ПРИКЛАДНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ОБРОБКИ ЕКОНОМІЧНИХ ДАНИХ

Самостійна робота за Змістовним модулем 3: *Технологія проектування експертних систем на основі продукційної моделі*

1. Продукційна модель експертних систем

Як було зазначено в попередньому розділі, експертними системами найбільш поширеного типу є системи, засновані на правилах. Правила, що організовані у вигляді IF-THEN структур, називаються продукційними правилами. Продукційні правила, разом з інтерпретатором, який управляє їх активізацією в залежності від наявних фактів, складають продукційну модель представлення та використання знань в експертних системах. Такі системи носять назву продукційних.

У продукційних системах знання представлені у формі множини правил, на основі яких формуються висновки, що повинні бути зроблені (або не зроблені) в різних ситуаціях. Висновки робляться на основі методів прямого або зворотного логічного висновку. Залежно від методу логічного висновку розрізняють два види продукційних систем: системи з прямим логічним висновком та системи із зворотним логічним висновком.

Загальна стратегія вирішення задач полягає в розбитті їх на фрагменти, які можна легше довести. При цьому системи з прямим логічним висновком знаходяться під управлінням фактів. Вони починають свою роботу з відомих початкових фактів і продовжують, використовуючи правила для створення висновків або виконання певних дій. Системи із зворотним логічним висновком керуються гіпотезами. Вони починають свою роботу з гіпотези, або мети, яку користувач намагається довести і продовжують, відшукуючи правила, які дозволять довести правдивість гіпотези.

Широке застосування систем, заснованих на продукційних правилах, обумовлене наявністю в них наступних особливостей:

1. Модульна організація. Завдяки модульній організації спрощується представлення знань та розширення експертної системи, нарощуючи її можливості крок за кроком.

2. Наявність засобів пояснення. Продукційні експертні системи за допомогою правил дозволяють легко створювати засоби пояснення. Засіб пояснення відстежує послідовність активованих правил і, на цій основі, дає можливість відновити хід міркувань, які привели до певного висновку.

3. Наявність аналогії з пізнавальним процесом людини. Згідно з результатами, одержаними Ньюеллом і Саймоном, правила є природним способом моделювання процесу рішення задач людиною. Тому в процесі виявлення експертних знань не виникають зайві складнощі в поясненні

експертам структури представлення знань, оскільки застосовується просте їх представлення у вигляді правил IF-THEN.

Загальну структуру експертної системи, заснованої на правилах, можна представити у вигляді схеми, зображеній на рис. 1.

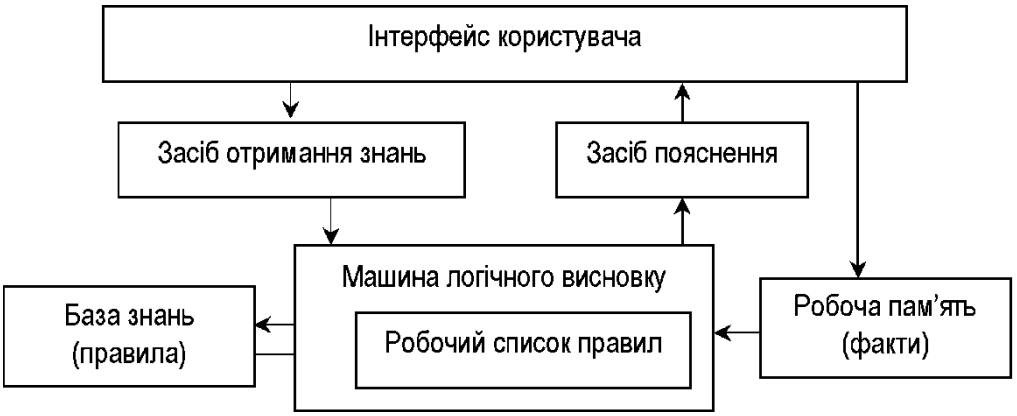

Рис. 1. Загальна структура експертної системи, заснованої на правилах

Наведена структура наглядно ілюструє основні аспекти продук-ційної моделі експертних систем, тому розглянемо детально сутність та призначення її компонентів.

Інтерфейс користувача. Інтерфейс користувача - це механізм, за допомогою якого відбувається спілкування користувача з експертною системою. Залежно від призначення системи інтерфейс користувача може використовувати простий текстовий дисплей або складний растровий дисплей із високою роздільною здатністю. Дисплеї з високою роздільною здатністю, зазвичай, застосовуються для задач моделювання, які вирішує експертна система.

Засіб отримання знань. Засіб отримання знань являє собою автоматизований спосіб, який дозволяє користувачу вводити знання в систему, не застосовуючи явного кодування знань за допомогою інженера по знанням. Цей інструментальний засіб у деяких експертних системах здатний навчатися, здійснюючи автоматичне формування правил на підставі прикладів. Для формування правил у машинному навчанні застосовуються такі методи й алгоритми, як штучні генетичні алгоритми, нейронні мережі.

База знань. Системи, що відповідають продукційній моделі, зберігають знання, необхідні для вирішення задач, у певній проблемній області в базі знань. Базу знань експертної системи, в якій знання закодовані у формі правил, називають продукційною пам'яттю. У ній правила виражені у форматі продукційного псевдокоду IF-THEN.

