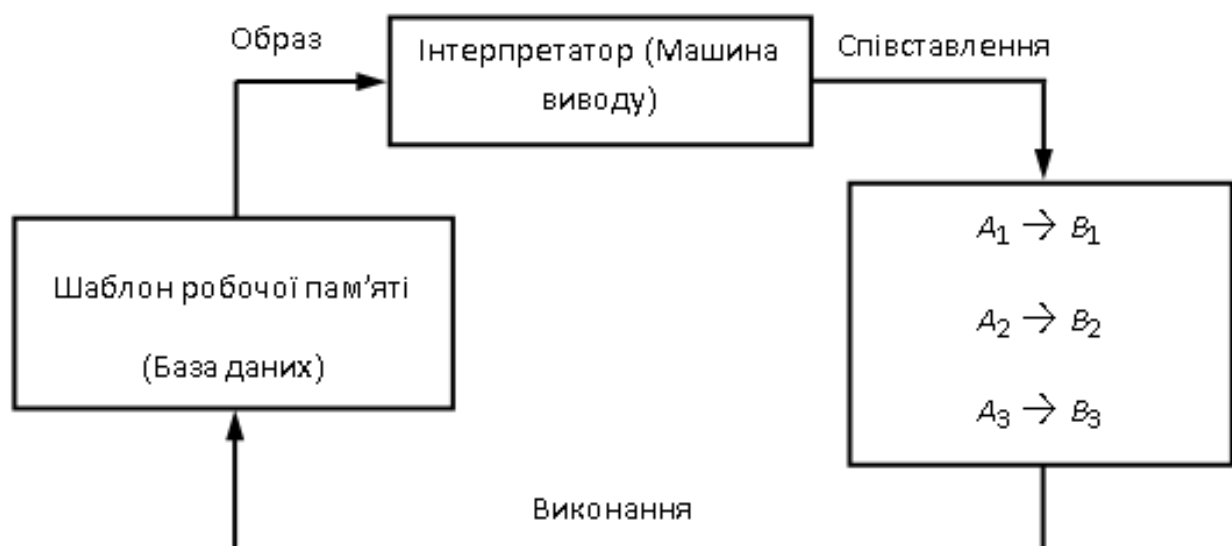


Рис. 2. Продукційна система. Цикл виконується до того часу, поки зразок робочої пам'яті не буде більш відповідати жодній з умов продукційних правил

Дана продукційна система сортує рядок, що складається з символів *a*, *b* і *c*. У цьому прикладі продукція є допустимою, якщо її умова відповідає частині рядка в робочій пам'яті. При виконанні правила підстрічка, яка відповідала його умові, замінюється рядком з правої частини правила. Продукційна система – це загальна модель обчислень, яка може бути запрограмована для виконання будь-якої задачі на комп'ютері. Однак справжнє її призначення – це реалізація інтелектуальних систем.



Набір продукцій:

1.  $ba \rightarrow ab$
2.  $ca \rightarrow ac$
3.  $cb \rightarrow bc$

Табл. 1. Робота простої виробничої системи

Ітерація	Робоча пам'ять	Конфліктна множина	Застосування правила
0	<i>cbaca</i>	1,2,3	1
1	<i>cabca</i>	2	2
2	<i>acbca</i>	2,3	2
3	<i>acbac</i>	1,3	1
4	<i>acabc</i>	2	2
5	<i>aacbc</i>	3	3
6	<i>aabcc</i>	$\emptyset$	Останов

Продукційні системи забезпечують модель представлення людського досвіду у формі правил і дозволяють розробляти алгоритми пошуку за зразком – центральний елемент заснованих на правилах експертних систем. В експертних системах продукційні моделі не обов'язково є точною реалізацією людського підходу до вирішення завдання. Однак існують такі аспекти продукційних систем, які роблять їх корисними в якості потенційної моделі вирішення задачі людиною (модульність правил, поділ знання та управління, поділ робочої пам'яті і навичок у вирішенні задач). Це служить ідеальним інструментом для розробки експертних систем. Як інші класичні продукційні системи відзначимо нормальні алгоритми Маркова і машину Тюрінга.

