

А. КОЗИРСЬ, І. ШУБІН

МЕТОД ЛІНІЙНО-ЛОГІЧНИХ ОПЕРАТОРІВ ТА ЛОГІЧНИХ РІВНЯНЬ У ЗАВДАННЯХ ВИДОБУВАННЯ ІНФОРМАЦІЇ

Реляційні та логічні методи подання знань відіграють ключову роль у створенні математичного базису для інформаційних систем. Алгебра предикатів і оператори предикатів є одними з найбільш ефективних інструментів для детального опису інформації. Ці засоби дозволяють легко формулювати формалізовану інформацію, створювати запити до баз даних та імітувати людську діяльність. У контексті нової необхідності в надійному та ефективному відборі даних виникає проблема в глибшому аналізі. **Предмет дослідження** – теорія кванторних лінійних рівнянь на базі алгебри лінійних предикатних операцій, формального апарата лінійно-логічних операторів і методів розв'язання логічних рівнянь у завданнях видобування інформації. **Мета роботи** – розроблення методу застосування лінійно-логічних операторів і логічних рівнянь для видобування інформації. Цей підхід може значно оптимізувати вилучення необхідної інформації, навіть у величезних базах даних. **Основні завдання:** аналіз наявних підходів до видобування інформації; розгляд теорії лінійно-логічних операторів; дослідження методів приведення логіки до алгебраїчної форми; аналіз логічних просторів та алгебри скінченних предикатних дій та теорії лінійно-логічних операторів. **Методи дослідження** передбачають систематичний аналіз математичної структури алгебри скінченних предикатів та предикатних функцій для ідентифікації ключових елементів, що впливають на процес формування запитів. Запропоновано метод застосування лінійно-логічних операторів і логічних рівнянь з метою видобування інформації. **Результати** дослідження показали, що метод використання лінійно-логічних операторів та логічних рівнянь є універсальним і адаптивним інструментом для роботи з алгебраїчними структурами даних. Він може бути застосований у широкому спектрі задач видобування інформації та здатний довести свою цінність як один із можливих методів оброблення інформації. **Висновок.** У роботі досліджено формальні методи інтелектуальних систем, зокрема способи подання знань відповідно до особливостей галузі застосування та мови, що дають змогу кодувати ці знання для зберігання в комп'ютерній пам'яті. Запропонований метод можна впроваджувати в розробленні мовних інтерфейсів для систем автоматизованого доступу до інформації, в алгоритмах пошукових систем, для логічного аналізу інформації в базах даних та експертних системах, а також у виконанні завдань, пов'язаних із розпізнаванням та класифікацією об'єктів.

Ключові слова: бази знань; інтелектуальні системи; алгебра скінченних предикатів; логічні оператори; кванторні лінійні рівняння.

Вступ

Здобутки у сфері обчислювальної техніки, у галузях мікроелектроніки та архітектури комп'ютерів установили технічні можливості сучасних обчислювальних пристроїв, такі як висока продуктивність і значна оперативна пам'ять. Це сприяло розширенню спектра завдань, що можуть бути виконані комп'ютерами, і підвищило їх роль у житті людини. Проте цей прогрес частіше має кількісний характер. Просте збільшення функціональних можливостей комп'ютера є ефективним лише за умови, що люди вміють їх ефективно використовувати. Інакше таке зростання стає безперспективним.

Поширення комп'ютерної техніки та її швидке вдосконалення привели до стрімкого розвитку інтелектуальних систем різних видів. Уже існують методи та підходи до створення та використання

інформаційних систем. Сучасні інтелектуальні системи можуть розв'язувати завдання, які раніше вважалися лише людською прерогативою: доводити математичні теореми, перекладати тексти з однієї мови іншою, діагностувати хвороби та виконувати багато інших функцій. Проте ідеальна обчислювальна машина має перевищувати здатність людини до логічного мислення, аналізу інформації та вирішення найскладніших завдань, а також взаємодії з навколишнім середовищем.

Важливу роль у розробленні інформаційних систем відіграють реляційні та логічні методи подання знань. Один з універсальних математичних способів для опису інформації – це алгебра предикатів та предикатних операцій. Ця алгебра дає змогу зручно подавати формалізовану інформацію, створювати запити до баз даних і моделювати діяльність людини.

Метою статті є створення методу із застосуванням розв'язання кванторних лінійних рівнянь на основі алгебри лінійних предикатних операцій, формальних інструментів лінійно-логічних операторів та методів розв'язання логічних рівнянь для задач видобування інформації.

Аналіз останніх досліджень і публікацій

Сучасні інтелектуальні системи працюють на основі глибоких знань про конкретну тематику, які вже наявні в їх базі знань. На відміну від зазначеного, старіші системи просто обробляли інформацію за допомогою програм. Є чітке розмежування між даними та знаннями, що підтверджено численними дослідженнями.

Під поданням знань розуміють формалізацію властивих переконань за допомогою фігур, записів або мов. Особливо цікавлять формалізації, сприймані комп'ютером. У зв'язку з цим розробляються формальні мови, що дають змогу відтворювати знання в пам'яті комп'ютера.

Особливість систем подання знань полягає в тому, що вони моделюють діяльність людини, здійснювану часто в нетиповому різновиді. Отже, важливим етапом розроблення інтелектуальних систем є створення оптимальної моделі подання знань про предметну галузь застосунку систем. Вочевидь вибір певного виду подання знань залежить від сфер формалізації. В останні роки була запропонована значна кількість різних моделей подання знань.

У логічній моделі подання знань використовується логіка предикатів першого порядку та виведення результатів за допомогою методу силіогізму. Використовувані в логічній моделі різниці предикатів можна легко єднати з доволі ефективним механізмом результату, наприклад резолюцією. Перевагами логічних моделей є одиничність теоретичного обґрунтування та можливість реалізації системи формально точних визначень і результатів. Саме з цієї причини інтелектуальні системи, що використовують логічну модель знань, значно поширені.

Особливістю роботи сучасних інтелектуальних систем є те, що процес оброблення інформації спирається на необхідні знання про проблемну сферу, заздалегідь додані до бази знань системи, тоді як раніше створені системи працювали з інформацією, що просто оброблялась різними програмами. Значна кількість досліджень присвячена

розбіжності даних від знань, зокрема їх основну ідею можна сформулювати так: знання є складно організованими типами даних, що різняться від традиційного подання чотирма основними особливостями:

1) знання розміщують у записі не тільки інформаційну частину, але й описову – у ній містяться всі дані про інформаційну одиницю, які можуть знадобитися в роботі користувача із системою;

2) знання в базі знань створюють складні, як правило, ієрархічні структури, що досягається впровадженням різних відношень на інформаційних одиницях, доданих до бази знань;

3) інформаційні одиниці, що відтворюють знання, можуть композуватися в більш складно організовані одиниці й декомпозуватися в простіші;

4) як частини інформаційних одиниць, що визначають якісь знання, можуть бути приєднані або вбудовані процедури, що дає змогу активізувати ці процедури внаслідок появи в базі знань різних інформаційних одиниць або зв'язків між ними.

Важливим кроком у розробленні таких систем є вибір найкращої моделі для подання знань у певній галузі. Вибір моделі, безумовно, залежить від сфери застосування. В останні роки багато нових моделей подання знань було запропоновано. Однак серед різних способів подання знань, що виникли внаслідок різноманіття структур знань, можна виокремити логічну модель, фреймові та продукційні системи, семантичні сітки. Будь-який спосіб подання має свої переваги й недоліки та пов'язаний з певною структурою та галузями застосування знань.