Кожне правило позначається ім'ям. Після нього починається IF-частина правила. Ця частина продукційного правила розташована між ключовими словами IF і THEN та носить назву антецедента або лівої частини (LHS - left-hand-side) правила. На практиці застосовуються також назви "умовний елемент" та "шаблон".

За частиною IF починається частина THEN правила. Вона містить висновки або список дій, які повинні бути виконані згідно з правилом. Ця частина правила називається консеквентом або правою частиною (RHS - Right-Hand Side). До складу дій консеквентів правил, зазвичай, входять додавання або видалення фактів із робочої пам'яті, або формування результатів. Формат опису цих дій залежить від синтаксису мови експертної системи.

Машина логічного висновку. Машина логічного висновку є програмним компонентом, який визначає антецеденти правил (якщо такі існують), що виконуються згідно з фактами. Для цього машина логічного висновку виконує наступні дії:

• обирає правила, яким відповідають факти;
• розподіляє обрані правила за пріоритетами;
• виконує правило з найвищим пріоритетом.

Як класичні стратегії рішення задач в експертних системах використовуються два загальні методи логічного висновку: прямий логічний висновок і зворотний логічний висновок. У число інших методів, що застовуються для виконання конкретизованих завдань, можуть входити: аналіз цілей та засобів, спрощення задачі, перебір з поверненнями, метод "запланувати-виробити-перевірити", ієрархічне планування.

Робоча пам 'ять. Робоча пам'ять призначена для розміщення фактів, що стосуються поточного стану об'єкта досліджень. Факти, що знаходяться в робочій пам'яті, не взаємодіють один з одним, на відміну від правил, що зберігаються в базі знань. Якщо в робочій пам'яті є факт, який відповідає умовній частині правила, машина логічного висновку розміщує це правило в робочому списку правил.

У випадку, якщо правило має декілька шаблонів, то для того, щоб правило можна було розмістити в робочому списку правил, усі ці шаблони повинні бути розпізнані як відповідні. Як умову відповідності деяких шаблонів можна назвати відсутність певних фактів у робочій пам'яті. Машина логічного висновку працює в режимі здійснення циклів "розпізнавання-дія".

Робочий список правил. Робочий список правил являє собою створений машиною логічного висновку та розташований за пріоритетами список правил, шаблони яких відповідають фактам, що знаходяться в робочій пам'яті. Правило, всі шаблони якого розпізнані як відповідні, називається активізованим або реалізованим. У робочому списку правил можуть бути одночасно присутні декілька активізованих правил. У цьому випадку машина логічного висновку повинна вибрати, залежно від пріоритету, одне з правил для запуску дії.

Після завершення виконання всіх правил управління повертається до інтерпретатора команд верхнього рівня, щоб користувач міг надати командному інтерпретатору експертної системи додаткові інструкції. Роль верхнього рівня системи виконує інтерфейс користувача, який, по суті, є механізмом інтерпретації команд користувача.

Засіб пояснення. Головною особливістю експертної системи є передбачений у ній засіб пояснення, який відображає інформацію про те, як

система дійшла певного висновку. У системах, заснованих на правилах, нескладно організувати пояснення, яким чином був отриманий певний висновок, оскільки хронологія активізації правил і вміст робочої пам'яті можна зберігати в стеку. Розвинені засоби пояснення можуть дати користувачу можливість ставити питання на зразок "що?", "якщо?" і вивчати альтернативні шляхи формування висновків за принципом гіпотетичних міркувань.

2. Методи практичної реалізації концепції продукційних правил

Методи реалізації продукційних правил пройшли еволюційний процес із середини XX століття. Оскільки спосіб представлення знань на основі продукційних правил є класичним засобом створення експертних систем, їх розробникам необхідно володіти системною інформацією щодо методичного змісту етапів розвитку концепції правил.

Продукційні системи були вперше використані в символічній логіці американським логіком Емілем Постом (Emil Post), тому ім'я цього вченого увійшло до назви вказаних систем. Основна ідея Поста полягала в тому, що на основі логічної та математичної систем можна представити набір правил, який встановлює порядок перетворення рядка символів в інший послідовний набір символів. Це означає, що продукційне правило, після отримання вхідного рядка (антецедента), здатне виробити новий рядок (консеквент), наприклад:

Клієнт має постійний дохід — клієнт кредитоспроможний

У цьому правилі стрілка означає, що один символьний рядок повинен бути перетворений в інший. Вказане правило можна інтерпретувати за допомогою системи позначень IF-THEN наступним чином:

IF клієнт має постійний дохід THEN клієнт кредитоспроможний

Слід зазначити, що маніпуляції з рядками засновані на синтаксисі, а не на семантиці. Іншими словами, продукційна система Поста застосовується лише як спосіб перетворення одного рядка в інший, без розуміння значення слів "постійний дохід" та "клієнт кредитоспроможний".