Розвитком моделі на основі правил є модель "дошки оголошень". Ця модель реалізована в системі розпізнавання розмовної мови HEARSAY - 2. Основний принцип організації моделі дошки оголошень полягає в розбитті продукцій по рівнях ієрархії. При цьому висновки продукцій на нижніх рівнях використовуються як вхідні умови для продукцій більш високого рівня. На нижньому рівні моделі дошки оголошень представлені факти, на верхньому – результуючий висновок.

Ієрархічне розбиття множини продукцій дозволяє ефективніше організувати їх виконання, істотно скоротивши витрати на перебір множини продукцій при перевірці умов їх спрацьовування, що визначає додатковий інтерес до продукційних систем.

### 3. Управління пошуком рішень у продукційних системах

Моделі продукційних систем надають додаткові можливості по додаванню евристичного управління до алгоритму пошуку. Ці додаткові зручності включають вибір стратегії (на основі даних або від мети), вибір самої структури правил і стратегій для вирішення конфліктів.

Вибір методу пошуку рішення залежить від наступних параметрів:

- розмір ПО;
- змінність області;
- повнота моделі, яка описує область;
- визначеність даних про розв'язувану задачу;
- кількість необхідних рішень;
- обмеження на результат і спосіб його одержання.

Залежно від цих параметрів використовуються такі методи:

1. Пошук рішення в одному просторі станів (малі статичні області, точні та повні дані);
2. Пошук в ієрархічних просторах (робота у великих статичних областях);
3. Пошук при неточних і неповних даних;
4. Пошук в динамічних проблемних областях;
5. Пошук з використанням декількох моделей розгляду області (з різних точок зору).

У задачі управління пошуком рішень у продукційних системах можливі два шляхи вирішення: централізований і децентралізований. При централізованому виконанні продукцій розв'язок приймається спеціальною системою управління, а при децентралізованому визначається ситуацією, що склалася в цей момент. Якщо порядок виконання продукцій довільний, тоді розв'язок визначається поточною ситуацією або системою управління; якщо порядок важливий, тоді в продукціях повинна бути інформація про вимоги до цього продукту. Якщо в післяумовах продукцій вказується ім'я продукції, яка повинна виконуватися після даної, система продукцій перетворюється в звичайну програму, тобто реалізує деякий алгоритм.

**Основні стратегії.** Вирішення конфліктів – важлива проблема, зв'язана з управлінням порядку застосування правил, що утворюють конфліктну множину. Порядок активізації правил конфліктної множини визначається вибраною стратегією вирішення

конфліктів. Зазвичай конфліктна множина правил представляється у вигляді впорядкованого списку. При цьому конфліктні правила дописуються в кінець цього списку. Прості стратегії вирішення конфлікту ґрунтуються на тому, що вибирається або перше, або останнє правило, яке входить у список. Вибір першого правила відповідає пошуку в ширину, а вибір останнього правила (тобто тільки що доданого) – пошуку в глибину. У багатьох продукційних системах найбільш часто використовують другий спосіб.

Принципами, що використовуються для розв'язування конфліктів, є:

- принцип «стопки книг»;
- принцип найбільш довгої умови;
- принцип метапродукції;
- принцип пріоритетного вибору;
- принцип «класної дошки»;
- принцип дошки оголошень;
- управління за іменем;
- $\alpha$ - $\beta$ -алгоритм;
- принцип розбиття на підзадачі.

**1. Принцип «стопки книг».** Заснований на ідеї, що продукція, яка найбільш часто використовується, є найбільш корисною. На самому верху «стопки» знаходиться продукція, яка використовувалася частіше за всіх. При актуалізації деякого фронту готових продукцій для поповнення вибирається та продукція, у якій частота використання є максимальною. Управління за принципом стопки зручно застосовувати, якщо продукції відносно незалежні, наприклад, коли кожна з них є правилом виду «ситуація А  $\rightarrow$  дія В». Тому вони використовуються в системах планування для роботів.