Як приклад, у роботі [1] розглядається оптимізація запитів *SPARQL* для децентралізованих графів знань, наголошується на ролі методів підвищення продуктивності запитів у розподілених наборах даних, що відкриває нові можливості для ефективної взаємодії з децентралізованою інформацією. У дослідженні [2] висвітлено рамкову концепцію для систем управління документами на основі хмари з інституціональною схемою бази даних, що вказує на розширення можливостей зберігання та доступу до корпоративної інформації, сприяючи гнучкості та масштабованості хмарних рішень. У роботі [3] підхід до інженерії запитів для вилучення структурованих даних з неструктурованого тексту за допомогою бесідних *LLM* відкриває нові перспективи для автоматизації оброблення інформації, зокрема де потрібна швидка адаптація до змінних умов даних. Аналіз кількісного

інтегритету зв'язаних даних на вебi, поданий у роботі [4], пропонує методи для забезпечення достовірності інформації, що є критично важливим для забезпечення надійності семантичних вебданих. Зберігання *Lua*-таблиць у реляційних базах даних, досліджене в статті [5], демонструє можливості інтеграції скриптових мов із традиційними системами управління базами даних, покращуючи гнучкість та ефективність розробки. Дослідження розподілених асинхронних регулярних запитів шляхів на графах, подане в роботі [6], відкриває нові напрями щодо аналізу складних структур даних, зокрема у сферах, де важлива висока продуктивність оброблення зв'язків. У праці [7] обговорюється вплив реконструкції на моделі подання метаданих, вказуючи на ключові фактори, які впливають на ефективність управління знаннями, що є важливим для оптимізації систем зберігання інформації. Об'єктно-орієнтований дизайн бази даних для ефективної класифікації, поданий у роботі [8], наголошує на потенціалі об'єктно-орієнтованих підходів в управлінні складними даними, сприяючи більшій гнучкості та ефективності. Фундаментальні концепції використання та поточні виклики систем управління векторними базами даних розглядаються в дослідженні [9], вказуючи на потенціал для оброблення значних обсягів інформації з високою точністю та швидкістю. У роботі [10] подано систему управління великими просторовими даними на основі *NoSQL*, що відкриває нові перспективи для оброблення просторових даних із високою ефективністю та масштабованістю. Тестування залежностей і правил виведення в базах даних, досліджене в статті [11], виявляє ключові аспекти для забезпечення цілісності та надійності інформації, що є фундаментальним для підтримки якості та безпеки інформаційних систем.

Після ретельного аналізу сучасних досліджень у галузі подання та оброблення інформації в базах даних стає очевидною необхідність розроблення нових методологій, що забезпечили б вищу ефективність оброблення запитів та аналізу даних, особливо в контексті зростання обсягів інформації. Огляд літератури дає змогу визначити наявні виклики та підходи, проте також відкриває перспективи для інновацій, зокрема використання лінійно-логічних операторів і логічних рівнянь у завданнях отримання інформації.

З огляду на визначені потреби та виклики, можна сформулювати **мету роботи** – створення

методу із застосуванням розв'язання кванторних лінійних рівнянь на основі алгебри лінійних предикатних дій і лінійно-логічних операторів для задач видобування інформації, що дасть змогу оптимізувати процеси запитів та аналізу інформації в базах даних. Цей метод передбачає використання складних логічних структур для формулювання та розв'язання задач визначення наявності певної інформації, забезпечуючи водночас високу точність та швидкість оброблення. Для досягнення цієї мети необхідно: проаналізувати наявні методики видобування інформації; ознайомитися з теоретичними основами лінійно-логічних операторів; дослідити способи перетворення логічних виразів у їх алгебраїчний аналог; проаналізувати логічні структури та алгебри, що використовуються для скінченних предикатних дій, а також дослідити теорію лінійно-логічних операторів.

Розроблений метод може бути застосований у створенні мовних систем для автоматизованих інформаційних систем, у системах пошуку інформації, у задачах логічного аналізу в базах даних, експертних системах, а також у вирішенні задач розпізнавання та класифікації об'єктів.

Матеріали й методи

Нещодавно виявлені численні практичні застосування абстрактної алгебри у сферах баз даних і систем штучного інтелекту спонукали до збільшення зацікавленості в алгебраїчному поданні інформації. Водночас практичний досвід вказує на нові неочікувані структури, що розширюють можливості алгебри [12–14]. Завдяки застосуванню алгебраїчних методів у програмуванні створено різноманітні компілятори для високорівневих мов та алгебричні алгоритми.

Автоматизація розроблення програмного забезпечення та проектування комп'ютерних систем є важливим та актуальним завданням у галузі обчислювальної техніки, що вимагає розвитку практично орієнтованої теорії алгоритмів. Одним із ключових завдань цієї теорії є створення оптимального компілятора для перекладу з однієї мови на іншу. Зазвичай, мова вищого рівня, призначена для конкретного спектра завдань, перетворюється на машинну.

Отже, необхідно провести трансляцію з мови програмування на машинну, й оптимізувати вихідний код. Виконання цього завдання передбачає декілька

проміжних кроків, на кожному з яких виконується часткова оптимізація алгоритму та його переклад на проміжну мову, що відповідає цьому етапу.

Нехай $s = \{f\}$ – множина мов і $P = \left\{ \begin{matrix} m \\ f', f'', f \end{matrix} \right\}$ – множина трансляторів, кожен з яких є програмою в мові f і перекладає програми із вхідної мови f' на вихідну f'' . Транслятор можна розглядати як унарну операцію, з областю визначення, що репрезентує собою мова f' , і областю значень f'' .

Нехай є транслятори

$$\left\{ \begin{matrix} m^1 \\ f', f'', f_1 \end{matrix} \right\} \text{ та } \left\{ \begin{matrix} m^2 \\ f', f'', f_2 \end{matrix} \right\}, \quad (1)$$

і можна обчислити операцію перекодування в такий спосіб:

$$m^2 \left[\begin{matrix} m^1 \\ f', f'', f_1 \end{matrix} \right] = \begin{matrix} m^1 \\ f', f'', f_2 \end{matrix}. \quad (2)$$

Над трансляторами $\begin{matrix} m \\ f_1, f_2, f_1 \end{matrix}$ й $\begin{matrix} m \\ f_2, f_3, f \end{matrix}$ відбувається операція композиції

$$\begin{matrix} m \\ f_1, f_2, f_1 \end{matrix} * \begin{matrix} m \\ f_2, f_3, f \end{matrix} = \begin{matrix} m \\ f_1, f_3, f \end{matrix}. \quad (3)$$

За допомогою наведених операцій можна формалізувати процеси, часто використовувані в програмуванні.

Щоб здійснювати еквівалентні переформування алгоритмів, необхідно розробити алгебру для подання алгоритмів, що дасть змогу виконувати такі переформування в зрозумілій алгебраїчній манері.

База даних – це інформаційна система, що зберігає та обробляє інформацію, надаючи відповіді на різноманітні запити. Вона має не лише надавати безпосередньо збережену інформацію, але й здатна виводити додаткову інформацію, базуючись на первинних даних. Цей процес виведення додаткової інформації тісно пов'язаний із реалізацією функцій у розумних системах [15].

У разі застосування алгебраїчного підходу до опису похідної інформації виокремлюється певна алгебраїчна система – алгебра запитів, у термінах якої похідна інформація записується через базисну. Базу даних подамо у вигляді деякої математичної моделі. Припустимо, що система множини даних – доменів – подана як $D = (d_i, i \in F)$, де F – множина імен доменів.

Для кожної системи множини $D = (d_i, i \in F)$ символ $\varphi = \phi$ типу τ реалізується як підмножина відношення в декартовому добутку $D_{i_1} \times \dots \times D_{i_n}$, тип τ характеризує розміщення змінних. Задаємо відображення $n: X \rightarrow F$, що розбиває множину змінних X за розрядами: $X = X_1 \cup \dots \cup X_K$, K – потужність множини F . Множина X має достатню кількість змінних для реалізації різних запитів. Систему показників $D = (d_i, i \in F)$ і введені на ній відношення розглядатимемо як модель. Правило, за яким реалізується будь-який символ φ відношення ϕ , позначимо f . Отже, стан бази даних у розглянутій схемі за умови заданої системи інформації можна визначити як функцію f , яка приписує кожному елементу типу τ відповідну підмножину $\varphi = \phi$ у визначеній $D_{i_1} \times \dots \times D_{i_n}$. Таку модель бази даних позначимо з допомогою (D, ϕ, F) , до того ж символи відношень реалізуються в певні миті. Далі в базі даних необхідно реалізувати можливість запиту. Запит U у стані f позначимо як $f * z$. Відповіді на запити подаються як підмножина D . Множина $f * z$ визначена для базисних запитів U , а довільний запит виражається через базисні запити.

