

О. КАРАТАЄВ, Д. СИТНИКОВ

МЕТОД ПОВТОРНОГО ВИКОРИСТАННЯ ЗНАТЬ У ФОРМІ ЛОГІЧНИХ РІВНЯНЬ

Предмет дослідження – процеси побудови та поповнення баз знань, повторного використання знань, і створення програмних систем на основі баз знань, інтерпретація знань як один із підходів до повторного їх застосування, що полягає у виведенні нових знань на основі наявних фактів у базі знань. **Мета дослідження** – розроблення методу повторного використання знань за допомогою вирішення логічних рівнянь скінченних предикатів для певної предметної галузі. Для досягнення поставленої мети визначено такі **завдання**: дослідити описативні підходи до логічного моделювання предметної галузі, що дають змогу повторного використання знань, заданих системою логічних рівнянь у межах алгебри скінченних предикатів; розробити метод поповнення бази знань у формі предикатних рівнянь за допомогою вилучення змінних із логічних систем, де рівняння є складними логічними зв'язками між дискретними ознаками об'єктів або процесів. Використано такі **методи**: алгебра скінченних предикатів, кванторні операції з предикатами для інтерпретації знань. Здобуто такі **результати**: проаналізовано описативні методи повторного використання знань; запропоновано метод поповнення бази знань у формі логічних рівнянь з метою спрощення подальшого використання неявних зв'язків між дискретними ознаками; розглянуто приклад поповнення бази знань медичної галузі, що дає змогу повторного використання знань, заданих неявно. **Висновки**. Запропонований метод дає змогу поповнювати базу знань у формі логічних рівнянь способом додавання предикатних рівнянь, що пов'язують окремі набори дискретних змінних, які цікавлять ученого або практика. Метод поповнення бази знань оснований на додаванні нових спрощених рівнянь; спрощені предикатні рівняння моделюють логічні закономірності, що неявно містяться в основній базі знань; проведено експериментальне дослідження.

Ключові слова: програмна інженерія; бази знань; повторне використання знань; алгебра скінченних предикатів; логічні рівняння.

Вступ

Повторне використання знань є актуальним питанням, що стосується процесу побудови та поповнення баз знань на основі аналізу та перетворення наявних знань. Однак повторного використання знань не відбувається або принаймні бажані результати не досягаються. Зазвичай припускають, що як тільки знання будуть належним чином збережені, їх повторне застосування відбудеться автоматично. Отже, питання, на яке потрібно відповісти, полягає в тому, як можна створювати бази знань, щоб їх надалі ефективно використовувати. Для цього необхідно проаналізувати системи повторного застосування знань, запропонувати підхід щодо створення бази знань та оцінити запропоноване рішення.

Проблема повторного використання знань щодо предметної галузі, де багато дискретних ознак пов'язані складними логічними залежностями, які важко розглянути в явній формі, полягає в тому, щоб використати неявні зв'язки між дискретними предикатними змінними для подання цільових логічних зв'язків у явній формі з метою спрощення наявних логічних залежностей. Це має дати змогу

багато разів використовувати такі спрощені залежності без необхідності аналізувати складну систему з несуттєвими дискретними змінними.

Мета дослідження – розроблення методу повторного використання знань за допомогою вирішення логічних рівнянь для певної предметної галузі. Метод поповнення бази знань має бути оснований на додаванні нових спрощених рівнянь, що неявно містяться в основній базі знань; дослідити процес спрощення повторного використання знань завдяки додаванню більш простих рівнянь до бази знань. Для досягнення поставленої мети мають бути виконані такі **завдання**: дослідити описативні підходи до логічного моделювання предметної галузі, що дають змогу повторного використання знань, заданих системою логічних рівнянь у межах алгебри скінченних предикатів, що є узагальненням алгебри логіки; змінні, які містить система логічних рівнянь, мають приймати значення зі скінченних алфавітів; розробити метод поповнення бази знань у формі предикатних рівнянь за допомогою вилучення змінних із логічних систем, де рівняння є складними логічними зв'язками між дискретними ознаками об'єктів або процесів; метод поповнення бази знань має бути оснований на додаванні нових спрощених

рівнянь, отриманих дедуктивним методом із застосуванням кванторних операцій; спрощені предикатні рівняння мають моделювати логічні закономірності, що неявно містяться в основній базі знань; дослідити на прикладі медичної діагностики, де класифікація групи ризиків спрощується завдяки додаванню більш простих рівнянь до бази знань, полегшуючи процес повторного використання знань, заданих неявно системою предикатних рівнянь.

Аналіз проблеми

Як гілка символічного штучного інтелекту, подання знань і логічне виведення спрямовані на проектування програмного забезпечення систем, що інтерпретують подання світу, схоже на інтерпретацію людини. Системи, основані на знаннях, мають обчислювальну модель предметної галузі, де символи відіграють роль артефактів реального світу, таких як фізичні об'єкти, події та відносини. Предметна галузь може охоплювати будь-яку частину реального світу або гіпотетичну систему, щодо якої є бажання подати та інтерпретувати знання для цілей проектування відповідних програмних систем.

У багатьох сучасних дослідницьких роботах аналізуються теоретичні та практичні аспекти окресленої теми [1, 2]. Широкий спектр можливостей повторного використання знань для підвищення ефективності розроблення програмного забезпечення запропоновано в роботах [3, 4]. Оскільки повторне використання знань є одним із найважливіших етапів процесу управління знаннями, у літературі можна знайти кілька досліджень про цей етап [5]. Компанії в усьому світі витрачають час, зусилля та гроші, намагаючись залучити найефективніші системи управління знаннями. Розглядаються не лише програмні засоби, а й культурні та стратегічні аспекти.

Повторне використання знань в інформаційних системах часто описується дескриптивними логічними мовами [6, 7]. Дескриптивний підхід насамперед належить до формалізмів логіки предикатів першого порядку. Його важливість випливає з того факту, що здебільшого всі формалізми символічного подання знань можуть бути зведені до логіки предикатів першого порядку. Логіка першого порядку відтворює сутність мислення людини введенням поняття логічного підсумку. Вона також дає змогу вводити поняття універсальної правильності в тому сенсі, що логічне твердження може бути правильним незалежно від будь-яких передумов. Логічний

підсумок та універсальна правильність можуть бути описані в термінах моделювання семантики тверджень. Модель для логічної теорії описує умови, за яких теорія правильна. Логічний підсумок є твердженням, правильним для всіх моделей теорії. Далі дедуктивне виведення пояснює, як підсумки виводяться з теорії. Дедукція дає доступ до знань, що неявно подані в теорії. У термінах предикатних рівнянь [8] це означає, що розв'язання рівняння щодо невідомих значень змінних чи відношень між ними дає нам додаткові знання, не подані в явному вигляді в системі рівнянь бази знань.

Додавання рівнянь, отриманих способом дедуктивного виведення з наявної бази знань, є поповненням бази знань та уможливує повторне використання знань. Це має багато практичних переваг, зокрема підвищення ефективності, якості та інноваційної діяльності, зменшення витрат часу й ресурсів, а також сприяння навчанню та розвитку. Проблеми повторного використання знань мають велике значення для різних галузей науки та практики.

Система логічних предикатних рівнянь здатна застосовувати знання предметної галузі, отримані з меншої проблеми, до більш складних проблем тієї самої або спорідненої галузі, але наразі переважній більшості логічних та еволюційних обчислювальних методів не вистачає цієї здатності. Відсутність здатності застосувати вже набуті знання про предметну галузь призводить до споживання більшої кількості ресурсів і часу для вирішення складніших проблем предметної галузі [8]. Залежно від того, як проблема збільшується в розмірах, її стає важко, а іноді навіть непрактично (якщо не неможливо) розв'язати через відсутність ресурсів і часу. Тому необхідна система, що має здатність повторно використовувати отримані знання про проблемну галузь для масштабування в цій галузі.