**2. Принцип найбільш довгої умови.** Полягає у виборі з фронту готових продукцій тієї, у якій стала істинною найбільш довга умова здійсненності ядра. Цей принцип спирається на міркування здорового глузду, що приватні правила, які стосуються вузького класу ситуації, важливіші загальних правил, оскільки враховується більше інформації про ситуацію. Труднощі використання даного принципу полягають в тому, що необхідно заздалегідь впорядкувати умови по входженню одна в одну за відношенням «приватне - загальне». Управління за цим принципом доцільно в тих випадках, коли знання і самі продукції добре структуровані прив'язкою до типових ситуацій, на яких задано відношення типу «приватне - загальне».

**3. Принцип метапродукції.** Заснований на ідеї введення в систему продукцій спеціальних метапродукцій, завданням яких є організація управління в системі продукції при можливості неоднозначного вибору з фронту готових продукцій (використовується в MYCIN при діагностиці інфекційних захворювань).

**4. Принцип пріоритетного вибору.** Пов'язаний з введенням статичних або динамічних пріоритетів на продукції. Статичні пріоритети можуть формувати апріорі на підставі відомостей про важливість продукційних правил в даній предметній області. Це зазвичай інформація, що отримується від експертів. Динамічні пріоритети виробляються в процесі функціонування системи продукцій і можуть відображати, наприклад, такий параметр, як час знаходження продукції у фронті готових продукцій.

Існує два типи виконання систем продукцій: прямий і зворотній. В першому випадку пошук здійснюється від лівих частин продукцій, тобто. перевіряються умови А і актуалізуються ті продукції, для яких А має місце; у другому – від початково заданих В, за яким визначаються необхідні для В значення А.

**5. Принцип класної дошки.** Заснований на ідеї спускових функцій. При реалізації принципу «класної дошки» в СШІ виділяється робоче поле пам'яті – аналог класної дошки, на якій пишуть оголошення і при необхідності стирають ганчіркою. На цій «дошці» процеси, які паралельно виконуються, знаходять інформацію, яка ініціює їх запуск, на неї ж вони і виносять інформацію про свою роботу, що цікавлять інші процеси.

Як правило, на «класній дошці» виділені спеціальні поля для формування умов застосування ядер продукцій, різні для різних сфер застосування продукцій, спеціальні поля для запису результатів спрацювання продукцій і для запису післяумов, якщо вони адресовані іншим продукціям. З принципом «класної дошки» може комбінуватися принцип управління з допомогою метапродукцій, оскільки він вимагає перевірки деяких умов, які фіксуються у робочому полі пам'яті, а також інші принципи управління.

**6. Принцип дошки оголошень.** На дошці оголошень розташовані в ієрархічному порядку гіпотези. На верхньому рівні – правила, на нижньому – факти, на проміжних – проміжні гіпотези. Навколо дошки оголошень розташовані згруповані правила, звані джерелами знань. Угрупування за рівнями дозволяє скоротити час пошуку фактів (обмежуючись певною групою на кожному рівні) порівняно з усією базою даних.

Зводиться до зменшення простору шляхів за допомогою видалення гілок, що не перспективні для пошуку успішного вирішення. Тому переглядають тільки ті вершини, в яких можна потрапити в результаті наступного кроку, після чого неперспективний виключаються. Наприклад, якщо колір предмета «не червоний», то його безглуздо шукати серед предметів червоного кольору. Він знаходить широке застосування в різних іграх ( шахах і т. д.).

**7. Принцип управління за іменами.** Засноване на задаванні імен продукцій, які входять в деяку систему, деякої формальної граматики або іншої процедури, що забезпечує звуження фронту готових продукцій і вибір з неї чергової продукції для виконання.

Наприклад, нехай система продукцій представлена чотирма простими продукціями, що складаються лише з ядерних частин: (а)  $A \rightarrow B$ ; (б)  $B \& D \rightarrow A$ ; (в)  $A \vee B \rightarrow D$ ; (г)  $D \rightarrow C$ . В такому виді система продукцій явно недетермінована. Якщо виконується А, то фронт готових продукцій включає продукції з іменами (а) і (в), а якщо виконуються В і D, тоді три останні продукції.