Візьмемо множину всіх підмножин D та позначимо її як MD . Далі множину всіх потенційних запитів позначимо як U . Відповідь на кожен запит U з множини U для будь-якого стану f є частиною MD . Щоб можна було подати будь-який запит за допомогою основних запитів, маємо додати алгебраїчні операції в множині U , що дають змогу маніпулювати запитами. Також необхідно впровадити подібні алгебраїчні операції для множини відповідей. У цьому контексті на основі використаних алгебраїчних операцій можемо визначити відповідь на будь-який запит з огляду на його структуру та подати його з допомогою вже відомих запитів. Розглянуті множини запитів та відповідей утворюють алгебри запитів та відповідей.

Запити до бази даних можна подати у формі формул певної логічної мови, такої як мова різниць висловлювань або різниці предикатів різного рівня. У цьому разі їх виразні здатності можуть розрізнятися. Існують різниці як для класичних, так і некласичних логік. Наприклад, булеві алгебри кореспондують з класичною різницею висловлювань,

тоді як спеціальні алгебри Гейтінга пов'язані з інтуїціоністською різницею висловлювань.

Різні алгебраїчні структури, пов'язані з порядковою різницею, були запроваджені протягом певного часу. З-поміж них відомі циліндричні алгебри Тарського та поліадичні алгебри Халмоша [12]. Алгебра Халмоша розширюється за допомогою додавання операторів до кванторної алгебри, що ґрунтується на булевій алгебрі з кванторними операціями, доданими до цієї алгебри. Дію квантора можна подати як певну операторну дію в алгебрі. У статті досліджуватимуться різні методи введення таких операторів, але спершу розглянемо геометричний аспект кванторів як операторів.

Припустимо, що маємо змінні x та y , визначені на множині U . Нехай B є підмножиною U , яка є декартовим добутком U . Множина U може бути інтерпретована як бінарний предикат $B(x, y)$, що визначений на U^2 і приймає значення "так" для всіх пар (x, y) у B . Зазначимо проєкції U^2 як U_X та U_Y . Квантор існування, визначений як $\exists x B(x, y)$, вказує на унарний предикат для y на U , що визначає підмножину в U_Y , яка містить елементи B_Y у U_Y , для яких існують відповідні B_X в U_X , так що пара (B_X, B_Y) належить B . Тобто, застосовуючи квантор існування до предиката $B(x, y)$, отримаємо унарний предикат для y , що описує множину B_X . Геометрично $\exists x B(x, y)$ є проєкцією множини B на U_Y (рис. 1).

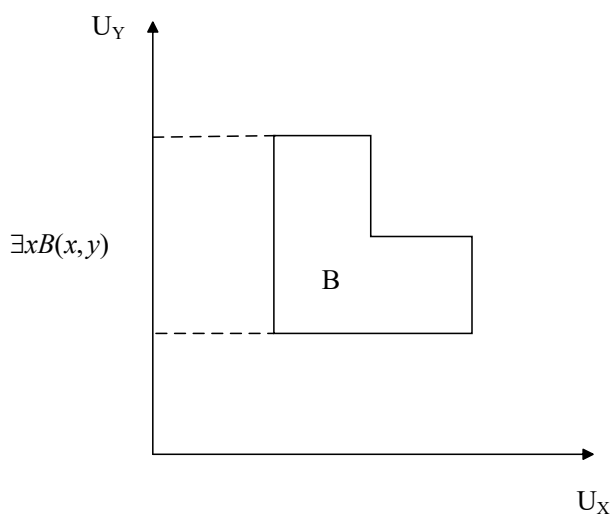


Рис. 1. Графічне подання квантора існування

Аналогічне застосування квантора існування по змінній y до предиката $B(x, y)$ визначає проєкцію множини B на U_Y . Квантор загальності позначається двоїтим виглядом. Геометрично $\forall x B(x, y)$ є проєкція найбільшого циліндра, що лежить в Y на множині U_Y .

Тобто коли розглядаємо квантор існування, фактично шукаємо підмножину властивостей з B , які задовольняють певні умови щодо y , і ця підмножина позначається на просторі U_Y .

З іншого боку, квантор загальності, який позначається на двоїстому вигляді, досліджує загальність деякої властивості для всіх елементів y . Геометрично це можна сприймати як проєкцію найбільшого циліндра, що лежить у Y , на множину U_Y (рис. 2). Це демонструє, як властивості розподіляються або простягаються крізь різні елементи простору Y .

Отже, обидва квантори, існування та загальності, слугують інструментами для аналізу різних аспектів взаємодії між елементами x та y у заданому предикаті $B(x, y)$.

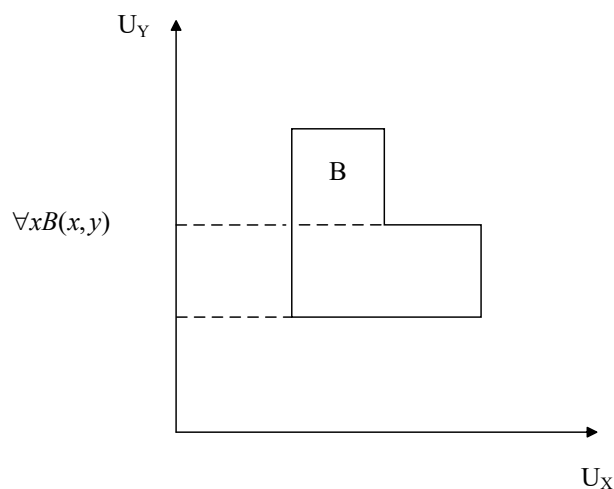


Рис. 2. Графічне подання квантора спільності

У разі багатомісних відношень застосування квантора існування по k змінних $(k < n)n$ – місцевого предиката $P(x_1, \dots, x_n)$, заданого на множині $U_n = U_1 \times U_n$, можна інтерпретувати як проєкцію деякої підмножини множині U_n на множині $U^{n-k} = \prod_{j=1}^{n-k} U_j$. Отже, на множині U^{n-k}

застосуванням квантора існування по k змінних предиката $P(x_1, \dots, x_n)$ позначається $n-k$ місцевий предикат. Цей вираз еквівалентний такому рівнянню (4):

$$\exists(x_{i_1}, \dots, x_{i_k}) P(x_1, \dots, x_n) = Q(x_{i_{n-k}}, \dots, x_n). \quad (4)$$

$$\exists(x_1, \dots, x_{n-1}) P(x_1, \dots, x_n) \wedge R_1(x_1) \wedge \dots \wedge R_{n-1}(x_{n-1}) = Q(x_n), \text{ якщо } R_1(x_1) = 1, \dots, R_{n-1}(x_{n-1}) = 1. \quad (5)$$

Отже, квантор існування може бути розглянутий як лінійний оператор щодо операції диз'юнкції з певною додатковою вимогою (5). Квантор загальності можна характеризувати як лінійний щодо операції кон'юнкції логічний оператор із певним додатковим обмеженням.

Хоча предикати першого порядку мають внутрішні обмеження, що впливають з того, що їх властивості, визначені "неперекривними межами опису, які чітко встановлені та обмежені", сучасні вимоги до інформаційних систем стосуються розширення дескриптивних засобів використовуваних логічних мов [16]. Можемо адаптувати синтаксичні правила мови логіки предикатів першого порядку, щоб додати можливість користування змінюваними предикативними символами. Ця модифікація синтаксису приводить до системи, відомої як логіка предикатів другого порядку, де можна застосовувати як аргументи предикатів не лише терми, а й вислови предикатів першого порядку. Неминуче, що ця нова система має більше дескриптивних засобів, ніж логіка предикатів першого порядку.

Використовуючи логіку предикатів другого порядку, можна встановити зв'язок з алгеброю предикатів і предикатних операцій. Отже, залежно від обраної бази для даних чи знань необхідно обрати відповідну алгебру для запитів та відповідей. Базу даних можна розглядати як машину певного типу

$$(L, U_{\text{зан}}, U_{\text{від}}), \quad (6)$$

де L – множина станів автомата; $U_{\text{зан}}$ – алгебра запитів і $U_{\text{від}}$ – алгебра відповідей. Далі позначається операція $L \times U_{\text{зан}} \rightarrow U_{\text{від}}$, яка має результат у множині відповідей. Якщо $l \in L$ – стан бази даних, $U_{\text{зан}} \in U_{\text{зан}}$ – деякий запит у базі даних, то $l * U_{\text{зан}} = U_{\text{від}}$ є відповіддю на зауважений запит $U_{\text{зан}}$ у належному стані бази. Подання бази даних у вигляді алгебраїчної структури виявляється корисним у різних випадках.