Продукційні правила також можуть мати декілька антецедентів, що розділяються зв'язкою AND, як показано в наступному прикладі:

Клієнт має постійний дохід AND він володіє нерухомістю вартістю 1000000$ — видати кредит

Безумовно, продукційні правила Поста були необхідні для формування певної частини фундаменту експертних систем, але вони не були достатніми для програмування продуктів, що мають практичну цінність. Основним обмеженням продукційних правил Поста, з погляду програмування, є відсутність стратегії управління, яка дозволяла б регламентувати застосування правил. Зрозуміти це обмеження можна легко, уявивши гіпотетичну ситуацію з візитом в бібліотеку. Якщо для пошуку потрібної книги не використовувати жодної системи управління процесом, можна згаяти безліч часу, переглядаючи усі можливі варіанти.

Наступним значним кроком у розробці методів застосування продукційних правил стало відкриття, зроблене Марковим, яке дозволило

визначити структуру управління для продукційних систем. Мар-ківський алгоритм - це впорядкована група продукцій, які застосовуються згідно з пріоритетами до вхідного символьного рядка. Якщо правило з найвищим пріоритетом є непридатним, то використовується наступне правило з нижчим пріоритетом і т.д. Марківський алгоритм завершує свою роботу після виявлення однієї з наступних умов: по-перше, остання продукція не була застосовна до рядка, або, по-друге, була застосована продукція, яка закінчується крапкою.

Марківські алгоритми можуть також застосовуватися до сегментів символьних рядків, починаючи зліва. Наприклад, продукційна система складається з одного правила:

ЗА — ВИ

Після її застосування вхідний рядок "ЗАЛУЧИТИ" перетворюється на новий рядок "ВИЛУЧИТИ". Оскільки тепер продукція застосовується до нового рядка, остаточним результатом стає рядок "ВИЛУЧИТИ".

Марківські алгоритми застосовують спеціальні символи. Зокрема, спеціальний символ А позначає порожній рядок, що не містить символів. Наступна продукція видаляє всі входження символу А у вхідному рядку:

А — л

Інші спеціальні символи Марківського алгоритму можуть представляти інші набори символів та позначаються буквами a, b, c, ..., y, z. Ці символи є односимвольними змінними і являють собою важливу частину сучасних мов експертних систем. Наприклад, наступне правило змінює місцями символи А та В в рядку, якщо між ними знаходиться будь-який одиничний символ:

АхВ —ВхА

Як спеціальні знаки пунктуації у Марківському алгоритмі використовуються грецькі літери б, в і т.д. Вони застосовуються тому, що їх можна легко відрізнити від звичайних літер алфавіту.

Марківський алгоритм може застосовуватися як основа експертної системи, але він не є достатньо ефективним способом створення систем із великою кількістю правил. Такі системи потребують алгоритму, який має повну інформацію про всі правила та може застосувати будь-яке з них, не роблячи спроби послідовно перевіряти кожне.

Рішенням цієї проблеми є rete-алгоритм, розроблений Чарльзом Л. Форгі (Charles L. Forgy) в 1979 році. Термін "rete-алгоритм" походить від латинського слова rete, яке означає мережу. Згідно з назвою rete-алгоритм функціонує як мережа, призначена для зберігання великого обсягу знань. Він заснований на використанні динамічної структури даних, яка автоматично реорганізується з метою оптимізації пошуку аналогічно В-дерева, що застосовується при індексації структур реляційних баз даних.

Rete-алгоритм є високошвидкісним засобом порівняння фактів із шаблонами, швидкодія якого досягається завдяки зберіганню в оперативній пам'яті інформації про правила, що знаходяться в мережі. В rete-алгоритмі втілені два емпіричні спостереження, на підставі яких була запропонована структура даних, покладених у його основу:

1. Тимчасова надмірність. Дія, що виникає в результаті запуску одного з правил, зазвичай, пов'язана з декількома фактами.

2. Структурна подібність. Один і той самий шаблон часто знаходиться в лівій частині більш ніж одного правила.

В rete-алгоритмі в циклах "розпізнавання-дія" контролюються тільки зміни в узгодженнях, тому в кожному циклі немає потреби погоджувати факти з кожним правилом. Завдяки цьому істотно підвищується швидкість узгодження фактів з антецедентами, оскільки статичні дані, які не змінюються від циклу до циклу, можуть бути проігноровані.

Необхідною умовою практичної реалізації концепції продукційних правил є використання формальної системи визначення продук-цій. У загальному вигляді під формальною системою визначення розуміють систему позначень, яка виконує роль метамови для визначення синтаксису інших мов. Мови поділяються на декілька типів: природні мови, логічні мови, мови математики, комп'ютерні мови тощо. Синтаксис мови визначає її форму, а семантика - значення її слів.

Однією із формальних систем позначень, що використовуються для визначення продукцій, є нормальна форма Бекуса-Наура (Backus-Naur form - BNF). Для її детального розгляду скористаємося простим правилом граматики англійської мови, представленим у вигляді продукційного правила:

<sentence>::= <subject> <verb> <end-mark>

Згідно з системою позначень форми Бекуса-Наура речення (sentence) складається з підмета (subject) та присудка (verb), за якими йде знак пунктуації, що позначає кінець речення (end-mark). У цьому правилі кутові дужки (<>) та символ ": : =" є символами метамови. Символ "::=" означає "визначено як". Він використовується замість символу ", який застосовувався в Марківських алгоритмах продук-ційних правил.