Для усунення такої недетермінованості можна ввести деяку граматику для імен продукцій: (а) $\rightarrow$ (б); (в) $\rightarrow$ (б); (а) $\rightarrow$ (г). Тоді якщо в деякий момент була виконана продукція з іменем (б), то нові продукції виконуватись не будуть. Якщо ж в деякий момент виконалась продукція з іменем (а), тоді після неї можуть виконуватись продукції з іменами (б) і (г). Але ліві частини ядер в цих продукціях такі, що неоднозначність виникає лише у випадку, коли В і D одночасно виконано. З допомогою граматики для імен продукцій можна задати однозначний алгоритмічний процес.

**8.  $\alpha$ - $\beta$ -алгоритм** (евристичний з оцінкою  $f(v) = g(v) + h(v)$ ). С допомогою цього алгоритму початкова задача зводиться до зменшення простору станів шляхом видалення в ньому гілок, неперспективних для пошуку успішного рішення, тобто переглядаються тільки ті вершини, в які можна потрапити в результаті наступного кроку, після чого неперспективні напрямлення виключаються. Наприклад, в БЗ продукційної системи, заповненої знаннями про тварин, не варто шукати тварин, які не відносяться до ссавців, в напрямленні, яке бере початок від вершини, що визначає ссавців. Дана стратегія є деяким компромісом між пошуком в ширину і пошуком в глибину. Для її успішної реалізації необхідно володіти додатковими евристичними знаннями, які використовуються при виборі перспективних напрямлень.

**9. Розбиття на підзадачі.** Декомпозиція дає позитивний ефект тільки для добре структурованих областей знань, так як застосування цієї стратегії засновано на правильному розумінні сутності задачі і можливості її представлення у виді системи ієрархічно зв'язаних цілей – підцілей, причому розбиття на підзадачі необхідно виконувати оптимальним способом.

#### **4. Переваги і недоліки продукційних систем**

Основні переваги, завдяки яким продукційні правила отримали широке розповсюдження, полягають в наступному:

- 1) Продукційні правила легкі для сприйняття людиною.
- 2) Окремі продукційні правила можуть бути незалежно добавлені в БЗ, виключені або змінені, при цьому не вимагається перепрограмування всієї системи. Як наслідок цього, представлення великих об'ємів знань не викликає затруднень.
- 3) З допомогою продукційних правил виражаються як декларативні, так і процедурні знання.

До недоліків систем продукцій можна віднести:

- відмінність від структур знань, властивих людині;
- неясність взаємовідношень правил;
- складність оцінки цілісного образу знань;
- низька ефективність обробки знань.

При розробці невеликих систем (десятки правил) проявляються в основному позитивні сторони систем продукцій, однак при збільшенні об'єму знань більш відчутними стають слабкі сторони.

Подальше вдосконалення архітектури, основане на використанні правил, полягає в застосуванні «крупно-зернистих» правил продукції, які представляють «джерела знань», і які працюють під управлінням гнучкого планувальника. Кожне із правил складається із вірця виклику, елемента, «термінового коду» і тіла. «Терміновий код» виконується завжди, коли відбувається співпадання з вірцем виходу, після чого правило переходить під управління планувальника; тіло виконується за командою планувальника.

## 5. Приклади продукційних систем

### Приклад 1. "8 - головоломка"

Простір пошуку для завдання "8 - головоломка", є досить складним і цікавим. У той же час воно настільки мало, що не викликає особливих труднощів у розгляді. "8 - головоломка" часто використовується для вивчення різних стратегій пошуку. Тут ми розглянемо продукційну систему.

Будемо говорити про "переміщенні порожньої клітки". Допустимі ходи визначені продукціями. Природно, якщо порожня клітка знаходиться в центрі, допустимі всі чотири ходи. Якщо порожня клітка знаходиться в одному з кутів, можливі тільки два ходи. Якщо початковий і цільовий стани для "8 - головоломки" визначені, то можна створити продукційну систему, що проглядає простір пошуку завдання.

У реалізації вирішення цього завдання кожену конфігурацію на ігровій дошці можна представити за допомогою предиката стану з дев'ятьма параметрами (для дев'яти можливих положень восьми фішок і порожньої клітини). З іншого боку, для опису стану ігрової дошки можна використовувати масиви або списки.