Наприклад, у разі необхідності визначити відхилення композицій та декомпозицій баз даних,

За умови $k = n-1$ наведена рівність є окремим випадком лінійного щодо операції диз'юнкції оператора (5).

Коли $k = n-1$, ця рівність стає окремим випадком лінійного оператора, що діє через операцію диз'юнкції.

а також з ізоморфізму та еквівалентності структур даних, алгебраїчний підхід дозволяє формалізувати ці процеси. Це спрощує аналіз структурних відмінностей між різними схемами даних, дозволяючи автоматизувати процеси виявлення і коригування несумісностей. Завдяки точним алгебраїчним операціям можна ефективно працювати з великими обсягами даних, оптимізуючи процеси інтеграції та синхронізації інформаційних ресурсів.

Реляційна модель даних посідає важливе місце в теорії баз даних [17]. Зазначена модель демонструє переваги алгебраїчного методу подання інформації. Код уперше описав ключові елементи реляційної алгебри, зокрема способи формулювання запитів до баз даних із використанням мов реляційної алгебри [18]. Вона містить операції, такі як селекція, проєкція та теоретико-множинні об'єднання. Однією з ключових особливостей реляційної алгебри є її замкненість: будь-яка її операція генерує нове відношення, до якого можна застосовувати ті самі операції. На відміну від інших мов, де для складних запитів потрібно вводити додаткові структури, у реляційній алгебрі можна легко формулювати запити будь-якої складності без додавання нових конструкцій.

Під час створення реляційних баз даних інформація про тематику подана через взаємодії певної структурної складності. Цей метод відображення даних також ефективний для створення експертних систем різного призначення. Кожен такий взаємозв'язок може бути поданий як окремий предикат, що кодується рядом з нулів і одиниць. Це дозволяє перетворити інформацію із взаємодій у двійкові коди фіксованої довжини.

Велика множина двійкових кодів відтворює наявність глибоких логічних зв'язків між ними. Деякі коди можуть бути виведені з інших за допомогою логічних операцій, показуючи повноту наявної інформації. Для вивчення цих залежностей потрібні вдосконалені математичні методи, щоб коректно описувати відношення між кодами на логіко-алгебраїчному рівні.

У контексті алгебри двійкових кодів основні операції є логічними функціями, що застосовуються до кодів на бітовому рівні. Існує також концепція базових кодів – елементів, з яких можна отримати будь-який інший код з допомогою наявних операцій. Важливим є поняття систем двійкових кодів, які є несприведеними для певних операцій, тобто елементи, які не можна отримати через комбінації інших. Зважаючи на різні підходи до оброблення інформації, із численних перетворень інформації, закодованої у двійковій формі, особливий інтерес мають лінійні перетворення, зосереджені на диз'юнкції або кон'юнкції кодів.

Лінійним щодо операції диз'юнкції перетворенням, визначеним на множині двійкових кодів, є оператор A , що переправляє один код в інший і відповідає таким двом умовам:

$$\begin{cases} A(0) = 0 \\ A(X \vee Y) = A(X) \vee A(Y), \end{cases} \quad (7)$$

де $0 = (0, \dots, 0)$, $X = (x_1, \dots, x_n)$. Відповідно оператор, лінійний щодо кон'юнкції, позначається двоїстим виглядом.

Відповідні умови виписуються в другому вигляді:

$$\begin{cases} A(0) = 0 \\ A(X \vee Y) = A(X) \vee A(Y). \end{cases} \quad (8)$$

За допомогою додавання лінійних операторів до наявних операцій алгебри двійкових кодів ми створили алгебраїчну систему з цікавими характеристиками. Ця алгебра разом із диз'юнктивною, кон'юнктивною та алгеброю постановних дій дає змогу використовувати зрозумілий алгебраїчний мовний інструментарій для формалізованого вираження умов, які має виконувати зазначена система взаємодій. Це особливо корисно, коли інформація, кодована двійковими кодами, проявляє лінійні та узгоджені характеристики.

Дослідження зосереджено на вивченні алгебри лінійних предикатних дій, їх характеристик та методів застосування, а також на методах розв'язання кванторних лінійних рівнянь. Потенціал лінійних предикатних дій у зв'язному висновку для відтворення лінгвістичних закономірностей та виконання завдань розпізнавання та класифікації об'єктів наголошує на важливості цього напрямку дослідження. Методи розв'язання кванторних лінійних рівнянь відіграють вирішальну роль у сфері баз даних. По-перше, квантори дозволяють формулювати глибокі

та деталізовані запити, зважаючи на умови "для всіх" або "існує", що значно розширює можливості взаємодії з інформацією. По-друге, ефективність цих методів сприяє оптимізації процесів вибірки, забезпечуючи швидке й менш ресурсомістке виконання запитів.

Крім того, методи з кванторами сприяють підтримці логічної цілісності в базах даних, гарантуючи, що записи, які відповідають певним умовам, існують або не існують у відповідних секторах. Завдяки високому рівню абстракції цих методів користувачі можуть з легкістю розуміти та інтерпретувати складні логічні умови.

Така гнучкість є незамінною в проєктуванні баз даних, оскільки вона дає змогу розробникам адаптувати систему до різних логічних структур і сценаріїв реального життя. Не менш важливо, що в контексті сучасних інтегрованих рішень зі штучним інтелектом кванторні лінійні рівняння можуть слугувати інструментом для глибокого аналізу даних.

З огляду на ці аспекти методи розв'язання кванторних лінійних рівнянь у базах даних не лише підвищують ефективність і точність запитів, але й сприяють кращому обслуговуванню користувачів, забезпечуючи високу продуктивність систем.

Вирішення завдання

Дані щодо модельованої системи є ключовою інформацією, що допомагає зрозуміти та обробити конкретну задачу. Насамперед необхідно брати до уваги розмірність задачі, яка вказує на складність системи й може містити інформацію про кількість використовуваних параметрів та їх взаємозв'язок.

Семантичні ознаки вказують на властивості вхідного вектора даних. Ці ознаки допомагають системі розпізнавати та інтерпретувати інформацію, передану в модель, забезпечуючи, щоб вхідні дані відтворювали реальність і відповідали вимогам задачі.

Система логічних рівнянь є центром цієї моделі. Вона містить рівняння, що описують взаємозв'язок між різними параметрами, а також умови та обмеження, які мають бути виконані. Ця система слугує основою для аналізу вхідної інформації та знаходження потрібних рішень.

Після того, як вся вхідна інформація буде оброблена програмною системою, ініціюється перевірка на несуперечність системи логічних рівнянь. Це критично важливий крок, оскільки гарантує, що отримані рішення є правильними та обґрунтованими.

Крім того, система визначає та вилучає несуттєві ознаки, що можуть ускладнювати рішення.

На завершення обчислення отримуємо весь масив можливих вирішень системи логічних

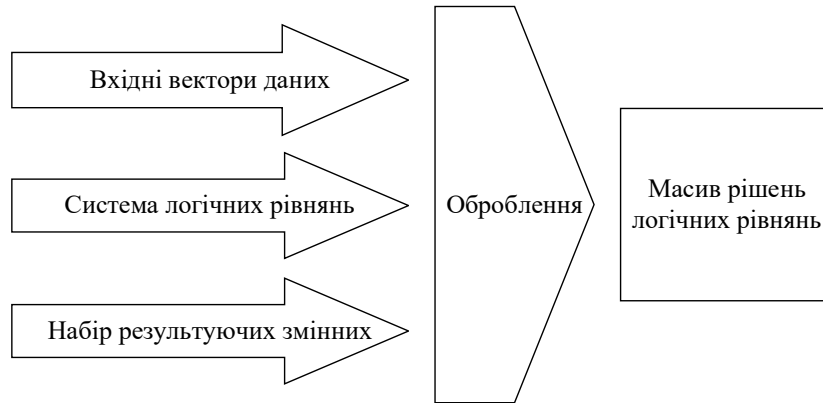


Рис. 3. Об'єктна модель

Ґрунтуючись на теорії лінійних логічних операторів, можемо перейти до етапу побудови алгоритму розв'язання рівнянь. Ця теорія забезпечує необхідний математичний апарат і понятійний базис, що спрощує процес створення алгоритмічної структури.