Система, основана на знаннях, підтримує базу знань, що зберігає символи обчислювальної моделі у формі тверджень щодо галузі, яка розглядається. У нашій ситуації ці твердження записуються як рівняння алгебри скінченних предикатів. Така система здійснює логічні виведення з наявних рівнянь, результати яких можуть поповнювати наявну базу знань, забезпечуючи цим повторне використання знань. Програмні застосунки можуть приймати рішення на основі запитів до бази знань.

Основна мета наукового напрямку полягає в тому, щоб розробити систему, здатну автономно масштабувати навчання, починаючи від незначних

проблем і завершуючи складнішими проблемами тієї самої чи спорідненої сфери, у поведінці, подібній до людської. Для того, щоб автономно масштабувати проблемну галузь, потрібно вміти формувати багаторазово використовувані логічні блоки знань. Для вилучення та повторного використання інформації з логічних рівнянь у проблемній галузі потрібне розширене кодування. Простір пошуку може розширитися, наприклад, у деяких формах генетичного програмування машинного навчання [9–11].

Як правило, агент містить еволюційні обчислення та машинне навчання для вирішення певного завдання в невідомому середовищі. Правила мають форму "якщо умова, то дія". Зазвичай умова подана бітовим рядком фіксованої довжини, визначеним у потрібному алфавіті $\{0,1,\#\}$, де $\#$ – символ, що означає "невизначено", тобто або 0 або 1, а дія подана числовою константою. Техніка класифікаційної системи навчання може масштабуватися в проблемних сферах, але щоразу її потрібно вивчати із самого початку. Крім того, збільшення розмірності проблеми, що призводить до розширення простору пошуку, вимагає великого простору пам'яті, значно збільшує час навчання та зрештою обмежує систему розміром проблеми [12]. За допомогою явної передачі знань домену в системі класифікації можна досягти масштабованості, але це додає упередженості та обмежує використання в кількох доменах.

У роботі [13] досліджується процес формування правил за допомогою алгебри скінчених предикатів і предикатних операцій. Аналізуються двійкові предикати, що використовуються для структурування наявних фактів. Розроблено метод підвищення ефективності повторного використання знань.

Мета цього дослідження – розробити метод повторного використання знань за допомогою вирішення логічних рівнянь для певної предметної галузі, наприклад, медицини. Об'єктом є процес побудови та поповнення баз знань, а предметом – логічне моделювання знань і логічне виведення. Основні завдання роботи: розробити метод поповнення бази знань за допомогою логічних рівнянь із суттєвими змінними, провести дослідження та оцінити результати.

Вирішення завдання

Теорія інтелекту з огляду на своє визначення має обмежитися вивченням власне машиноподібних функцій людського інтелекту, що характеризуються

детермінованістю, дискретністю та скінченністю. До того ж виникають питання про те, якою мірою введені обмеження звужують клас функцій людського інтелекту, що допускають моделювання в межах теорії інтелекту.

Будь-яка обчислювальна система має такі обмеження: 1) алфавіт літер, з яких будуються слова для будь-якої конкретної обчислювальної системи, завжди скінченний; 2) довжина слів, які здатна сприймати та формувати ця система, обмежена деяким кінцевим, наперед заданим числом букв, що визначається конструкцією системи, її швидкістю та терміном служби; 3) реакції обчислювальної системи чітко детерміновані: повторне подання вхідного слова завжди призводить до формування системою того самого вихідного слова.

Проблема повторного використання має три основні причини. Перша стосується розуміння контексту та застосованого рішення, інша – змісту знань, що документуються, і остання стосується всієї системи бази знань або програмного забезпечення, яке використовується для її підтримки. Коли говорять про обмін і повторне використання знань, існує припущення, що знання є такими, що їх можна відтворювати та переміщувати з місця на місце; сутність, що можна отримати від людини-експерта й перенести з однієї комп'ютерної системи чи програми в іншу. Але як можна запропонувати поділитися або повторно використовувати те, що не має таких властивостей, як локальність і стійкість? Знання розглядається як абстракція, що неможливо записати й ніколи не можна мати в руках. Знання – це те, що спостерігач пояснив би розумному агенту, зі свого боку це дає змогу агенту раціонально моделювати свою поведінку для досягнення певних цілей, що сприймаються, відповідно до того, що він дізнався від спостерігача.

Якщо припустити, що все вищезазначене виконано, то проблема ІТ-підтримки процесу управління знаннями все ще потребує вирішення. Метою поточних баз знань є сприяння обміну знаннями та повторному використанню. Для подання знань необхідно враховувати значну кількість факторів. Термінології, онтології та методи вирішення проблем – це декілька з них [14]. Проблема полягає в тому, щоб мати змогу поділитися знаннями, які містяться в різних базах знань, оскільки всі ці фактори відрізняються від однієї бази до іншої. Несумісність систем і форматів також унеможливило об'єднання двох баз знань.

Можна виокремити чотири основні причини.

1. Неоднорідність подання. Існує кілька підходів до подання знань, але один формалізм подання не може бути безпосередньо доданий до іншого. Не існує універсального формалізму подання знань, що ідеально відповідав би всім вимогам, і тому обмін знаннями передбачає переклад змісту однієї бази в іншу.

2. Мовні діалекти. Обмін знаннями між системами може бути дуже складним, якщо знання закодовані різними діалектами. Це може повністю змінити зміст повідомлення або його інтерпретацію.

3. Відсутність визначених правил спілкування. Теоретично окремі системи можуть спілкуватися одна з одною і таким чином мати користь від обміну знаннями, навіть без спільної бази. Але зазвичай це неможливо, оскільки бракує узгодженого протоколу, що б давав змогу системам взаємодіяти та запитувати одна одну.

4. Невідповідності моделі на рівні знань. Навіть у разі усунення всіх перешкод проблеми з термінологією також стануть перешкодою для ефективного спілкування між різними базами. Відсутність спільного словникового запасу не дає змоги співвідносити знання однієї бази з іншою.

Отже, знання розглядається як здатність реагувати певним чином, а не як матеріальна субстанція. Навіть інформацію, що застосовується для подання знань, не можна вважати такою; правила, символи та фрейми не здатні генерувати розумну поведінку. Для того щоб мати змогу математично описувати функції інтелекту, необхідно створити формальну мову, якою можна було б робити такий опис. Формальна мова має обиратися в такий спосіб, щоб нею можна було в зручній формі записати будь-який скінченний алфавітний оператор. Такою мовою є алгебра скінченних предикатів [8].

Поняття скінченного предиката було введено таким чином. Нехай A – скінченний алфавіт, що складається з k літер a_1, a_2, \dots, a_k , а Σ – множина, яка містить два елементи, що позначаються символами 0, 1 і названі відповідно – похибкою та істиною. Змінну, задану на множині A , називають літерною, а змінну, задану на множині Σ , – логічною.

Скінченним n -місцевим предикатом над алфавітом A називається будь-яка функція $f(x_1, x_2, \dots, x_n) = t$ від n буквених аргументів x_1, x_2, \dots, x_n , заданих на множині A , і яка приймає логічні значення t . Іноді скінченний предикат f

називається k -ічним. Цим наголошується, що його алфавіт A складається з k літер.

Кожному скінченному алфавітному оператору може відповідати певний власний скінченний предикат. Зробити це можна таким чином. Нехай $F(x_1, x_2, \dots, x_m) = y_1 y_2 \dots y_m$ – довільно обраний скінченний алфавітний оператор, що перетворює вхідні слова x_1, x_2, \dots, x_m довжини m у вихідні слова y_1, y_2, \dots, y_m тієї самої довжини, які складаються з літер алфавіту A .