Оскільки шлях до вирішення може знаходитися дуже глибоко, і якщо його не направляти, пошук буде обмежений граничною глибиною. Простий прийом для реалізації граничної глибини пошуку – стежити за довжиною поточного шляху, і в разі перевищення граничної довжини відстежувати шлях у зворотному напрямку, тобто запускати механізм пошуку з поверненням. На рис. 3 гранична глибина пошуку дорівнює 5. Зауважимо, що число можливих станів робочої пам'яті зростає експоненціально зі збільшенням глибини пошуку.

2	8	3
1	6	4
7		5

а)

1	2	3
8		4
7	6	5

б)

Рис. 3. Початковий (а) і цільовий (б) стани продукційної системи

### продукційна множина

Умова

цільовий стан в робочій пам'яті →  
 порожня клітинка не біля лівої межі →

Дія

останов  
 перемістити порожню комірку вліво

порожня клітинка не біля верхньої межі →  
 порожня клітинка не біля правої межі →  
 порожня клітинка не біля нижньої межі →

перемістити порожню комірку вгору  
 перемістити порожню комірку вправо  
 перемістити порожню комірку вниз

Робоча пам'ять містить поточний і цільовий стани ігрової дошки.

### Режим управління

1. Випробувати кожне правило за порядком.
2. Не допускати циклів.
3. Завершити роботу при знаходженні мети.