Нехай потрібно розв'язати таке предикатне рівняння:

$$Q(Y) = \exists X (P(X) \wedge K(Y, X)). \quad (9)$$

Предикати $Q(Y)$ і $P(X)$ задані на ділянці $U = (U_1, \dots, U_n)$, що містить n елементів, бінарний предикат $K(Y, X)$ заданий на ділянці $U \times U$. Потрібно обчислити предикат $P(X)$, якщо відомі предикати $Q(Y)$ і $K(Y, X)$.

З огляду на те, що предикатна змінна X пов'язана квантором існування, рівняння (9) матиме вигляд

$$Q(Y) = \bigvee_{j=1}^n (P(u_j) \wedge K(Y, u_j)). \quad (10)$$

Рівність (10) виконується тільки тоді, коли вона задовольняє умови для всіх можливих значень предикатної змінної Y з множини U . Отже, маємо такі n рівностей:

$$Q(u_i) = \bigvee_{j=1}^n (P(u_j) \wedge K(u_i, u_j)) \quad (11)$$

для будь-якого $i \in 1, \dots, n$.

Значення предикатів $Q(u_i)$ і $P(u_j)$ позначимо відповідно y_i і x_j , де $y_i, x_j \in \{0, 1\}$ і $i, j \in 1, \dots, n$.

рівнянь (рис. 3). Ці рішення подаються у вигляді таблиць, графіків або в будь-якій іншій формі, що найкраще відповідає конкретній задачі та потребам користувача.

Значення бінарного предиката $K(u_i, u_j)$ позначимо як $k_{ij} \in \{0, 1\}, i, j \in 1, \dots, n$. Зважаючи на додані позначки, рівність (11) матиме вигляд

$$y_i = \bigvee_{j=0}^n (x_j \wedge k_{ij}). \quad (12)$$

Відомо, що якщо для довільних предикатів $P(t)$ і $Q(t)$ виконується співвідношення $\psi: P(t) \rightarrow X$, тоді правильно також і $\psi: (P(t) \vee Q(t)) \rightarrow X \vee Y$. Звідси здобуємо операторне рівняння вигляду

$$K(X) = Y, \quad (13)$$

де K – лінійний логічний оператор, заданий на просторі E_{\vee}^n , з матрицею оператора

$$K = \begin{pmatrix} k_{11} & \dots & \dots & \dots & k_{1n} \\ \vdots & \ddots & \dots & \ddots & \vdots \\ k_{i1} & \dots & k_{ij} & \dots & k_{in} \\ \vdots & \ddots & \dots & \ddots & \vdots \\ k_{n1} & \dots & \dots & \dots & k_{nn} \end{pmatrix}. \quad (14)$$

Отже, предикатне рівняння (9) еквівалентне операторному (13), визначеному на логічному просторі E_{\vee}^n . Відповідно до ідеї про суцільний тип матриці лінійно-логічного постійного оператора, що діє з простору E_{\vee}^n , для зворотності необхідно й достатньо, щоб у кожному рядку й стовпці матриці такого оператора був один і тільки один елемент, який дорівнює одиниці. Якщо матриця (14) задовольняє зазначені вище умови ідеї, то розв'язання рівняння (13) буде таким:

$$X = K^{-1}(Y). \quad (15)$$

Матриця зворотного оператора збігається із транспонованою матрицею оператора K . Отже, розв'язання операторного рівняння (15) у матричному типі буде таким:

$$X = RT \times Y. \quad (16)$$

Після розв'язання предикатне рівняння (9) можна записати у вигляді

$$P(X) = \exists Y(Q(Y) \wedge K(X, Y)). \quad (17)$$

У разі, якщо оператор K не є регулярним, розв'язок предикатного рівняння записати у вигляді (17) не можна, однак, скориставшись алгебраїчним записом предикатного рівняння (9), будемо шукати розв'язок рівняння в такий спосіб.

Операторне рівняння (13) напишемо у вигляді системи логічних рівнянь. Припустимо, що вектор Y не одиничний.

$$\begin{cases} \bigvee_{j=1}^n (k_{1j} \wedge x_j) = y_1 \\ \bigvee_{j=1}^n (k_{ij} \wedge x_j) = y_i \\ \bigvee_{j=1}^n (k_{nj} \wedge x_j) = y_n \end{cases} \quad (18)$$

Нехай одиниці розташовані в Y на місцях $(d_1, \dots, d_{y^{(i)}}) = D$, а нулі на місцях $(z_1, \dots, z_{y^{(i)}}) = Z$, $D \cap Z = \emptyset$, $D \cup Z = N$, $N = (1, \dots, n)$. Множина індексів, на яких елементи вектора X дорівнюють нулю, позначимо як $L = (l_1, \dots, l_{x^{(i)}})$. Символом * позначатимемо місця, на яких можуть бути нулі або одиниці. Символи * розташовані на місцях $(m_1, \dots, m_{x^{(i)}}) = M$.

Наведемо узагальнені кроки методу.

1. Ініціалізація. $i := z_1$.

2. Створюємо множину з координат вектора X , які дорівнюють нулю.

2.1. $j := 1$.

2.2 Якщо $K[i, j] = 1$, то $X[j] := l_1$.

2.3 Організуємо перебір індексів j від 0 до n .

3. Індексу i прирівнюється наступний елемент із множини Z і перехід до п. 2.1 доти, доки не будуть обрані всі елементи множини Z .

4. Формуємо множину M . Отримуємо деякий логічний вектор X , що містить нулі та символи *.

5. Перевірка системи (18) на несуперечність.

5.1 Підставляємо знайдений вектор у систему.

5.2 Організуємо розв'язання отриманої системи згідно з формулою $\bigvee_{j=1}^n (k_{ij} \wedge x_j) = y_i$.

5.3 Якщо система не сумісна, то вектор не є розв'язком системи.

6. Формування розв'язання системи.

6.1 У вектор $X^{(*)}$ підставляємо замість першого символу * одиницю, а замість інших символів нулі.

6.2 Перехід до п. 5.

6.3 Якщо сформований логічний вектор є розв'язком системи, запам'ятовуємо його в масив вирішень.

6.4 Організуємо різні підстановки нулів і одиниць замість символів * з переходом у разі кожної нової комбінації до п. 5.

7. Випишемо всі отримані розв'язки системи, якщо масив вирішень не порожній. А якщо ні, то результат повідомлення про суперечливість системи.

Приклад

Наведений приклад ілюструє можливість використання теорії лінійно-логічних операторів і методу розв'язання кванторного предикатного рівняння для оброблення та зберігання інформації в базах даних. Припустимо, що база даних розміщує інформацію про чотири фабрики, що виготовляють деталі для автомобілів. Нехай фабрика f_1 виготовляє деталі d_1 і d_2 , фабрика f_2 – деталі d_2 і d_3 , фабрика f_3 – деталі d_1 і d_4 і фабрика f_4 виготовляє деталі d_3 і d_4 .

Зв'язок між фабриками та деталями може бути поданий за допомогою такого бінарного предиката (19):

$$P_1(f, d) = \begin{cases} 1, \text{ якщо фабрика } f \text{ виготовляє деталь } d, \\ 0 \text{ в іншому разі,} \end{cases} \quad (19)$$

де $f \in \{f_1, \dots, f_4\}$ і $d \in \{d_1, \dots, d_4\}$.