Пропонується такий метод поповнення бази знань новими логічними рівняннями, що спрощують повторне використання логічних зв'язків між дискретними ознаками. У загальному вигляді розглядається система рівнянь алгебри скінченних предикатів, що інтерпретує логічну структуру наявних знань щодо предметної галузі. Така система завжди може бути подана у вигляді одного рівняння:

$$f(x_1, x_2, \dots, x_n) = 1,$$

де x_1, x_2, \dots, x_n – всі наявні дискретні ознаки, а формула f відображає логічну структуру бази знань.

Обираються набори ознак, для яких передбачається повторне використання в логічних висновках: $x_{i1}, x_{i2}, \dots, x_{im}$. Для кожного такого набору всі змінні, що не входять до нього, вилучаються квантором існування. Якщо можливо, застосовуються властивості квантора, що допомагають уникнути складних обчислень. Кожне отримане рівняння додається до бази знань.

Крім того, у процесі вирішення практичних задач виникають запитання:

- як конкретні значення цієї ознаки, підставлені в логічне рівняння, вплинуть на зв'язки між іншими ознаками?
- наскільки сильним є логічний зв'язок між двома (або більше) заданими ознаками?

Для відповіді на перше запитання природно виокремити ті предикати (і, відповідно, рівняння), які під час підставлення певного значення ознаки перетворюються на предикати, що приводять до сильнішого зв'язку між змінними, а також такі предикати, підставлення в які цього значення спричиняє послаблення логічного зв'язку між ознаками [13].

Щоб отримати відповідь на друге запитання, необхідно вилучити з вихідного рівняння за допомогою квантора існування всі змінні, крім тих, що розглядаються, і дослідити отримане рівняння

з меншою кількістю змінних, яке описує всі допустимі множини значень досліджуваних ознак.

Коли виникає необхідність повторно використати логічні зв'язки між ознаками конкретного набору, вирішується не початкове складне рівняння, а те рівняння, що пов'язує обрані дискретні змінні.

Для спрощення обчислювальних процедур важливо вилучити з рівняння всі змінні, крім розглянутих, і дослідити отримане рівняння з меншою кількістю змінних. Ці додаткові рівняння поповнюють базу знань і дають змогу уникнути використання значної кількості обчислювальних ресурсів для логічних висновків у базі знань. У роботі [14] розглядається досить широкий клас предикатів, для яких можна задати ефективний алгоритм елімінації змінних без збільшення розміру вихідної формули. Розглянемо такі властивості квантора існування:

- $\exists x x^a = 1$.
- $\exists x \neg x^a = 1$.
- $\exists x (\neg (P(x)Q(x))) = \exists x \neg P(x) \vee \exists x \neg Q(x)$.
- $\exists x ((P(x) \vee Q(x))) = \exists x P(x) \vee \exists x Q(x)$.
- $\exists x (P(x) \& Q(y)) = \exists x P(x) \& Q(y)$.
- $\exists y (P(x) \rightarrow Q(y)) = P(x) \rightarrow \exists y Q(y)$.
- $\exists y (P(x) \rightarrow Q(y)) = P(x) \rightarrow \exists y Q(y)$.
- Припустимо, $P_i(x) \& P_j(x) = 0, i \neq j, j = 1, 2, \dots, k$, тоді:

$$\exists y ((P_1(x) \rightarrow Q_1(y)) \& (P_2(x) \rightarrow Q_2(y)) \& \dots$$

$$\& (P_k(x) \rightarrow Q_k(y))) = (P_1(x) \rightarrow \exists y Q_1(y)) \&$$

$$\& (P_2(x) \rightarrow \exists y Q_2(y)) \& \dots \& (P_k(x) \rightarrow \exists y Q_k(y)).$$
- Якщо тотожність $P_i(x) \equiv 0$ не є істинною для будь-якої $i = 1, 2, \dots, k$ і $P_i(x) \& P_j(x) = 0$ for $i \neq j, i, j = 1, 2, \dots, k$, тоді

$$\exists x ((P_1(x) \rightarrow Q_1(y)) \& (P_2(x) \rightarrow Q_2(y)) \& \dots$$

$$\& (P_k(x) \rightarrow Q_k(y))) = Q_1(y) \vee Q_2(y) \vee \dots \vee Q_k(y).$$

Перелічені вище властивості дають змогу описати широкий клас скінченних предикатів (відповідно до рівнянь), визначених на множині змінних $\{x, y, \dots, z\}$, для яких легко знайти зв'язки між вибраними змінними без збільшення розміру вихідних формул. Визначимо такий клас рекурсивно.

1. Усі "розпізнавання" x^a, x^b, \dots, x^c (a, b, \dots, c – символи, що належать домену для змінної x) належать до Δx .

2. Усі заперечення $\neg x^a, \neg x^b, \dots, \neg x^c$ належать до Δx .

3. Якщо предикати $\neg P(x), \neg Q(x)$ належать до Δx , то предикат $\neg (P(x)Q(x))$ належить до Δx .

4. Будь-який предикат, що не залежить від змінної x , належить до множини Δx .

5. Якщо предикати P_1 та P_2 належать до Δx , тоді предикат $P = P_1 \vee P_2$ належить до Δx .

6. Якщо предикат P_1 належить до Δx і предикат P_2 не залежить від x , тоді предикат $P = P_1 \& P_2$ належить до Δx .

7. Якщо предикат P_1 не залежить від x і предикат P_2 належить до Δx тоді предикат $P = P_1 \rightarrow P_2$ належить до Δx .

8. Нехай предикати P_1, P_2, \dots, P_k не залежать від x ; $P_i \& P_j = 0$ для $i \neq j, i, j = 1, 2, \dots, k$, предикати Q_1, Q_2, \dots, Q_k належать до Δx , тоді

$$P = (P_1 \rightarrow Q_1) \& (P_2 \rightarrow Q_2) \& \dots \& (P_k \rightarrow Q_k)$$

належить до Δx .

9. Якщо предикати P_1, P_2, \dots, P_k залежать тільки від x , $P_i \& P_j = 0$ для $i \neq j, i, j = 1, 2, \dots, k$; для будь-якого $i = 1, 2, \dots, k$ тоді тотожність $P_i \equiv 0$ не є істинною; предикати Q_1, Q_2, \dots, Q_k не залежать від x ; тоді предикат

$$P = (P_1 \rightarrow Q_1) \& (P_2 \rightarrow Q_2) \& \dots \& (P_k \rightarrow Q_k)$$

належить до Δx .

Також може виникнути потреба вилучити зайві змінні з допомогою універсального квантора. У цьому разі можемо скористатися такими властивостями цього квантора:

- $\forall x x^a = 0$
- $\forall x \neg x^a = 0$
- $\forall x \neg (P(x) \vee Q(x)) = \forall x \neg P(x) \& \forall x \neg Q(x)$
- $\forall x (P(x) \& Q(x)) = \forall x P(x) \& \forall x Q(x)$
- $\forall x (P(x) \vee Q(x)) = \forall x P(x) \vee \forall x Q(x)$
- $\forall y (P(x) \& Q(y)) = P(x) \& \forall y Q(y)$

7. Припустимо,

$$P_i(x) \& P_j(x) = 0, i \neq j, i, j = 1, 2, \dots, k,$$

тоді:

$$\begin{aligned} & \forall y ((P_1(x) \& Q_1(y)) \vee (P_2(x) \& Q_2(y)) \vee \dots \\ & \vee (P_k(x) \& Q_k(y))) = (P_1(x) \& \forall y Q_1(y)) \vee \\ & \vee (P_2(x) \& \forall y Q_2(y)) \vee \dots \vee (P_k(x) \& \forall y Q_k(y)). \end{aligned}$$

8. Якщо ідентичність $P_i(x) \equiv 0$ не є істинною для будь-якої $i = 1, 2, \dots, k$ і $P_i(x) \& P_j(x) = 0$ для $i \neq j, i, j = 1, 2, \dots, k$, тоді

$$\begin{aligned} & \forall x ((P_1(x) \& Q_1(y)) \vee (P_2(x) \& Q_2(y)) \vee \dots \\ & \vee (P_k(x) \& Q_k(y))) = Q_1(y) \& Q_2(y) \& \dots \& Q_k(y). \end{aligned}$$

Визначимо рекурсивно клас предикатів Σ_x , з якого можна вилучити змінну x без збільшення розміру формули.