Інтуїтивно визначений вираз метамови, що являє собою формальний ідентифікатор об'єкта, носить назву терма. Терми, поміщені в кутові дужки, прийнято називати нетермінальними символами. Нете-рмінальний символ - це змінна, що представляє інший терм. У свою чергу, в якості такого "іншого" терму може використовуватися нетер-мінальний або термінальний символ.

Термінальний символ не може бути замінений іншим символом і тому є константою. З його допомогою реалізується остання ланка ланцюга утворень. Наприклад, наступні правила дозволяють розкривати значення нетермінальних символів, оскільки вказують на термінальні символи, якими вони можуть бути замінені:

<subject> :: = Гривня | Долар | Євро

<verb> ::= дорожчає | дешевшає

<end-mark> ::= . | ! | ?

У даній метамові вертикальна риска означає "або".

Усі можливі речення мови, тобто продукції, можуть бути створені шляхом послідовної заміни кожного нетермінального символу відповідними йому нетермінальними або термінальними символами, взятими з правої

частини правил, до тих пір, доки не будуть усунені всі нетермінальні символи. Як приклад застосування продукцій даної мови можна навести такі речення:

Гривня дорожчає. Долар дешевшає! Євро дешевшає?

Послідовність термінальних символів називається рядком символів мови. Якщо рядок символів може бути одержаний з початкового символу шляхом заміни нетермінальних символів із застосуванням правил, що їх визначає, то такий рядок називається дійсним реченням. В іншому випадку рядок не вважається дійсним реченням, наприклад:

Євро дешевшає дешевшає

Повний набір продукційних правил, що однозначно визначає мову, називається її граматикою. Розглянуті в прикладі правила визначають деяку граматику, але вона є дуже обмеженою, оскільки забезпечує створення невеликої кількості можливих продукций. Складність граматики збільшується за рахунок різних доповнень, наприклад:

<sentence>::= <subject> <verb> <object> <end-mark>

<object> ::=на 1% | на 5% | на 10%

Однак, незважаючи на рівень досягнутої складності граматики, принцип системи визначення продукцій залишається незмінним.

Як приклад практичної реалізації концепції продукційних правил наведемо фрагмент програми управління надзвичайними ситуаціями на підприємстві, яка реалізована в системі CLIPS. Експертна система в системі CLIPS може виконувати змістовну роботу тільки в тому випадку, якщо в ній присутні факти і правила. В нашому прикладі розглянемо, які типи фактів і правил можуть виявитися корисними в тій ситуації, коли доводиться здійснювати поточний контроль і реагувати на декілька можливих надзвичайних ситуацій. Однією з таких надзвичайних ситуацій може виявитися пожежа, а іншою - повінь. Нижче наведений псевдокод одного з можливих правил в експертній системі поточного контролю на підприємстві.

IF the emergency is a fire

THEN the response is to activate the sprinkler system

Перш ніж перетворити цей псевдокод в правило, необхідно визначити конструкції deftemplate для фактів такого типу, які згадують -ся в цьому правилі. Надзвичайна ситуація може бути представлена за допомогою наступної конструкції deftemplate:

(deftemplate emergency (slot type))

У цій конструкції поле type факту emergency повинно містити такі символи, як fire (пожежа), flood (повінь) і power outage (припинення подачі електроенергії). Аналогічним чином, відповідь експертної системи (response), може бути представлена наступною конструкцією deftemplate:

(deftemplate response (slot action))

У даній конструкції поле action факту response указує на те, яка відповідь повинна бути вироблена системою.

Правило, виражене за допомогою синтаксису CLIPS, наведене нижче, в ньому використовуються коментарі, відповідні окремим частинам правила. Коментарі починаються з крапки з комою і продовжуються до кінця рядка.

Коментарі ігноруються системою CLIPS. Правило можна ввести, набираючи його текст у вбудованому редакторі системи CLIPS:

```
; Заголовок правила
(defrulefire-emergency "An example rule" ; Шаблони
(emergency (type fire)) ; Стрілка THEN
=>
; Дії
(assert (response
(action activate-sprinkler-system))))
```

У правилі присутній один шаблон і одна дія. Заголовок правила складається з трьох частин. Правило повинно починатися з ключового слова defrule, за яким йде ім'я правила - fire-emergency. За заголовком правила слідує від нуля і більше умовних елементів. Простим типом умовного елемента є умовний елемент шаблону, або просто шаблон. Кожен шаблон складається з одного або декількох обмежень, які призначені для зіставлення з полями факту, заданого за допомогою конструкції deftemplate. У правилі fire-emergency шаблоном є (emergency (type fire)). Обмеження для поля type указує на те, що це правило задовольняється тільки для фактів emergency, що містять в своєму полі type символ fire. Система CLIPS робить спроби зіставити шаблони правил з фактами в списку фактів. Якщо всі шаблони правила узгоджуються з фактами, правило активізується і потрапляє в робочий список правил - в колекцію активізованих правил. У робочому списку правил може знаходитися від нуля і більше правил.

Символ =>, який іде за шаблонами в правилах, називається стрілкою. Стрілка - це символ, що представляє початок частини THEN правила IF-THEN. Частина правила, що знаходиться перед стрілкою, називається лівою частиною (Left-Hand Side — LHS), а частина, що знаходиться за стрілкою, називається правою частиною (Right-Hand Side — RHS).