Отже, інформацію про фабрики можна подати за допомогою формули предиката $P_1(f, d)$. Якщо потрібно визначити, які фабрики виготовляють деталь d_1 , відповідний предикат, який відтворює цей запит, може бути записаний таким чином:

$$P_2(d) = \begin{cases} 1, \text{ якщо } d = d_1, \\ 0 \text{ в іншому разі.} \end{cases} \quad (20)$$

Як результат, предикат $P_3(f)$, що відповідає шуканому запиту, подається у формі кванторного рівняння:

$$x \exists d P_1(f, d) \wedge P_2(d) = P_3(f) \quad (21)$$

і задає такий взаємозв'язок:

$$P_3(f) = \begin{cases} 1, & \text{якщо фабрика } f \text{ виготовляє деталь } d_1, \\ 0 & \text{в іншому разі.} \end{cases} \quad (22)$$

Розв'язок зазначеного кванторного предикатного рівняння впливає з розв'язування відповідного операторного рівняння $A * X = Y$ у лінійно-логічному просторі E_n . Матриця оператора A має такий вигляд:

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad (23)$$

а вектор $X = (1 \ 0 \ 0 \ 0)$. Унаслідок застосування оператора A до вектора X отримуємо вектор Y зі значеннями $(1 \ 0 \ 1 \ 0)$, що вказує на те, що фабрики f_1 та f_3 виготовляють деталь d_1 . Отже, пошук потрібної інформації у базі даних трансформується в дію операторного множення. Далі, якщо нам потрібно визначити, які фабрики виготовляють деталі d_1 і d_3 , використовуючи адитивний характер лінійного логічного оператора A , отримаємо:

$$A * X_1 \vee A * X_3 = A(X_1 \vee X_3) = A * X_4. \quad (24)$$

$$K_1(m, d) = \begin{cases} 1, & \text{якщо в автомобілі } m \text{ використовується деталь } d, \\ 0 & \text{в іншому разі,} \end{cases} \quad (26)$$

де $m \in \{m_1, \dots, m_4\}$ і $d \in \{d_1, \dots, d_4\}$. Подібно до попередньої ситуації, із бази даних можна легко визначити, в яких автомобілях використовуються певні деталі, якщо застосувати рішення відповідного кванторного рівняння та перетворити його на операторне рівняння. Тоді маємо такий вираз:

$$\exists d K_1(m, d) \wedge K_2(m) = K_3(d), \quad (27)$$

де унарні предикати $K_2(m)$ і $K_3(d)$ задають, відповідно, конкретний автомобіль і конкретну деталь. Розкриваючи зазначене рівняння за допомогою предикатів $K_2(m)$ чи $K_3(d)$, зможемо отримати потрібну інформацію з бази даних. Розглянемо більш складне завдання: визначити, які фабрики виготовляють компоненти для конкретного

Вектори X_1 та X_3 формуються на основі предикатів, що відповідають деталям d_1 та d_3 . Тому відповіді на комплексні запити в базі даних впливають з розв'язку операторних рівнянь. Застосовуючи алгоритм розв'язання таких рівнянь, описаний у попередній секції, можна визначити, які деталі виготовляють певні фабрики. Наприклад, нам потрібно з'ясувати, які деталі виготовляє фабрика f_2 . Тоді логічний вектор Y дорівнює $(0 \ 1 \ 0 \ 0)$. Розв'язуючи операторне рівняння такого типу, отримаємо

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \quad (25)$$

Щодо X маємо вектор $(0 \ 1 \ 0 \ 0)$ і $(0 \ 0 \ 1 \ 0)$. Отже, фабрика f_2 виготовляє деталі d_2 і d_3 .

Згадаймо, що в базі даних зберігається інформація про деталі, що встановлюються в різних автомобілях. Як приклад, в автомобілі m_1 застосовується деталь d_2 , в автомобілі m_2 встановлені деталі d_2 та d_3 , в автомобілі m_3 – також деталі d_2 і d_3 , а в m_4 – деталі d_3 і d_4 . Ця залежність між автомобілем та деталями подана за допомогою бінарного предиката $K_1(m, d)$, що визначений так (25):

автомобіля. Для цього завдання маємо такий набір кванторних предикатних рівнянь:

$$\begin{cases} \exists d P_1(f, d) \wedge P_2(d) = P_3(f), \\ \exists d K_1(m, d) \wedge K_2(m) = P_2(d). \end{cases} \quad (28)$$

$$\text{Цю систему відтворимо у вигляді рівняння} \\ \exists d P_1(f, d) \wedge (\exists d K_1(m, d) \wedge K_2(m)) = P_3(f). \quad (29)$$

Кванторному предикатному рівнянню (27) відповідає операторне рівняння вигляду

$$B * T = X \quad (30)$$

у лінійно-логічному просторі E_n . Логічні вектори T і X побудовані відповідно до бінарних предикатів $K_2(m)$ і $K_3(d)$.

Лінійний логічний оператор Y має матрицю такого вигляду:

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (31)$$

побудовану за бінарним предикатом $K_1(m, d)$. Отже, предикатному рівнянню (28) відповідає операторне рівняння вигляду

$$A(B(T)) = X. \quad (32)$$

Отримане рівняння перетворимо на такий вигляд:

$$C(T) = X, \quad (33)$$

де лінійний логічний оператор C дорівнює суперпозиції операторів A і B . У цьому разі маємо

$$C = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}. \quad (34)$$

Отже, пошук значної кількості інформації в базі даних можна спростити до матричного обчислення оператора I_3 , використовуючи операцію множення матриць операторів A та B . Розглянемо, наприклад, фабрики, що виготовляють деталі для автомобіля m_1 . Кореспондуючий логічний вектор T поданий у вигляді $(1 \ 0 \ 0 \ 0)$. Отже маємо

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \quad (35)$$

Отже, виробниками деталей для автомобіля m_1 є фабрики f_1 і f_2 .

Результати дослідження

Під час дослідження розроблено метод, оснований на застосуванні кванторних лінійних рівнянь та теорії лінійно-логічних операторів для формування запитів у базах даних. Розроблено комплексну математичну модель, яка дає змогу кодифікувати запити до баз даних у формі кванторних лінійних рівнянь, що значно розширює межі можливостей оброблення даних. Запропонована модель виявилася здатною ефективно маніпулювати як простими, так і складними запитам.

Метод застосування кванторних логічних лінійних рівнянь для створення запитів у базах даних може бути адаптованим до конкретних потреб і умов використання. Перелічимо деякі з можливих підходів.

1. Розроблення спеціалізованого програмного забезпечення: створення програмних інструментів або модулів, що інтегруються із сучасними системами управління базами даних (СУБД) для оброблення та оптимізації кванторних логічних лінійних запитів. Це може передбачати розроблення інтерфейсів для легкого формування таких запитів користувачами.

2. Упровадження мов високого рівня: розширення наявних мов запитів, зокрема *SQL*, новими конструкціями, що дають змогу формулювати кванторні логічні лінійні рівняння безпосередньо. Це може потребувати додаткового компілятора або інтерпретатора, що транслюватиме ці розширення у виконувани запити.

3. Інтеграція з вебсервісами: розроблення вебсервісів або *API*, що дають змогу користувачам надсилати кванторні логічні лінійні рівняння як запити до бази даних з допомогою інтернету. Це може забезпечити більш широкий доступ до функціоналу для розробників, які працюють із різними платформами.

Застосування цих методів вимагатиме детального розуміння особливості домену, де вони використовуватимуться, а також глибоких знань у галузі баз даних і програмування. Однак потенціал, який вони відкривають для оптимізації та автоматизації оброблення інформації, робить їх дуже перспективними для подальшого розвитку та впровадження.

4. Аналітика даних: застосування для складних аналітичних запитів у великих базах даних, зокрема для фінансового аналізу, наукових досліджень і соціальних мереж. Методику можна використовувати для виявлення складних взаємозв'язків та закономірностей в інформації.

5. Штучний інтелект та машинне навчання: інтеграція із системами ШІ для формування та оптимізації запитів до баз даних у процесі машинного навчання. Це може поліпшити якість інформації, що використовується для тренування моделей, забезпечуючи більш точні прогнози та аналітику.

6. Розроблення прикладних програм: використання в спеціалізованих програмних застосунках, наприклад, у системах електронного урядування, медичних інформаційних системах або системах управління ресурсами підприємства (*ERP*), для забезпечення більш ефективного доступу до інформації та її оброблення.

7. Оптимізація внутрішніх процесів: покращення внутрішніх процесів у компаніях завдяки оптимізації запитів до корпоративних баз даних, що може зменшити витрати часу на оброблення інформації та підвищення продуктивності.

Важливою особливістю запропонованого методу, на відміну від традиційних підходів до виконання запитів до баз даних, є його здатність забезпечувати більшу адаптивність у формулюванні запитів, використовуючи складні логічні операції та квантори. Така гнучкість у поєднанні з ефективністю оброблення суттєво знижує часові витрати на оброблення запитів.

Застосування розробленого методу має великий потенціал у багатьох сферах, що вимагають швидкого та точного аналізу значних обсягів інформації, зокрема фінансовий аналіз, наукове дослідження та управління складними інформаційними системами тощо.