1. Усі "розпізнавання" x^a, x^b, \dots, x^c належать Σ_x .

2. Усі заперечення $\neg x^a, \neg x^b, \dots, \neg x^c$, які не залежать від x , належать Σ_x .

3. Якщо $\neg P_1$ і $\neg P_2$ належать до Σ_x , тоді $\neg(P_1 \vee P_2)$ належить до Σ_x .

4. Якщо предикати P_1 та P_2 належать до Σ_x , тоді предикат $P = (P_1 \& P_2)$ належить до Σ_x .

5. Якщо предикат P_1 належить до Σ_x і предикат P_2 не залежить від x , тоді предикат $P = P_1 \vee P_2$ належить до Σ_x .

6. Якщо предикат P_1 не залежить від x і предикат P_2 належить до Σ_x , тоді предикат $P = P_1 \& P_2$ належить до Σ_x .

7. Припустимо, що предикати P_1, P_2, \dots, P_k не залежать від x , $P_i \& P_j = 0$ для $i \neq j, i, j = 1, 2, \dots, k$ предикати Q_1, Q_2, \dots, Q_k належать до Σ_x , тоді

$$P = (P_1 \& Q_1) \vee (P_2 \& Q_2) \vee \dots \vee (P_k \& Q_k)$$

належить до Σ_x .

8. Якщо предикати P_1, P_2, \dots, P_k залежать тільки від x , $P_i \& P_j = 0$ для $i \neq j, i, j = 1, 2, \dots, k$ для будь-якого $i = 1, 2, \dots, k$ тотожність $P_i \equiv 0$ не є істинною, предикати Q_1, Q_2, \dots, Q_k не залежать від x , тоді предикат $P = (P_1 \& Q_1) \vee (P_2 \& Q_2) \vee \dots \vee (P_k \& Q_k)$ належить до множини Σ_x .

Універсальним способом розв'язання систем рівнянь алгебри скінченних предикатів є приведення предиката, заданого системою рівнянь і початковими умовами, до досконалої диз'юнктивної нормальної форми. Однак така процедура передбачає перебір багатьох проміжних розв'язків, а її практична реалізація потребує значних затрат комп'ютерного часу. Для деяких типів предикатних рівнянь, зважаючи на особливості їх структури, можна розробити простіші алгоритми розв'язання.

Отже, показано, що для репрезентації та інтерпретації знань за умови подання у формальному вигляді інформації про об'єкти й процеси в базах знань використовуються різні методи алгебри скінченних предикатів і предикатних операцій. Так, логічні методи розпізнавання образів передбачають складання та розв'язання логічних рівнянь зі змінними, що набувають значень 1 та 0 залежно від того, чи має об'єкт певну властивість, чи ні. Розв'язання таких рівнянь дає змогу або ідентифікувати об'єкт за наявними наборами значень атрибутивних змінних, або встановити невідомі властивості цього об'єкта. У багатьох практичних задачах, пов'язаних із семантичним обробленням медичних даних, природно-мовної інформації, відомостей про клієнтів, немає необхідності отримувати всі набори значень семантичних ознак, але потрібно отримати один або кілька наборів значень ознак (цільових змінних), що цікавлять користувача. Часто необхідно знайти значення цільових змінних за заданих початкових умов, які є фіксованим набором значень інших ознак. У процесі розв'язання таких задач інші змінні, які відсутні в початкових умовах і не є цільовими, вилучаються з рівняння способом зв'язування їх з екзистенціальними кванторами.

Приклад

Розглянемо медичний приклад поповнення бази знань новими рівняннями. План експерименту такий. Ми використовуємо реальні медичні дані та кодуємо їх за допомогою предикатних рівнянь. Зауважимо, що хоча деякі змінні можуть набувати значення "невідомо", а втім це випадок закритого світу, оскільки "невідомо" означає лише значення з алфавіту, на якому визначена змінна. Отже, кожна галузь для будь-якої змінної є закритою. Після того, як ми написали систему рівнянь за допомогою

експертів, починаємо вилучати змінні, які вважаємо несуттєвими на цей момент. Це не означає, що в іншому разі інші змінні вважатимуться несуттєвими. Істотні змінні – це ті, для яких ми хочемо визначити логічні зв'язки. На виході отримуємо рівняння, де несуттєві змінні вилучено. Отримане рівняння простіше, ніж початкова система, і можна проаналізувати зв'язки між основними змінними більш простим способом. Так, ми поповнюємо базу знань додатковими рівняннями, отриманими як результати дедуктивного виведення з основної бази знань.

Якщо розглядати інформаційний скринінг медичних показників для оцінювання розвитку та профілактики захворювань серця й судин [15–17], то можна виокремити набір ознак для формалізації скринінгових процедур. Розглянемо такі ознаки та їх значення.

$$\text{Стать: } X_1 = \{x_1^1, x_1^2\},$$

де x_1^1 означає жінка; x_1^2 – чоловік.

$$\text{Вік: } X_2 = \{x_2^1, x_2^2, x_2^3\},$$

де x_2^1 – менше ніж 40 років; x_2^2 – від 40 до 50 років; x_2^3 – понад 50 років.

$$\text{Цукровий діабет: } X_3 = \{x_3^1, x_3^2, x_3^3, x_3^4\},$$

де x_3^1 – так; x_3^2 – ні (фактичний діагноз); x_3^3 – ні (діагноз не встановлений); x_3^4 – невідомо.

$$\text{Артеріальна гіпертензія: } X_4 = \{x_4^1, x_4^2, x_4^3, x_4^4\},$$

де x_4^1 – так; x_4^2 – ні (фактичний діагноз); x_4^3 – ні (діагноз не встановлений); x_4^4 – невідомо.

$$\text{Проблеми з нирками: } X_5 = \{x_5^1, x_5^2, x_5^3\},$$

де x_5^1 – так; x_5^2 – ні; x_5^3 – невідомо.

$$\text{Тахікардія: } X_6 = \{x_6^1, x_6^2, x_6^3, x_6^4, x_6^5\},$$

де x_6^1 – так (фактичний діагноз); x_6^2 – так (діагноз не встановлений); x_6^3 – ні (справжній діагноз); x_6^4 – ні (діагноз не встановлений); x_6^5 – невідомо.

Спадковість щодо захворювань серця та судин:

$$X_7 = \{x_7^1, x_7^2, x_7^3\},$$

де x_7^1 – так; x_7^2 – ні; x_7^3 – невідомо.

$$\text{Паління: } X_8 = \{x_8^1, x_8^2, x_8^3\},$$

де x_8^1 – так; x_8^2 – ні; x_8^3 – невідомо.

$$\text{Проблеми з алкоголем: } X_9 = \{x_9^1, x_9^2, x_9^3\},$$

де x_9^1 – так; x_9^2 – ні; x_9^3 – невідомо.

$$\text{Гіподинамія } X_{10} = \{x_{10}^1, x_{10}^2, x_{10}^3\},$$

де x_{10}^1 – так; x_{10}^2 – ні; x_{10}^3 – невідомо.

Ці особливості дають змогу розробити модель ідентифікації діагностичних параметрів, за допомогою якої можна визначити групу здоров'я пацієнта $R = \{r_1, r_2, r_3, r_4\}$, де r_1 – низький ризик захворювань серця та судин; r_2 – помірний ризик; r_3 – високий ризик; r_4 – дуже високий ризик.