Останньою частиною правила є список дій, що виконуються після запуску правила. У даному прикладі одна з дій передбачає внесення факту (response(action activate-sprinkler-system)) в список фактів.

3. Представлення знань на основі формальної логіки

В експертних системах для представлення знань, окрім правил, можуть використовуватися символи формальної логіки. Нагадаємо, що логіка - це наука, яка вивчає правила формування обґрунтованих міркувань. Термін "формальний" означає, що логіка, до якої він належить, розповсюджується тільки на форму логічних тверджень, але не враховує їх значення. Іншими словами, у формальній логіці розглядається тільки синтаксис тверджень і не розглядається їх семантика. В результаті відділення форми від семантики з'являється можливість об'єктивно оцінювати правильність доказу, не піддаючись дії упереджень, викликаних семантикою.

Аналогією по відношенню до формальної логіки може розглядатися алгебра, в якій правильність таких виразів, як X + X = 2X залишається

незаперечною, незалежно від того, що позначає X: кількість яблук або аеропланів. Така властивість формальної логіки є корисною при створенні експертних систем, оскільки дозволяє відділити знання від міркувань. Тобто вислови, які, на перший погляд, виглядають, як міркування, можуть, у дійсності, виконувати роль знань.

Важливою частиною процесу проведення міркувань є логічне виведення висновків із посилань. На цьому ґрунтується найбільш рання система формальної логіки, яка була розроблена давньогрецьким філософом Аристотелем у IV ст. до н.е. Ключовим поняттям Аристоте-левої логіки є силогізм. У силогізмі посилання виконують роль свідоцтв, з яких повинен обов'язково бути сформований висновок. Силогізми мають два посилання і один висновок, який з них витікає. Класичний приклад силогізму може мати наступний вигляд:

Посилання 1: Усі люди смертні

Посилання 2: Викладач - людина

Висновок: Викладач смертний

Для графічного способу представлення силогізмів добре підходять діаграми Венна. Приклад представлення знання, приведеного в силогізмі, у вигляді діаграми Венна зображений на рис. 2.

Рис. 2. Представлення силогізму у вигляді діаграми Венна

З погляду математики, кожен еліпс на діаграмі Венна представляє певну множину, тобто колекцію об'єктів. Таким чином, механізм теорії множин може бути успішно застосований для формального представлення знань у логічній моделі. Діаграми Венна, що зображують основні операції над множинами, представлені на рисунках 3-5.

У теорії множин певну зручність надає практика вважати досліджувані множини підмножинами однієї універсальної множини - універсуму. Графічно універсум зображується як прямокутник, що оточує свої підмножини. Найбільш уживані символи в теорії множин наступні:

• символом у вигляді грецької букви епсилон (Є) позначається приналежність об'єкта або множини до іншої множини;

• символом ∩ - перетин двох множин;

• символом ∪ - об'єднання двох множин;

• символом штрих (') - доповнення множини.

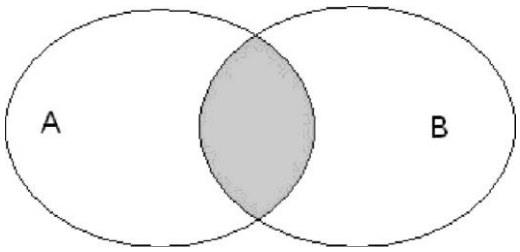

Рис. 3. Перетин множин

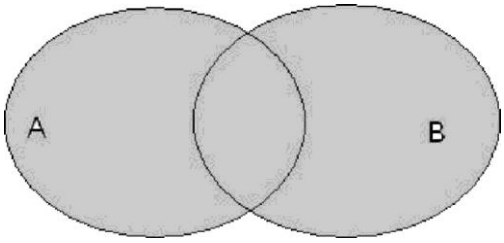

Рис. 4. Об'єднання множин

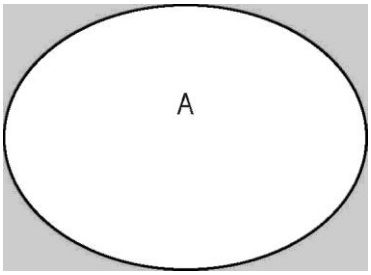

Рис. 5. Доповнення множини

Аристотелеві силогізми залишалися фундаментом логіки до 1847 року, коли англійський математик Джордж Буль (George Boole) опублікував першу книгу з описом символічної логіки. Одним з нових понять, запропонованих Булем, з'явилася модифікація представлення Аристотеля, що має назву "екзистенціальне значення", згідно з яким суб'єкт міркувань повинен існувати. Наприклад, відповідно до класичних Аристотелевих поглядів такий вислів, як "Дід Мороз приносить подарунки 31 грудня", не може використовуватися як посилання або наслідок, оскільки Діда Мороза насправді не існує.

Булеві представлення, що мають наразі статус сучасних представлень, дозволяють міркувати про порожні класи об'єктів. Тобто дозволяють створювати логічні конструкції для прогнозування наслідків використання результатів послідовності дій, які планується отримати вперше.