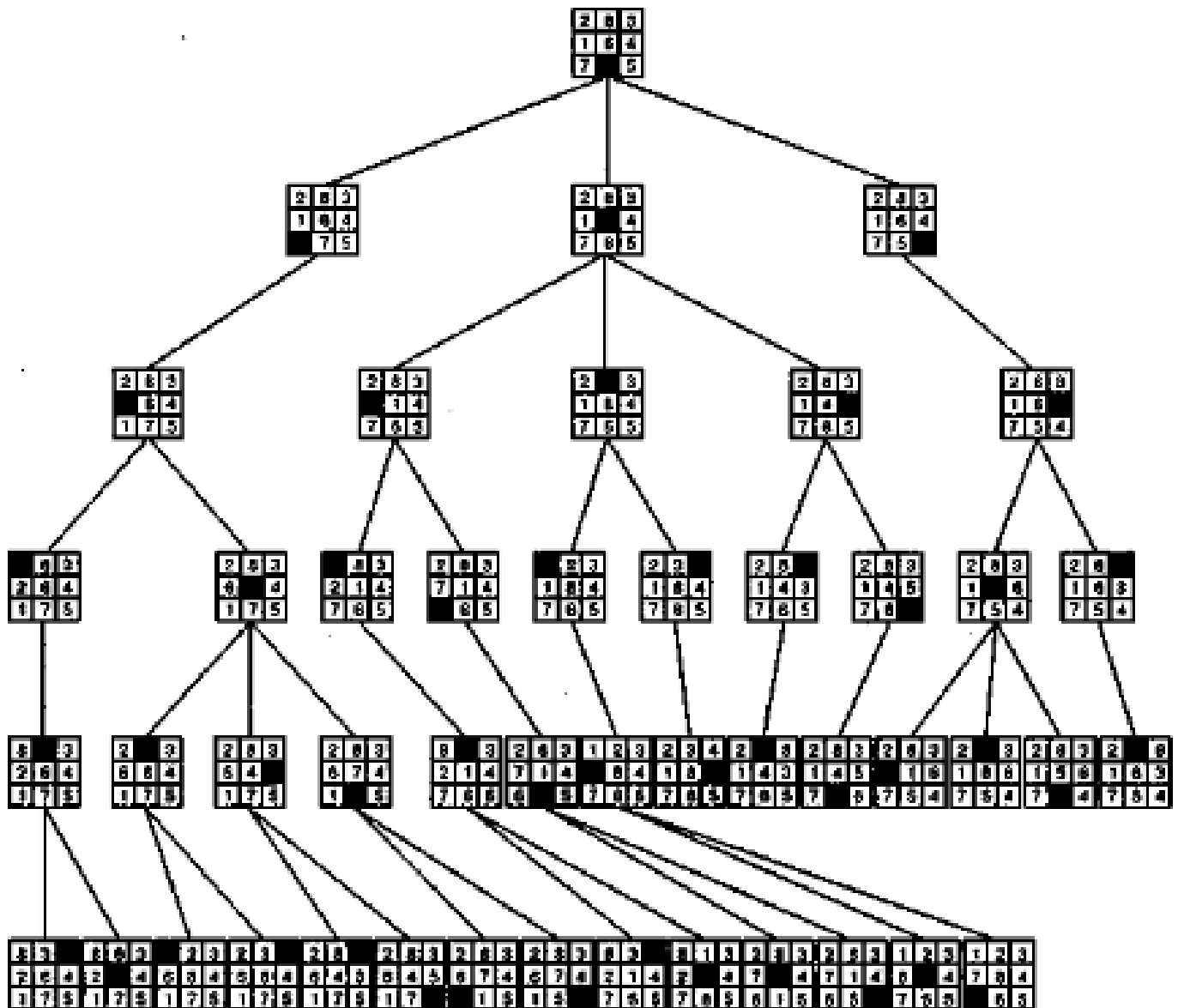


Рис. 4. Рішення завдання "8-головоломки" за допомогою продукційної системи, в якій пошук обмежений глибиною 5



**Приклад 2. Завдання ходу конем**

У шахах кінь може переміщатися на два поля по горизонталі або вертикалі і на одне поле в перпендикулярному напрямку. При цьому він не повинен вийти за межі шахової дошки. Таким чином, у загальному випадку існує не більше восьми можливих ходів. Традиційна формулювання завдання полягає в тому, щоб знайти таку послідовність ходів конем, при якій він стає на кожную клітину тільки один раз. У даному прикладі розглядається спрощена версія завдання ходу конем. Необхідно знайти таку послідовність ходів, при якій кінь побував би в кожному полі зменшеною шахівниці ( $3 \times 3$ ) тільки один раз (рис. 5).

1	2	3
4	5	6
7	8	9

*Рис. 5. Графічне представлення шахівниці  $3 \times 3$*

Це завдання, може бути вирішена за допомогою продукційних систем. У цьому випадку кожен хід можна представити як правило, передумови якого описують розташування коня в конкретній клітці, а дія переміщує коня в іншу клітку. Всі можливі ходи коня описуються за допомогою шістнадцяти продукційних правил.

Робоча пам'ять містить і поточний і цільовий стани дошки. У режимі керування правила застосовуються до того часу, поки поточний стан не зрівняється з цільовим. Тоді процес зупиняється. За простою схемою вирішення конфліктів запускається перше ж правило, яке не викликало зациклення пошуку. Пошук може привести до тупикових станів, з яких кожне можливе переміщення призводить у вже відвіданий стан і, отже, викликає зациклення. Тому режим управління повинен забезпечити повернення. Дії цієї виробничої системи при визначенні існування шляху з поля 1 в поле 2 представлені в табл. 1.

Продукційні правила – це факти переміщень move, перший параметр яких визначає умову (на дошці має бути достатньо місця, щоб зробити хід), а другий – дію (поле, в яке кінь може перейти). Цикл "розпізнавання - дія" реалізується за допомогою рекурсивного предикату шляху path. Робоча пам'ять містить поточний і бажаний цільовий стан. Її можна представити параметрами предиката шляху path. На даній ітерації конфліктна множина – це всі вирази переміщень, які уніфікуються з метою move (X, Z). Ця програма використовує просту стратегію вирішення конфліктів, що складається у виборі і активізації першого предиката переміщення в базі знань, який не веде до повторного

стану. Контролер також здійснює повернення з тупикових станів. Ця версія визначення предиката path для продукційної системи показана на рис. 6.