Для визначення групи здоров'я використовується набір агрегованих ознак $Q_1 - Q_3$, де Q_1 виражається через X_1 та X_2 ; Q_2 виражається через X_7 до X_{10} ; Q_3 виражається через $X_3 - X_6$.

Значення кожної групи здоров'я та кожної агрегованої ознаки розподілено на чотири класи згідно з відповідною медико-технологічною документацією (уніфікованим клінічним протоколом і локальними протоколами щодо профілактики хвороб серця й судин).

Наприклад, для формування ознаки Q_2 можна скласти таку систему предикатних рівнянь:

$$\left\{ \begin{array}{l} q_2^1 = x_7^2 x_8^2 (x_9^2 \vee x_9^3 (x_{10}^2 \vee x_{10}^3)) \vee x_7^2 x_8^3 x_9^2 x_{10}^2 \vee x_7^3 x_8^2 x_{10}^2 (x_9^2 \vee x_9^3) \\ q_2^2 = x_7^2 ((x_8^1 (x_9^1 x_{10}^2 \vee x_9^2) \vee x_9^3 (x_8^1 x_{10}^2 \vee x_8^2 x_{10}^1)) \vee (x_7^2 (x_8^2 x_9^1 \vee x_8^3 x_9^2) \vee (x_7^2 x_9^3 \vee x_7^3 x_9^1) x_8^3 x_{10}^2 \vee \\ \vee (x_7^2 x_8^3 \vee x_7^3 x_8^2) x_9^1 (x_{10}^2 \vee x_{10}^3) \vee x_7^3 x_8^2 (x_9^2 \vee x_9^3)) (x_{10}^1 \vee x_{10}^3) \vee x_7^3 x_8^1 (x_9^2 x_{10}^2 \vee x_9^3) \vee x_7^3 x_8^3 x_9^2 (x_{10}^1 \vee x_{10}^2) \\ q_2^3 = x_7^1 x_{10}^2 (x_8^1 x_9^2 \vee x_8^2 (x_9^1 \vee x_9^2)) \vee (x_7^1 x_9^3 (x_8^1 \vee x_8^2) \vee (x_7^1 x_8^3 \vee x_7^3 x_8^1) x_9^1) (x_{10}^2 \vee x_{10}^3) \vee x_7^1 x_8^3 (x_9^2 \vee x_9^3) \vee \\ \vee (x_7^2 (x_8^1 (x_9^1 \vee x_9^3) \vee x_8^3 x_9^3) \vee (x_7^2 x_8^3 \vee x_7^3 x_8^2) x_9^1 x_{10}^1 \vee x_7^3 (x_8^1 x_9^2 \vee x_8^3 x_9^1)) (x_{10}^1 \vee x_{10}^3) \vee x_7^3 x_8^3 (x_9^2 x_{10}^1 \vee x_9^3) \\ q_2^4 = x_7^1 x_9^3 x_{10}^1 (x_8^1 \vee x_8^2) \vee (x_7^1 x_8^3 \vee x_7^3 x_8^1) x_9^1 x_{10}^1 \vee (x_7^1 x_8^2 x_9^1 \vee x_7^1 x_9^2 (x_8^1 \vee x_8^2)) (x_{10}^1 \vee x_{10}^3) \end{array} \right.$$

Остаточна класифікація може бути виражена

такою системою:

$$\begin{cases} r_1 = q_1^1 q_2^1 (q_3^1 \vee q_3^2) \vee (q_1^1 q_2^2 \vee (q_1^2 \vee q_1^3) q_2^1) q_3^1 \\ r_2 = q_1^1 (q_2^1 q_3^3 \vee q_2^2 q_3^2) \vee (q_1^1 (q_2^3 \vee q_2^4) \vee q_1^2 (q_2^2 \vee q_2^3) \vee q_1^3 q_2^2 \vee q_1^4 (q_2^1 \vee q_2^2)) (q_3^1 \vee q_3^2) \vee (q_1^2 \vee q_1^3) q_2^1 (q_3^2 \vee q_3^3) \vee \\ \vee (q_1^2 q_2^4 \vee (q_1^3 \vee q_1^4) q_2^3) q_3^1 \\ r_3 = q_2^1 q_3^4 \vee (q_1^1 \vee q_1^2 \vee q_1^3) (q_2^3 \vee q_2^2) (q_3^3 \vee q_3^4) \vee q_1^3 q_3^2 (q_2^3 \vee q_2^4) \vee (q_1^3 \vee q_1^4) q_2^4 q_3^1 \vee (q_1^1 q_2^4 \vee q_1^4 (q_2^1 \vee q_2^2)) q_3^3 \vee \\ \vee (q_1^2 q_2^4 \vee q_1^4 q_2^3) (q_3^2 \vee q_3^3) \\ r_4 = (q_1^1 \vee q_1^2) q_2^4 q_3^4 \vee q_1^3 q_2^4 (q_3^3 \vee q_3^4) \vee q_1^4 q_3^4 (q_2^2 \vee q_2^3) \vee q_1^4 q_2^4 (q_3^2 \vee q_3^3 \vee q_3^4) \end{cases}$$

Дослідимо логічні зв'язки між дискретними ознаками $x_1 - x_{10}$. Насамперед перепишемо систему

предикатних рівнянь у такому вигляді:

$$\begin{aligned} P(q_2, x_1, \dots, x_{10}) = & \\ = & q_2^1 (x_7^2 x_8^2 (x_9^2 \vee x_9^3 (x_{10}^2 \vee x_{10}^3)) \vee x_7^3 x_8^3 x_9^2 x_{10}^2 \vee x_7^3 x_8^3 x_{10}^2 (x_9^2 \vee x_9^3)) \vee q_2^2 (x_7^2 (x_8^1 (x_9^1 x_{10}^2 \vee x_9^2) \vee x_9^3 (x_8^1 x_{10}^2 \vee x_8^2 x_{10}^1)) \vee \\ & \vee (x_7^2 (x_8^2 x_9^1 \vee x_8^3 x_9^2) \vee (x_7^2 x_9^3 \vee x_7^3 x_9^1) x_8^3 x_{10}^2 \vee (x_7^2 x_8^3 \vee x_7^3 x_8^2) x_9^1 (x_{10}^2 \vee x_{10}^3) \vee x_7^3 x_8^2 (x_9^2 \vee x_9^3)) (x_{10}^1 \vee x_{10}^3) \vee \\ & \vee x_7^3 x_8^1 (x_9^2 x_{10}^2 \vee x_9^3) \vee x_7^3 x_8^3 x_9^2 (x_{10}^1 \vee x_{10}^2)) \vee \\ & \vee q_2^3 (x_7^1 x_{10}^2 (x_8^1 x_9^2 \vee x_8^2 (x_9^1 \vee x_9^2)) \vee (x_7^1 x_9^3 (x_8^1 \vee x_8^2) \vee (x_7^1 x_8^3 \vee x_7^3 x_8^1) x_9^1) (x_{10}^2 \vee x_{10}^3) \vee x_7^1 x_8^3 (x_9^2 \vee x_9^3) \vee \\ & \vee (x_7^2 (x_8^1 (x_9^1 \vee x_9^2) \vee x_8^3 x_9^3) \vee (x_7^2 x_8^3 \vee x_7^3 x_8^2) x_9^1 x_{10}^1 \vee x_7^3 (x_8^1 x_9^2 \vee x_8^3 x_9^1)) (x_{10}^1 \vee x_{10}^3) \vee x_7^3 x_8^3 (x_9^2 x_{10}^3 \vee x_9^3)) \vee \\ & \vee q_2^4 (x_7^1 x_9^3 x_{10}^1 (x_8^1 \vee x_8^2) \vee (x_7^1 x_8^3 \vee x_7^3 x_8^1) x_9^1 x_{10}^1 \vee (x_7^1 x_8^2 x_9^1 \vee x_7^1 x_9^2 (x_8^1 \vee x_8^2)) (x_{10}^1 \vee x_{10}^3) \vee x_7^1 x_8^1 x_9^1) = 1 \end{aligned}$$

Видно, що цей предикат належить до класу Δ_{x_7} .