Ще один внесок у розвиток символічної логіки, зроблений Булем, полягав у тому, що вчений дав визначення поняття формулювання аксіом. Формулювання аксіом, згідно з Булем, складається із символів, які використовуються для представлення об'єктів і класів, а також операцій алгебри - для маніпулювання цими символами.

4. Застосування елементів пропозиціональної логіки в представленні знань

Для маніпулювання висловами використовується символічна логіка, що має назву "пропозиціональна логіка". Зокрема, пропозиціона-льна логіка може використовуватися для маніпулювання логічними змінними, що позначають вислови. Модель представлення знань із застосуванням пропозиціональної логіки часто представляється в науковій літературі як обчислення висловів.

У пропозиціональній логіці розглядається певний тип речень природної мови. Усі речення природної мови поділяються на чотири основні типи: наказові, питальні, окличні, оповідні. Пропозиціональна логіка розглядає оповідні речення. Вони можуть розділятися на істинні та помилкові. Речення, істинне значення якого може бути визначене, називається твердженням або висновком. Твердження прийнято називати також закритим реченням, оскільки його значення не підлягає обговоренню, зважаючи на його істинність.

Речення, про істинність яких неможливо дати однозначну відповідь, називаються відкритими реченнями. Наприклад, відкритим є речення "Якість позичальника визначається наявністю постійного місця роботи", оскільки воно є істинним для одних людей і помилковим для інших. Неоднозначністю такого роду неможливо ефективно оперувати за допомогою пропозиціональної логіки. Подібні проблеми розв'язуються з використанням нечіткої логіки.

Обчислення висловів здійснюється на основі логічних зв'язок до окремих висловів, у результаті застосування яких формуються складні вислови. Для цього застосовуються логічні зв'язки, найбільш поширені з яких наступні:

• AND (Λ) - кон'юнкція;
• OR (V) - диз'юнкція;
• якщо... то... (->) - умовна операція;
• якщо і тільки якщо (<->) - двостороння умовна операція;
• NOT (~) - заперечення.

Кон'юнкція здійснюється завдяки застосуванню логічного "І" між двома висловами. Її результат - складний вислів буде істинним, якщо істинні обидва вислови. Диз'юнкція здійснюється завдяки застосуванню логічного "АБО" між двома висловами. Її результат буде істинним, якщо істинний хоча б один із цих висловів.

Умовна операція аналогічна формі IF-THEN продукційних правил. Наприклад, наступний вираз:

IF клієнт має постійний дохід THEN видати кредит

може бути представлений у такій формі:

p -> q

де застосовуються наступні позначення: p - клієнт має постійний дохід, q - видати кредит.

Двостороння умовна операція p q еквівалентна виразу:

(p -> q) Λ (q -> p)

Вона є істинною тільки тоді, коли p і q мають одночасно істинне або помилкове значення.

Усі операції пропозіціональної логіки, що були розглянуті, належать до бінарних. На відміну від них, операція заперечення є унарною, оскільки застосовується до одного виразу. Вона має вищий пріоритет у порівнянні з іншими операціями.

Сутність логічних зв'язок добре демонструє їх застосування для формалізованого опису таких логічних понять, як тавтологія та суперечність. Тавтологія - це складний вислів, який завжди є істинним, незалежно від того, істинні чи помилкові окремі вислови, з яких він складається. Прикладом тавтології може бути вислів: "Начальник завжди має рацію, оскільки він ніколи не може не мати рації".

На відміну від фактів, які можуть бути в реальному світі істинними або ні, тавтологія завжди, в чисто логічному значенні, істинна, оскільки посилається для доказу на саму себе. А суперечність являє собою складний вислів, який завжди є помилковим. Вислови, які не є ні тавтологією, ні суперечністю, у пропозиціональній логіці називаються контингентними.

Тавтології та суперечності називаються, відповідно, аналітично істинними і аналітично помилковими висловами, оскільки істинність їхніх значень може бути визначена виключно на підставі аналізу форми. Зокрема, істинне значення для виразу:

p V ~p

показує, що це - тавтологія, а для виразу:

p Λ ~p,

що це - суперечність.

Множина логічних зв'язок називається адекватною, якщо за допомогою зв'язок, узятих виключно з цієї множини, можна представити будь-яку функцію істини. До прикладів адекватних множин можна віднести множини: {~, Λ, V}, {~, Λ }, {~, V} .

Незважаючи на те, що пропозиціональна логіка є дуже корисною, в представленні знань вона має певні обмеження. Основна проблема полягає у тому, що пропозиціональна логіка може застосовуватися тільки з повним висловом. Це означає, що з її допомогою не можна досліджувати внутрішню структуру вислову. Наприклад, пропозіцио-нальная логіка не дозволяє довести обґрунтованість наведеного раніше силогізму стосовно смертності викладача.

5. Логіка предикатів в експертних системах

З метою забезпечення вирішення проблем пропозиціональної логіки була розроблена логіка предикатів. Причому пропозиціональна логіка розглядається сьогодні як підмножина логіки предикатів. Предикат (лат. ргесіісапші - заявлене, сказане) - це термін логіки та мовознавства, що означає конститутивний член вислову. Тобто предикат являє собою дещо, що стверджується або заперечується про суб'єкт.