№ Правила	Умова	Дія
1	Кінь в полі 1	Хід конем в поле 8
2	Кінь в полі 1	Хід конем в поле 6
3	Кінь в полі 2	Хід конем в поле 9
4	Кінь в полі 2	Хід конем в поле 7
5	Кінь в полі 3	Хід конем в поле 4
6	Кінь в полі 3	Хід конем в поле 8
7	Кінь в полі 4	Хід конем в поле 9
8	Кінь в полі 4	Хід конем в поле 3
9	Кінь в полі 6	Хід конем в поле 1
10	Кінь в полі 6	Хід конем в поле 7
11	Кінь в полі 7	Хід конем в поле 2
12	Кінь в полі 7	Хід конем в поле 6
13	Кінь в полі 8	Хід конем в поле 3
14	Кінь в полі 8	Хід конем в поле 1
15	Кінь в полі 9	Хід конем в поле 2
16	Кінь в полі 9	Хід конем в поле 4

Таблиця 2

Продукційна система для вирішення задачі ходу конем на поле 3х3

№ ітерації	Робоча пам'ять		Конфліктна множина (№ правила)	Активізація правила
	Поточне поле	Цільове поле		
0	1	2	1, 2	1
1	8	2	13, 14	13
2	3	2	5, 6	5
3	4	2	7, 8	7
4	9	2	15, 16	15
5	2	2		Вихід

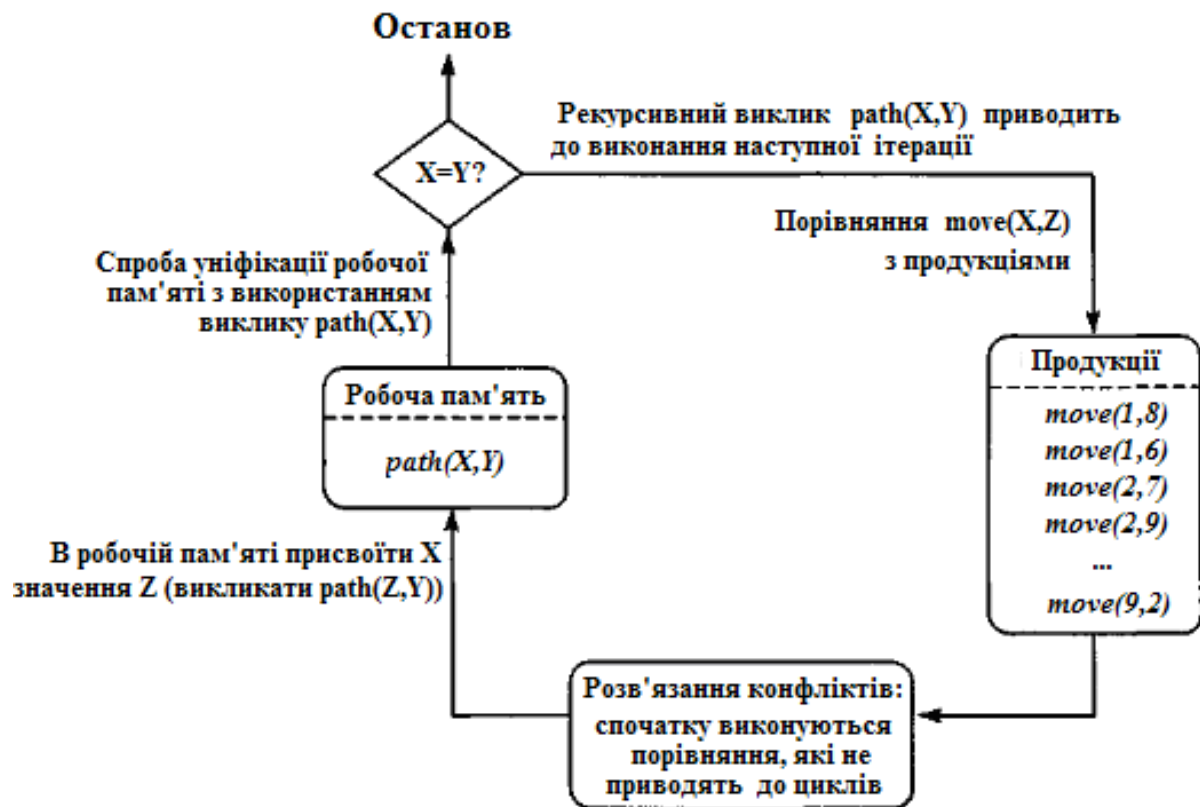


Рис. 6. Рекурсивний алгоритм обчислення шляху в продукційній системі