Це вилучення дасть нам зв'язок між змінними

Дослідимо зв'язок між усіма змінними, крім x_7 .

$q_2, x_1, \dots, x_6, x_8, x_9, x_{10}$:

$$\begin{aligned} F = \exists x_7 P(q_2, x_1, \dots, x_{10}) = & \\ = & q_2^1 (x_8^2 (x_9^2 \vee x_9^3 (x_{10}^2 \vee x_{10}^3)) \vee x_8^3 x_9^2 x_{10}^2 \vee x_8^3 x_{10}^2 (x_9^2 \vee x_9^3)) \vee q_2^2 ((x_8^1 (x_9^1 x_{10}^2 \vee x_9^2) \vee x_9^3 (x_8^1 x_{10}^2 \vee x_8^2 x_{10}^1)) \vee \\ & \vee ((x_8^2 x_9^1 \vee x_8^3 x_9^2) \vee (x_9^3 \vee x_9^1) x_8^3 x_{10}^2 \vee (x_8^3 \vee x_8^2) x_9^1 (x_{10}^2 \vee x_{10}^3) \vee x_8^2 (x_9^2 \vee x_9^3)) (x_{10}^1 \vee x_{10}^3) \vee x_8^1 (x_9^2 x_{10}^2 \vee x_9^3) \vee \\ & \vee x_8^3 x_9^2 (x_{10}^1 \vee x_{10}^2)) \vee q_2^3 (x_{10}^2 (x_8^1 x_9^2 \vee x_8^2 (x_9^1 \vee x_9^2)) \vee (x_9^3 (x_8^1 \vee x_8^2) \vee (x_8^3 \vee x_7^3 x_8^1) x_9^1) (x_{10}^2 \vee x_{10}^3) \vee \\ & \vee x_8^3 (x_9^2 \vee x_9^3) \vee ((x_8^1 (x_9^1 \vee x_9^2) \vee x_8^3 x_9^3) \vee (x_8^3 \vee x_8^2) x_9^1 x_{10}^1 \vee (x_8^1 x_9^2 \vee x_8^3 x_9^1)) (x_{10}^1 \vee x_{10}^3) \vee x_8^3 (x_9^2 x_{10}^3 \vee x_9^3)) \vee \\ & \vee q_2^4 (x_9^3 x_{10}^1 (x_8^1 \vee x_8^2) \vee (x_8^3 \vee x_8^1) x_9^1 x_{10}^1 \vee (x_8^2 x_9^1 \vee x_9^2 (x_8^1 \vee x_8^2)) (x_{10}^1 \vee x_{10}^3) \vee x_8^1 x_9^1) = 1 \end{aligned}$$

Необхідно зазначити, що розмір формули не збільшився, тому що предикат $P(q_2, x_1, \dots, x_{10})$ належить до Δ_{x_7} .

Припустимо, нас цікавить зв'язок між q_2, x_9, x_{10} .

Вилучимо інші ознаки з предиката $F(x_1, \dots, x_{10})$:

$$\begin{aligned}
G(q_2, x_9, x_{10}) &= \exists x_1 \exists x_2 \exists x_3 \exists x_4 \exists x_5 \exists x_6 \exists x_7 \exists x_8 P(q_2, x_1, x_{10}) = \\
&= q_2^1 \left((x_9^2 \vee x_9^3 (x_{10}^2 \vee x_{10}^3)) \vee x_9^2 x_{10}^2 \vee x_{10}^2 (x_9^2 \vee x_9^3) \right) \vee q_2^2 \left(((x_9^1 x_{10}^2 \vee x_9^2) \vee x_9^3 (x_{10}^2 \vee x_{10}^1)) \vee \right. \\
&\vee \left((x_9^1 \vee x_9^2) \vee (x_9^3 \vee x_9^1) x_{10}^2 \vee x_9^1 (x_{10}^2 \vee x_{10}^3) \vee (x_9^2 \vee x_9^3) \right) (x_{10}^1 \vee x_{10}^3) \vee (x_9^2 x_{10}^2 \vee x_9^3) \vee x_9^2 (x_{10}^1 \vee x_{10}^2) \vee \\
&\vee q_2^3 \left(x_{10}^2 (x_9^1 \vee x_9^2) \vee x_9^3 (x_{10}^2 \vee x_{10}^3) \vee (x_9^2 \vee x_9^3) \vee ((x_9^1 \vee x_9^3) \vee x_9^1 x_{10}^1 \vee (x_9^2 \vee x_9^1)) (x_{10}^1 \vee x_{10}^3) \vee (x_9^2 x_{10}^3 \vee x_9^3) \right) \vee \\
&\vee q_2^4 \left(x_9^3 x_{10}^1 \vee x_9^1 x_{10}^1 \vee (x_9^1 \vee x_9^2) (x_{10}^1 \vee x_{10}^3) \vee x_9^1 \right) = 1.
\end{aligned}$$

Отже, скоротили початкову формулу й маємо більш просту залежність між обраними медичними ознаками. Отримавши необхідну залежність, можна розв'язати рівняння з однією або декількома цільовими змінними. Останнє рівняння додаємо до бази знань, щоб нові знання, отримані внаслідок наведених вище обчислень, поповнили наявну базу знань. Розглянемо можливі рішення цього рівняння.

Наприклад, $x_9 = 1$, тобто є проблеми з алкоголем. Тоді отримаємо таку залежність між ознаками:

$$q_2^2 x_{10}^2 \vee q_2^3 x_{10}^2 \vee q_2^4 = 1.$$

З останнього рівняння видно, що за умови $x_{10}^2 = 1$, q_2^4 має дорівнювати 4, тобто якщо є проблеми з алкоголем та невідомо, чи є гіподинамія, у класифікації пацієнтів необхідно використовувати значення агрегованої ознаки, що дорівнює 4.

Отже, вилучаючи різні набори змінних з рівнянь, що задовольняють наведені умови, поповнюємо базу знань і повторно використовуємо знання, які були неявно задані в початковій базі даних. Це спрощує процес логічного виведення та масштабує логічну систему подання знань. У цьому разі неявно задані знання стають явними.

Висновки

Дескриптивні підходи до моделювання процедур повторного використання знань ефективні у ситуаціях, коли розглядаються дискретні ознаки об'єктів, які пов'язані між собою складними логічними умовами. Алгебра скінченних предикатів дає змогу формувати бази знань у формі логічних рівнянь із дискретними змінними. Застосування кванторних операцій допомагає поповнювати наявні бази знань новими рівняннями, що спрощує повторне використання знань, заданих неявно.

Зручність диз'юнктивної алгебри скінченних предикатів у тому, що її мовою стисло й зручно записуються закони істинності та хибності, які відіграють особливу роль, оскільки вони задають

вимоги, необхідні й достатні для коректного введення змінних ознак на скінченних множинах. Закон істинності задає область зміни змінної, а закон хибності забезпечує попарну різницю всіх елементів множини, на якій задана змінна. Алгебра скінченних предикатів дає змогу інтерпретувати знання в чіткій математичній формі, де різні ознаки та їх значення пов'язані між собою за допомогою булевих і предикатних операцій.

Для ефективного розв'язання рівнянь із невідомими скінченними предикатами доцільно спочатку дослідити рівняння, записані за допомогою операцій булевої алгебри, оскільки саме ці операції найчастіше мають місце під час запису довільних предикатних рівнянь.