Досягнуті результати відкривають перспективи для подальшого розвитку напряму дослідження, зокрема щодо вдосконалення наявних алгоритмів розв'язання кванторних логічних лінійних рівнянь, що може сприяти поліпшенню ефективності та універсальності запропонованої методики у розв'язанні задач оброблення даних.

Висновки

У роботі проаналізовано формальні способи інтелектуальних систем: способи подання знань залежно від конкретних галузей застосування систем; формальні мови, що дозволяють подавати знання в пам'яті комп'ютера. Розглянуто сфери впровадження алгебраїчних методів в автоматизованих системах. Встановлено принципи застосування абстрактної алгебри у сферах, пов'язаних з базами даних та розробкою інтелектуальних систем. На основі впровадження алгебраїчних методів розроблено метод використання лінійно-логічних операторів і логічних рівнянь з метою видобування інформації.

Список літератури

1. Ahmad A. Y. A. B., Kumari D. K., Shukla A., Deepak A., Chandnani M., Pundir S., Shrivastava A. Framework for Cloud Based Document Management System with Institutional Schema of Database. *International Journal of Intelligent Systems and Applications in Engineering*. 2024. Vol. 12, No. 3s. P. 672–678. URL: <https://ijisae.org/index.php/IJISAE/article/view/3853> (дата звернення: 12.03.2024).

Показано перспективність упровадження запропонованого методу використання логічних рівнянь в інформаційних системах, зокрема в базах даних. Він забезпечує можливість отримання не тільки інформації, що безпосередньо зберігається в базі даних, але й похідної інформації, отриманої на основі базисної. Задача отримання похідної інформації безпосередньо пов'язана із задачею результату в інтелектуальних системах, водночас у разі застосування алгебраїчного підходу до опису похідної інформації виокремлюється певна алгебраїчна система – алгебра запитів, у термінах якої похідна інформація записується через базисну.

Для того щоб довільний запит можна було виражати яким-небудь видом через базисні запити, на множини запитів додаються алгебраїчні операції, що дозволяють оперувати із запитами. Аналогічно подібні алгебраїчні операції мають додаватися й на множини відповідей. У цьому разі способами наявних алгебраїчних операцій відповідь на довільний запит можна обчислити відповідно до структури запиту, записати його через запити, відповіді на які вже відомі. Розглянуті в роботі множини запитів і відповідей є алгебрами запитів і відповідей.

Застосування алгебраїчних структур значно спрощує процес інтеграції різноманітних джерел інформації, даючи змогу ефективніше керувати складною інформацією. Це сприяє підвищенню якості інформаційної взаємодії з базами знань, забезпечує гнучкішу та швидшу відповідь на запити. Крім того, використання алгебраїчного підходу дозволяє формалізувати та оптимізувати процедури оброблення інформації, що особливо актуально для сфер, яким властиві значні обсяги даних або високі вимоги до точності інформаційних запитів. Нарешті, подальший розвиток алгебраїчних методів і моделей в інформаційних системах може сприяти створенню нових поколінь інтелектуальних аналітичних систем, здатних на глибоке розуміння та оброблення інформації, що відкриває нові перспективи для досліджень у цій галузі.

2. Yang X., Guan X., Pang Z., Kui X., Wu H. GridMesa: A NoSQL-based big spatial data management system with an adaptive grid approximation model. *Future Generation Computer Systems*. 2024. Vol. 155. P. 324–339. DOI: <https://doi.org/10.1016/j.future.2024.02.010>
3. Taipalus T. Vector database management systems: Fundamental concepts, use-cases, and current challenges. *Cognitive Systems Research*. 2024. Vol. 85. 13 p. DOI: <https://doi.org/10.1016/j.cogsys.2024.101216>
4. Davydovskiy M. Storing of Lua tables in relational databases. *AIP Conference Proceedings*. 2023. DOI: <https://doi.org/10.1063/5.0132449>
5. Aishwarya V. A Prompt Engineering Approach for Structured Data Extraction from Unstructured Text Using Conversational LLMs. *ACM International Conference Proceeding Series*. 2023. P. 183–189. DOI: <https://doi.org/10.1145/3639631.3639663>
6. Aebeloe C., Montoya G., Hose K. Optimizing SPARQL queries over decentralized knowledge graphs. *Semantic Web*. 2023. Vol. 14, No. 6. P. 1121–1165. DOI: <https://doi.org/10.3233/SW-233438>
7. Braun C. H. J., Käfer T. Quantifiable integrity for Linked Data on the web. *Semantic Web*. 2023. Vol. 14, No. 6. P. 1167–1207. DOI: <https://doi.org/10.3233/SW-233409>
8. Faltín T., Trigonakis V., Berdai A., Fusco L., Iorgulescu C., Lee J., Yaghob J., Hong S., Chafi H. Distributed Asynchronous Regular Path Queries (RPQs) on Graphs. *Middleware Industrial Track 2023 – Proceedings of the 2023 24th International Middleware Conference Industrial Track, Part of: Middleware 2023*. 2023. P. 35–41. DOI: <https://doi.org/10.1145/3626562.3626833>
9. Iglesias-Molina A., Toledo J., Corcho O., Chaves-Fraga D. Re-Construction Impact on Metadata Representation Models. *K-CAP 2023 – Proceedings of the 12th Knowledge Capture Conference 2023*. 2023. P. 197–205. DOI: <https://doi.org/10.1145/3587259.3627554>
10. Zykin S.V. Testing Dependencies and Inference Rules in Databases. *Automatic Control and Computer Sciences*. 2023. Vol. 57, No. 7. P. 788–802. DOI: <https://doi.org/10.3103/S0146411623070179>
11. Sathesh A., Kumar A. An Object-Oriented Database Design for Effective Classification. *International Journal of Intelligent Systems and Applications in Engineering*. 2022. Vol. 10, No. 4. P. 111–119. URL: <https://ijisae.org/index.php/IJISAE/article/view/2204> (дата звернення: 12.03.2024).
12. Figallo M., Monica-Gomes C. The Subalgebra Lattice of a Finite Diagonal-Free Two-Dimensional Cylindric Algebra. *Computación y Sistemas*. 2023. Vol. 27, No. 1. DOI: <https://doi.org/10.13053/cys-27-1-4544>
13. Yang T., Wang Y., Sha L., Engelbrecht, J. Knowledgebra: An Algebraic Learning Framework for Knowledge Graph. *Machine Learning and Knowledge Extraction*. 2022. Vol. 4, No. 2. P. 432–445. DOI: <https://doi.org/10.3390/make4020019>
14. Gilray T., Kumar S. Distributed Relational Algebra at Scale. *2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, Hyderabad, India, 17–20 December 2019. 2019. DOI: <https://doi.org/10.1109/hipc.2019.00014>
15. Luo S., Gao Z.J., Gubanov M., Perez L. L. and Jermaine C. Scalable Linear Algebra on a Relational Database System. *IEEE Transactions on Knowledge and Data Engineering*. 2019. Vol. 31, No. 7. P. 1224–1238. DOI: <https://doi.org/10.1109/tkde.2018.2827988>
16. Shubin I., Kozyriev A., Liashik V., Chetverykov G. Methods of adaptive knowledge testing based on the theory of logical networks. *CEUR Workshop Proceedings*. 2021. P. 1184–1193. URL: <https://ceur-ws.org/Vol-2870/paper86.pdf> (дата звернення: 12.03.2024).
17. Harrington J.L. *Relational Database Design and Implementation: Fourth Edition*. Elsevier Inc., 2016. 689 p. DOI: <https://doi.org/10.1016/C2015-0-01537-4>
18. Meijer E., Bierman G. A co-relational model of data for large shared data banks. *Communications of the ACM*. 2011. Vol. 54, No. 4. P. 49–58. DOI: <https://doi.org/10.1145/1924421.1924436>

References

1. Ahmad, A.Y. A. B., Kumari, D.K., Shukla, A., Deepak, A., Chandnani, M., Pundir, S., Shrivastava, A. (2024), "Framework for Cloud Based Document Management System with Institutional Schema of Database". *International Journal of Intelligent Systems and Applications in Engineering*. No.12(3s), P. 672–678, available at: <https://ijisae.org/index.php/IJISAE/article/view/3853> (last accessed 12.03.2024).