Логіка предикатів дозволяє розглядати внутрішню структуру висловів. У ній допускається використання таких спеціальних слів, як: "все", "деякі",

"жоден". Ці слова називаються кванторами. Квантори дозволяють надавати явні кількісні оцінки іншим словам і точніше формулювати вислови. Всі квантори відповідають на питання "скільки", і тому дозволяють застосовувати ширший круг виразів порівняно з пропозиціональною логікою. Існують різні види кванторів, і для детального розгляду їх сутності візьмемо найпоширеніші з них - квантор загальності та квантор існування.

Квантор загальності. Квантор загальності забезпечує організацію універсальних висловів. Вислів з квантором загальності приймає однаково істинне значення при підстановці всіх об'єктів з однієї області визначення.

Квантор загальності представляється за допомогою символу" , за яким йде один або декілька параметрів, що відповідають змінним

області визначення. Символ " інтерпретується як "для кожного" або "для всіх". Наприклад, вираз:

(V x)(x + x = 2x)

свідчить про те, що для кожного x (де x — число) вираз x + x = 2x є істинним. Якщо цей вираз буде позначений символом p, то наведене вище твердження може бути представлене більш стисло наступним чином:

(V x)(p).

Слід зазначити, що разом з фіктивними змінними x і p можна використовувати інші змінні, вислови та функції. Припустимо, що H - предикативна функція, що позначає людей, а M - функція, що позначає смертних істот. У такому разі твердження, згідно з яким всі люди смертні, можна записати наступним чином:

(V x)(H(x) -> M(x)).

Це твердження читається так: для всіх x, якщо x - людина, то x -смертна. Ця пропозиція логіки предикатів може бути також виражена в термінах продукційних правил:

IF x - людина THEN x - смертна

Квантор загальності може інтерпретуватися як кон'юнкція предикатів, що належать до окремих екземплярів. Під екземпляром тут розуміється конкретний випадок. Наприклад, припустимо, що людина з прізвищем Петренко Р.М. є конкретним екземпляром класу клієнтів банку. Тоді цю думку можна виразити через предикативну функцію "Клієнт", для якої Петренко Р.М. буде аргументом:

Клієнт(Петренко Р.М.).

Використання кон'юнкції дозволяє вислів логіки предикатів, представлений у вигляді:

(V x)P(x)

інтерпретувати в термінах екземплярів ai: P(a1) Λ P(a2) Λ P(a3) Λ ... Λ P(aN)

У цьому випадку послідовність крапок (...) вказує, що дія предиката розповсюджується на всі елементи даного класу. Таким чином, у наведеному виразі сказано, що предикат застосовується до всіх екземплярів класу.

У виразах може бути декілька кванторів. Наприклад, для формулювання закону комутативності суми чисел потрібні два квантори:

(Vx)(Vy)(x + y = y + x)

У цьому виразі стверджується, що "для кожного x і для кожного y сума x та y дорівнює сумі y та x".

Квантор існування. Квантор існування визначає твердження як істинне стосовно, принаймні, одного елемента області визначення. Він є обмеженою формою квантора загальності. Квантор існування записується як символ $, за яким йде одна або декілька змінних, наприклад:

($x)(x • x = 1),

($x)(Клієнт (x) Л Прізвище(Петренко Р. М.))

У першому з виразів вказано, що існує число x, результат множення якого на самого себе дорівнює одиниці. У другому виразі сказано, що існує клієнт на прізвище Петренко Р. М.

Квантор існування можна прочитати декількома способами, зокрема: "існує", "деякий", "щонайменше один". Так само, як квантор загальності може бути виражений за допомогою кон'юнкції, квантор існування може бути виражений за допомогою диз'юнкції екземплярів, a,:

P(ai) VP(a2) VP(аз) V ... VP(aN).

Багато типів висловів можна представити на основі логіки предикатів із використанням кванторів загальності та існування. Проте вона має деякі обмеження для представлення знань в експертних системах. Наприклад, у логіці предикатів неможливо виразити наступний вислів: "Більшість клієнтів банку взяли кредит у доларах". У ньому квантор "більшість" означає "більше половини".

Квантор "більшість" не може бути виражений у термінах квантора загальності та квантора існування. Для реалізації квантора "більшість" у логіці повинні бути передбачені предикати, які забезпечують підрахунок кількості елементів (що можливо при використанні нечіткої логіки).

Ще одне обмеження логіки предикатів полягає у тому, що вона не дозволяє виражати залежності, які можуть бути істинними іноді, але не завжди. Вказана проблема також може бути розв'язана за допомогою нечіткої логіки. Проте впровадження в логічну систему засобів проведення обчислень спричиняє появу додаткових ускладнень.

6. Нечітка логіка в експертних системах

Міркування, що спирається виключно на точні факти та точні висновки, які виходять із цих фактів, називаються строгими міркуваннями. У випадках, коли для прийняття рішень потрібно використовувати невизначені факти, строгі міркування стають непридатними. Тому однією з найсильніших сторін будь-якої експертної системи вважається її здатність формувати міркування в умовах невизначеності так само успішно, як це роблять експерти-люди. Такі міркування мають характер нестрогих.