Показано, що в процесі побудови системи ознак для бази знань доцільно використовувати алгебро-логічну модель предметної галузі. Запропоновано підхід, оснований на алгебрі скінченних предикатів. Рекомендовано виокремити ті предикати (i , відповідно, рівняння), які за умови підставлення певного значення ознаки перетворюються на предикати, що дають сильніший зв'язок між змінними, а також такі предикати, підставлення в які цього значення призводить до послаблення логічного зв'язку між ознаками.

Запропоновано алгоритм вилучення несуттєвих змінних у розв'язанні алгебро-логічних рівнянь, який призводить до зменшення обсягу необхідних обчислень.

Розглянуто скінченні предикатні рівняння різних типів. Наведено задачі класифікації на основі предикатних рівнянь. Задача класифікації об'єктів на основі ознак, що набувають дискретних значень, описана математично як розв'язок предикатних рівнянь.

Запропоновано метод поповнення бази знань у формі логічних рівнянь для спрощення процедури повторного використання знань. Спочатку вилучаються змінні, що вважаються несуттєвими. База знань поповнюється простішими рівняннями, які можуть бути багаторазово використані за потреби. У практичних задачах, пов'язаних з інтерпретацією

знань, що здебільшого подані в текстовій формі, немає необхідності отримувати всі набори значень семантичних ознак, але потрібно отримати один або кілька суттєвих наборів значень ознак (цільових змінних), які є фіксованим набором значень інших ознак. У розв'язанні таких задач інші змінні, що не містяться в початкових умовах і не є цільовими, вилучаються з рівняння способом зв'язування їх з екзистенціальними кванторами.

Використання математичного апарату теорії інтелекту, методу компараторної ідентифікації та інструментарію алгебри скінченних предикатів дає змогу побудувати модель ідентифікації медично-діагностичних параметрів у вигляді системи

предикатних рівнянь, у процесі розв'язання яких маємо інтерпретацію медичних знань у певній галузі.

Описано широкий клас предикатів, з яких можна вилучити зайві змінні й зосередитися на зв'язках між суттєвими змінними. Запропоновано метод вилучення несуттєвих змінних способом застосування екзистенціального квантора й продемонстровано його на реальному медичному прикладі.

Розглянуто практичний приклад рішення цієї задачі. Подальші дослідження планується присвятити аналізу предикатних операцій, що дають змогу якомога швидше отримувати нові предикатні рівняння для автоматичної генерації логічних правил і визначати важливість інформаційних ознак.

Список літератури

1. L. He and P. Jiang Manufacturing Knowledge Graph: A Connectivism to Answer Production Problems Query With Knowledge Reuse, in *IEEE Access*, Vol. 7, 2019. P. 101231–101244. DOI: 10.1109/ACCESS.2019.2931361
2. B. Zhou, J. Bao, Y. Liu and D. Song BA-IKG: BiLSTM Embedded ALBERT for Industrial Knowledge Graph Generation and Reuse, *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*, Warwick, United Kingdom, 2020. P. 63–69. DOI: 10.1109/INDIN45582.2020.9442198
3. L. He and P. Jiang P. SaaS: knowledge service-oriented manufacturing workflow model for knowledge collaboration and reuse, *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, Hong Kong, China, 2020, P. 570–575. DOI: 10.1109/CASE48305.2020.9216974
4. D. Sapra and A. D. Pimentel Deep Learning Model Reuse and Composition in Knowledge Centric Networking, *29th International Conference on Computer Communications and Networks (ICCCN)*, Honolulu, HI, USA, 2020. P. 1–11. DOI: 10.1109/ICCCN49398.2020.9209668
5. N. Sharonova et al. Issues of Fact-based Information Analysis. *International Conference on Computational Linguistics and Intelligent Systems*, 2018. URL: <https://www.semanticscholar.org/paper/Issues-of-Fact-based-Information-Analysis-Sharonova-Doroshenko/f923b77b8561736202388db853e51df9bb7b9301>
6. W. Adjandra, Y. B. Putrapratama, A. Wiraguna, D. I. Sensuse and N. Safitri Systematic Literature Review Knowledge Reuse in Software Development, *International Conference on Computer Science and Engineering (IC2SE)*, Padang, Indonesia, 2021, P. 1–7. DOI: 10.1109/IC2SE52832.2021.9792093
7. Y. B. Putrapratama, W. Adjandra, A. Wiraguna, D. I. Sensuse and N. Safitri Knowledge Reuse Evaluation in Software Development: A Case Study on a Startup Company, *Sixth International Conference on Informatics and Computing (ICIC)*, Jakarta, Indonesia, 2021, P. 1–7. DOI: 10.1109/ICIC54025.2021.9632904
8. I. Shubin Development of conjunctive decomposition tools. *CEUR Workshop Proceedings*, 2021, Vol. 2870, P. 890–900 URL: <https://ceur-ws.org/Vol-2870/>
9. K. U. Sarker, A. B. Deraman and R. Hasan, Descriptive Logic for Software Engineering Ontology: Aspect Software Quality Control, *4th International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, Malaysia, 2018. P. 1–5. DOI: 10.1109/ICCOINS.2018.8510585
10. N. Kamide Sequential Fuzzy Description Logic: Reasoning for Fuzzy Knowledge Bases with Sequential Information, *IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*, Miyazaki, Japan, 2020. P. 218–223. DOI: 10.1109/ISMVL49045.2020.000-2
11. R. Rawal, K. Goel and C. Gupta COVID-19: Disease Pattern Study based on Semantic-Web Approach using Description Logic, *IEEE International Conference for Innovation in Technology (INOCON)*, Bangluru, India, 2020. P. 1–5. DOI: 10.1109/INOCON50539.2020.9298278
12. I. Shubin, S. Snisar and S. Litvin Formalization and Application of Algebraic Methods in Automated Intelligent Systems, *IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*, Kharkiv, Ukraine. 2021. P. 67–70. DOI: 10.1109/PICST54195.2021.9772174
13. Каратаєв О.А., Шубін І.Ю. Проблеми повторного використання знань при проєктуванні програмних систем. *Сучасний стан наукових досліджень та технологій в промисловості*, 2023. No. 2 (24). С. 25–34. DOI: 10.30837/ITSSI.2023.24.025

14. Karataiev O., Sitnikov D., Sharonova N. A Method for Investigating Links between Discrete Data Features in Knowledge Bases in the Form of Predicate Equations, *CEUR Workshop Proceedings*, 2023. P. 224–235. URL: <https://ceur-ws.org/Vol-3387/paper17.pdf>
15. Melnik K., Cherednichenko O., Glushko V. Towards medical screening information technology: the healthgrid-based approach *Information Systems: Methods, Models, and Applications*. Heidelberg: Springer, 2013. P. 202–204. DOI: 10.1007/978-3-642-38370-0_21
16. H. Kandil et al. Analysis of The Importance of Systolic Blood Pressure Versus Diastolic Blood Pressure in Diagnosing Hypertension: MRA Study. *IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, United Arab Emirates, 2020. P. 443–447. DOI: 10.1109/ICIP40778.2020.9190990
17. M. Zhang, T. Zhang and Y. Cheng Intelligent Method of Non-Invasive Detection and Grading of Hypertension Using Fingertip Photoplethysmography. *3rd International Conference on Applied Machine Learning (ICAML)*, Changsha, China, 2021. P. 418–422. DOI: 10.1109/ICAML54311.2021.00094