2. Yang, X., Guan, X., Pang, Z., Kui, X., Wu, H. (2024), "GridMesa: A NoSQL-based big spatial data management system with an adaptive grid approximation model". *Future Generation Computer Systems*. Vol 155, P. 324–339. DOI: <https://doi.org/10.1016/j.future.2024.02.010>
3. Taipalus, T. (2024), "Vector database management systems: Fundamental concepts, use-cases, and current challenges". *Cognitive Systems Research*. Vol. 85. 13 p. DOI: <https://doi.org/10.1016/j.cogsys.2024.101216>
4. Davydovskiy, M. (2023), "Storing of Lua tables in relational databases". In: *AIP Conference Proceedings*. DOI: <https://doi.org/10.1063/5.0132449>
5. Aishwarya, V. (2023), "A Prompt Engineering Approach for Structured Data Extraction from Unstructured Text Using Conversational LLMs". In: *ACM International Conference Proceeding Series*. P. 183–189. DOI: <https://doi.org/10.1145/3639631.3639663>
6. Aebeloe, C., Montoya, G., Hose, K. (2023), "Optimizing SPARQL queries over decentralized knowledge graphs". *Semantic Web*. No. 14(6), P. 1121–1165. DOI: <https://doi.org/10.3233/SW-233438>
7. Braun, C. H. J., Käfer, T. (2023), "Quantifiable integrity for Linked Data on the web". *Semantic Web*. No. 14(6), P. 1167–1207. DOI: <https://doi.org/10.3233/SW-233409>
8. Faltín, T., Trigonakis, V., Berdai, A., Fusco, L., Iorgulescu, C., Lee, J., Yaghob, J., Hong, S., Chafi, H. (2023), "Distributed Asynchronous Regular Path Queries (RPQs) on Graphs". In: *Middleware Industrial Track 2023 – Proceedings of the 2023 24th International Middleware Conference Industrial Track, Part of: Middleware 2023*. P. 35–41. DOI: <https://doi.org/10.1145/3626562.3626833>
9. Iglesias-Molina, A., Toledo, J., Corcho, O., Chaves-Fraga, D. (2023), "Re-Construction Impact on Metadata Representation Models". In: *K-CAP 2023 – Proceedings of the 12th Knowledge Capture Conference 2023*. P. 197–205. DOI: <https://doi.org/10.1145/3587259.3627554>
10. Zykin, S.V. (2023), "Testing Dependencies and Inference Rules in Databases". *Automatic Control and Computer Sciences*. 57(7), P. 788–802. DOI: <https://doi.org/10.3103/S0146411623070179>
11. Satheesh, A., Kumar, A. (2022), "An Object-Oriented Database Design for Effective Classification". *International Journal of Intelligent Systems and Applications in Engineering*. No.10(4), P. 111–119, available at: <https://ijisae.org/index.php/IJISAE/article/view/2204> (last accessed 12.03.2024).
12. Figallo, M., Monica-Gomes, C. (2023), "The Subalgebra Lattice of a Finite Diagonal-Free Two-Dimensional Cylindric Algebra". *Computación y Sistemas*. No. 27(1). DOI: <https://doi.org/10.13053/cys-27-1-4544>
13. Yang, T., Wang, Y., Sha, L., Engelbrecht, J., Hong, P. (2022), "Knowledgebra: An Algebraic Learning Framework for Knowledge Graph". *Machine Learning and Knowledge Extraction*. No. 4(2), P. 432–445. DOI: <https://doi.org/10.3390/make4020019>
14. Gilray, T., Kumar, S. (2019), "Distributed Relational Algebra at Scale". In: *2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC), 17–20 December 2019, Hyderabad, India*. IEEE. DOI: <https://doi.org/10.1109/hipc.2019.00014>
15. Luo, S., Gao, Z.J., Gubanov, M., Perez, L.L., Jermaine, C. (2019), "Scalable Linear Algebra on a Relational Database System". *IEEE Transactions on Knowledge and Data Engineering*. No. 31(7), P. 1224–1238. DOI: <https://doi.org/10.1109/tkde.2018.2827988>
16. Shubin, I., Kozyriev, A., Liashik, V., Chetverykov, G. (2021), "Methods of adaptive knowledge testing based on the theory of logical networks". *CEUR Workshop Proceedings*. CEUR-WS. P. 1184–1193, available at: <https://ceur-ws.org/Vol-2870/paper86.pdf> (last accessed 12.03.2024).
17. Harrington, J.L. (2016), "Relational Database Design and Implementation: Fourth Edition". *Elsevier Inc*. 689 p. DOI: <https://doi.org/10.1016/C2015-0-01537-4>
18. Meijer, E., Bierman, G. (2011), "A co-relational model of data for large shared data banks". *Communications of the ACM*. No. 54(4), P. 49–58. DOI: <https://doi.org/10.1145/1924421.1924436>

Надійшла 15.03.2024

Відомості про авторів / About the Authors

Козирев Андрій Дмитрович – Харківський національний університет радіоелектроніки, аспірант кафедри програмної інженерії, Харків, Україна; e-mail: andrii.kozyriev@nure.ua; ORCID ID: <https://orcid.org/0000-0001-6383-5222>

Шубін Ігор Юрійович – кандидат технічних наук, доцент, Харківський національний університет радіоелектроніки, професор кафедри програмної інженерії, Харків, Україна; e-mail: igor.shubin@nure.ua; ORCID ID: <https://orcid.org/0000-0002-1073-023X>

Kozyriev Andrii – Kharkiv National University of Radio Electronics, Postgraduate Student at the Software Department, Kharkiv, Ukraine.

Shubin Ihor – PhD (Engineering Sciences), Professor at the Software Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

METHOD OF LINEAL-LOGIC OPERATORS AND LOGICAL EQUATIONS IN INFORMATION EXTRACTION TASKS

Relational and logical methods of knowledge representation play a key role in creating a mathematical basis for information systems. Predicate algebra and predicate operators are among the most effective tools for describing information in detail. These tools make it easy to formulate formalized information, create database queries, and simulate human activity. In the context of the new need for reliable and efficient data selection, a problem arises in deeper analysis. **Subject of the study** is the theory of quantum linear equations based on the algebra of linear predicate operations, the formal apparatus of linear logic operators and methods for solving logical equations in information extraction tasks. **Aim of the study** is a developing of a method for using linear logic operators and logical equations to extract information. This approach can significantly optimize the process of extracting the necessary information, even in huge databases. **Main tasks:** analysis of existing approaches to information extraction; consideration of the theory of linear logic operators; study of methods for reducing logic to an algebraic form; analysis of logical spaces and the algebra of finite predicate actions and the theory of linear logic operators. **The research methods** involve a systematic analysis of the mathematical structure of the algebra of finite predicates and predicate functions to identify the key elements that affect the query formation process. The method of using linear logic operators and logical equations for information extraction is proposed. **The results** of the study showed that the method of using linear logic operators and logical equations is a universal and adaptive tool for working with algebraic data structures. It can be applied in a wide range of information extraction tasks and prove its value as one of the possible methods of information processing. **Conclusion.** The paper investigates formal methods of intelligent systems, in particular, ways of representing knowledge in accordance with the peculiarities of the field of application and the language that allows encoding this knowledge for storage in computer memory. The proposed method can be implemented in the development of language interfaces for automated information access systems, in search engine algorithms, for logical analysis of information in databases and expert systems, as well as in performing tasks related to object recognition and classification.

Keywords: knowledge bases; intelligent systems; algebra of finite predicates; logical operators; quantile linear equations.

Бібліографічні описи / Bibliographic descriptions

Козирев А. Д., Шубін І. Ю. Метод лінійно-логічних операторів та логічних рівнянь у завданнях видобування інформації. *Сучасний стан наукових досліджень та технологій в промисловості*. 2024. № 1 (27). С. 81–95. DOI: <https://doi.org/10.30837/ITSSI.2024.27.081>

Kozyriev, A., Shubin, I. (2024), "The method of linear-logical operators and logical equations in information extraction tasks", *Innovative Technologies and Scientific Solutions for Industries*, No. 1 (27), P. 81–95. DOI: <https://doi.org/10.30837/ITSSI.2024.27.081>