Невизначеність може розглядатися як недостатність адекватної інформації для прийняття рішення. Невизначеність стає проблемою, оскільки

може перешкоджати створенню найкращого рішення і навіть стати причиною того, що буде знайдене неякісне рішення. Слід відмітити, що якісне рішення, знайдене в реальному часі, часто вважається більш прийнятним, ніж найкраще рішення, для обчислення якого потрібна велика кількість часу. Наприклад, затримка в наданні лікування з метою проведення додаткових аналізів може привести до того, що пацієнт помре, не дочекавшись допомоги.

Причиною невизначеності стає наявність в інформації різноманітних помилок. Спрощена класифікація цих помилок може бути представлена в їхньому розділенні на наступні типи:

• неоднозначність інформації, виникнення якої пов'язано з тим, що деяка інформація може інтерпретуватися різними способами;

• неповнота інформації, яка пов'язана з відсутністю деяких даних;

• неадекватність інформації, обумовлена застосуванням даних, що не відповідають реальній ситуації (можливими причинами є суб'єктивні помилки: брехня, дезінформація, несправність устаткування);

• погрішності вимірювання, що виникають через недотримання вимог правильності та точності критеріїв кількісного представлення даних;

• випадкові помилки, проявом яких є випадкові коливання даних щодо середнього їх значення (причиною можуть бути: ненадійність устаткування, броунівський рух, теплові ефекти тощо).

На сьогодні розроблена значна кількість теорій невизначеності, в яких робиться спроба усунення деяких або навіть усіх помилок та забезпечення надійного логічного висновку в умовах невизначеності. До найбільш уживаних на практиці належать теорії, засновані на класичному визначенні ймовірності та на апостеріорній імовірності.

Одним з найстаріших і найважливіших інструментальних засобів рішення задач штучного інтелекту є ймовірність. Імовірність - це кількісний спосіб урахування невизначеності. Класична ймовірність бере початок із теорії, яка була вперше запропонована Паскалем і Ферма в 1654 році. З того часу була проведена велика робота у сфері вивчення ймовірності, здійснені численні застосування ймовірності в науці, техніці, бізнесі, економіці та інших областях.

Класичну ймовірність називають також апріорною ймовірністю, оскільки її визначення належить до ідеальних систем. Термін "апріорна" позначає ймовірність, що визначається "до подій", без урахування багатьох чинників, що існують у реальному світі. Поняття апріорної ймовірності розповсюджується на події, що відбуваються в ідеальних системах, несхильних до зношення або впливу інших систем. В ідеальній системі поява будь-якої з подій відбувається однаково, завдяки чому їх аналіз стає набагато простішим.

Фундаментальна формула класичної імовірності (P) визначена наступним чином: $P = W/N$

У цій формулі W - кількість очікуваних подій, а N - загальна кількість подій з рівними ймовірностями, які є можливими результатами експерименту або випробування. Наприклад, імовірність випадання будь-якої грані

шестигранної грального кістки дорівнює 1/6, а витягання будь-якої карти з колоди, що містить 52 різні карти - 1/52.

Формальна теорія ймовірності може бути створена на основі трьох аксіом:

1. Аксіома 1. У цій аксіомі стверджується, що областю визначення ймовірності події (Е) є дійсні числа від 0 до 1. Від'ємні значення ймовірності не допускаються. Достовірній події привласнюється ймовірність 1, а неможливій події - імовірність 0.

2. Аксіома 2. У даній аксіомі стверджується, що сума ймовірності всіх подій, незалежних одна від одної, що називаються взаємовиключними подіями, дорівнює 1.

3. Аксіома 3. У цій аксіомі вказано, що якщо події Еі і Е2 не можуть виникати одночасно (тобто є взаємовиключними подіями), то ймовірність виникнення тієї або іншої події дорівнює сумі ймовірностей цих подій.

Наведені аксіоми дозволили закласти фундамент теорії ймовірності, проте в них не розглядається імовірність подій, що відбуваються в реальних - неідеальних системах. На відміну від апріорного підходу, в реальних системах для визначення ймовірності деякої події Р(Е) застосовується спосіб визначення експериментальної ймовірності як ліміту розподілу частот.

В основу визначення апостеріорної ймовірності покладене вимірювання частоти, з якою виникає деяка подія під час проведення великої кількості випробувань. Наприклад, визначення соціального типу кредитоспроможного клієнта банку на основі емпіричного досвіду.

Події, що не належать до взаємовиключних, можуть впливати одна на одну. Такі події входять до класу складних. Імовірність складних подій може бути обчислена шляхом аналізу відповідних їм вибіркових просторів. Ці вибіркові простори можуть бути представлені за допомогою діаграм Венна.

Існує задача, яка є по суті протилежною задачі обчислення умовної ймовірності. Вона полягає у визначенні зворотної імовірності, яка показує ймовірність попередньої події з урахуванням тих подій, що відбулись у подальшому. На практиці з імовірністю такого типу доводиться зустрічатися досить часто, наприклад, при проведенні медичної діагностики або діагностики устаткування, при якій виявляються певні симптоми, а задача полягає в тому, щоб знайти ймовірну причину.

Для вирішення цієї задачі застосовується теорема Байєса, названа на честь британського математика XVIII століття Томаса Байєса. Байєсівська теорія в наші дні широко використовується для аналізу дерев рішень в економіці та суспільних науках.