References

1. He, L., Jiang, P. (2019), "Manufacturing Knowledge Graph: A Connectivism to Answer Production Problems Query With Knowledge Reuse", in *IEEE Access*, Vol. 7. P. 101231–101244. DOI: 10.1109/ACCESS.2019.2931361
2. Zhou, B., Bao, J., Liu, Y., Song, D. (2020), "BA-IKG: BiLSTM Embedded ALBERT for Industrial Knowledge Graph Generation and Reuse," *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*, Warwick, United Kingdom. P. 63–69, DOI: 10.1109/INDIN45582.2020.9442198
3. He, L., Jiang, P. (2020), "P-SaaS: knowledge service-oriented manufacturing workflow model for knowledge collaboration and reuse," *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, Hong Kong, China. P. 570–575. DOI: 10.1109/CASE48305.2020.9216974
4. Sapra, D., Pimentel, A. (2020), "Deep Learning Model Reuse and Composition in Knowledge Centric Networking," *29th International Conference on Computer Communications and Networks (ICCCN)*, Honolulu, HI, USA. P. 1–11. DOI: 10.1109/ICCCN49398.2020.9209668
5. Sharonova, N. et al. "Issues of Fact-based Information Analysis". *International Conference on Computational Linguistics and Intelligent Systems*, 2018. available at: <https://www.semanticscholar.org/paper/Issues-of-Fact-based-Information-Analysis-Sharonova-Doroshenko/f923b77b8561736202388db853e51df9bb7b9301>
6. Adjandra, W., Putrapratama, Y., Wiraguna, A., Sensuse, D. and Safitri, N. (2021), "Systematic Literature Review Knowledge Reuse in Software Development," *International Conference on Computer Science and Engineering (IC2SE)*, Padang, Indonesia. P. 1–7. DOI: 10.1109/IC2SE52832.2021.9792093
7. Putrapratama, Y., Adjandra, W., Wiraguna, A., Sensuse, D. and Safitri, N. (2021), "Knowledge Reuse Evaluation in Software Development: A Case Study on a Startup Company", *Sixth International Conference on Informatics and Computing (ICIC)*, Jakarta, Indonesia. P. 1–7. DOI: 10.1109/ICIC54025.2021.9632904
8. Shubin, I. "Development of conjunctive decomposition tools". *CEUR Workshop Proceedings*, 2021, Vol. 2870, P. 890–900 available at: <https://ceur-ws.org/Vol-2870/>
9. Sarker, K., Deraman, A. and Hasan, R. (2018), "Descriptive Logic for Software Engineering Ontology: Aspect Software Quality Control", *4th International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, Malaysia. P. 1–5. DOI: 10.1109/ICCOINS.2018.8510585
10. Kamide, N. (2020) "Sequential Fuzzy Description Logic: Reasoning for Fuzzy Knowledge Bases with Sequential Information", *IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*, Miyazaki, Japan. P. 218–223. DOI: 10.1109/ISMVL49045.2020.000-2
11. Rawal, R., Goel, K., Gupta, C. (2020), "COVID-19: Disease Pattern Study based on Semantic-Web Approach using Description Logic", *IEEE International Conference for Innovation in Technology (INOCON)*, Bangluru, India. P. 1–5. DOI: 10.1109/INOCON50539.2020.9298278
12. Shubin, I., Snisar S., Litvin, S. (2021), "Formalization and Application of Algebraic Methods in Automated Intelligent Systems", *IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*, Kharkiv, Ukraine. P. 67–70. DOI: 10.1109/PICST54195.2021.9772174
13. Karataiev, O., Shubin, I. (2023), "Reuse of Information Based on the Interpretation of Knowledge". ["Problemy povtornoho vykorystannia znan pry proiektuvanni prohramnykh system"]. *Technologies and Scientific Solutions for Industries*. No. 2(24). P. 25–34. DOI: 10.30837/ITSSI.2023.24.025
14. Karataiev, O., Sitnikov, D., Sharonova, N. "A Method for Investigating Links between Discrete Data Features in Knowledge Bases in the Form of Predicate Equations", *CEUR Workshop Proceedings*, 2023. P. 224–235. available at: <https://ceur-ws.org/Vol-3387/paper17.pdf>

15. Melnik, K., Cherednichenko, O., Glushko, V. (2013), "Towards medical screening information technology: the healthgrid-based approach". *Information Systems: Methods, Models, and Applications*. Heidelberg: Springer. P. 202-204. DOI: 10.1007/978-3-642-38370-0_21
16. Kandil, H. et al. (2020), "Analysis of The Importance of Systolic Blood Pressure Versus Diastolic Blood Pressure in Diagnosing Hypertension: MRA Study", *IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, United Arab Emirates. P. 443–447. DOI: 10.1109/ICIP40778.2020.9190990
17. Zhang, M., Zhang, T. and Cheng, Y. (2021), "Intelligent Method of Non-Invasive Detection and Grading of Hypertension Using Fingertip Photoplethysmography". *3rd International Conference on Applied Machine Learning (ICAML)*, Changsha, China. P. 418–422. DOI: 10.1109/ICAML54311.2021.00094

Надійшла 15.09.2023

Відомості про авторів / About the Authors

Каратаєв Олександр Анатолійович – Харківський національний університет радіоелектроніки, аспірант кафедри програмної інженерії, Харків, Україна; e-mail: tosanik@gmail.com; ORCID ID: <https://orcid.org/0009-0007-6654-1327>

Ситніков Дмитро Едуардович – кандидат технічних наук, доцент, Харківський національний університет радіоелектроніки, професор кафедри системотехніки, Харків, Україна; e-mail: dmytro.sytnikov@nure.ua; ORCID ID: <https://orcid.org/0000-0003-1240-7900>

Karataiev Oleksandr – Kharkiv National University of Radio Electronics, Postgraduate, Kharkiv, Ukraine.

Sytnikov Dmytro – PhD (Engineering Sciences), Associate Professor, Kharkiv National University of Radio Electronics, Professor at the Department of System Engineering, Kharkiv, Ukraine.

THE METHOD OF REUSE OF KNOWLEDGE IN THE FORM OF LOGICAL EQUATIONS

The subject of research is the processes of building and replenishing knowledge bases, the processes of knowledge reuse and the creation of software systems based on knowledge bases, the interpretation of knowledge is one of the approaches to their repeated application, which consists in deriving new knowledge based on the available facts in the knowledge base. The **purpose** of the research is to develop a method of knowledge reuse by solving logical equations of finite predicates for a certain subject area. The **tasks** were performed: to investigate descriptive approaches to logical modeling of the subject area, which make it possible to reuse the knowledge given by the system of logical equations within the algebra of finite predicates, to develop a method of replenishing the knowledge base in the form of predicate equations by extracting variables from logical systems, where equations represent complex logical connections between discrete features of objects or processes. The **methods** were used to solve the listed problems: the algebra of finite predicates, quantifier operations with predicates for the interpretation of knowledge. The **results** were obtained: descriptive methods of knowledge reuse were considered; a method of replenishing the knowledge base in the form of logical equations is proposed simplify the further use of implicit connections between discrete features; an example of replenishment of the knowledge base of the medical field is considered, which makes it possible to reuse knowledge given implicitly. **Conclusions:** the proposed method makes it possible to replenish the knowledge base in the form of logic equations by adding predicate equations connecting separate sets of discrete variables that are of interest to a scientist or a practitioner. The method of replenishing the knowledge base is based on adding new simplified equations; simplified predicate equations model logical patterns implicitly contained in the main knowledge base; an experimental study was conducted.

Keywords: software engineering; knowledge bases; reuse of knowledge; algebra of finite predicates; logical equations.

Бібліографічні описи / Bibliographic descriptions

Каратаєв О. А., Ситніков Д. Е. Метод повторного використання знань у формі логічних рівнянь. *Сучасний стан наукових досліджень та технологій в промисловості*. 2023. № 3 (25). С. 15–26. DOI: <https://doi.org/10.30837/ITSSI.2023.25.015>

Karataiev, O., Sytnikov, D. (2023), "The method of reuse of knowledge in the form of logical equations", *Innovative Technologies and Scientific Solutions for Industries*, No. 3 (25), P. 15–26. DOI: <https://doi.org/10.30837/ITSSI.2023.25.015>