

## РОЗВИТОК ТА ПОРІВНЯЛЬНІ ХАРАКТЕРИСТИКИ ЛОГІКО-ОНТОЛОГІЧНИХ ФОРМАЛЬНИХ ТЕОРІЙ

**Abstract:** In the paper are aspects of evolution of the logic-ontological theories of representation of knowledge contained in the natural language are shown. In particular, there are presented three fundamental methods of ontological categories generating. The milestones in the evolution of the classical logics' main subsets and their practical application versions are described; the systems' abilities and shortcomings are pointed out. Basing on described logical systems' abilities, the comparative analysis was carried out and comparative estimations were brought together in a table. Using the table's data, an advisability of using the logical systems is proved and a probable scheme of their applying is suggested.

**Key words:** knowledge-based information system, ontology, first order logics, syllogistics, frame systems, conceptual graphs.

**Анотація:** У роботі описані аспекти розвитку формальних теорій логіко-онтологічного представлення знань, що містяться у природномовних текстах. Зокрема, представлено три фундаментальні методи генерування онтологічних категорій; описано найвагоміші віхи розвитку основних підмножин класичної логіки та варіантів їх реалізації у прикладних системах; вказано на можливості й обмеження даних підмножин і систем. На основі описаних можливостей логічних систем проведено їх порівняльний аналіз; відносні оцінки аналізу зведено в таблицю. За даними таблиці обґрунтовано доцільність та запропоновано можливу схему використання розглянутих систем логіки.

**Ключові слова:** знанняорієнтована інформаційна система, онтологія, логіка першого порядку, силогістика, фреймові системи, концептуальні графи.

**Аннотация:** В работе описаны аспекты развития формальных теорий логико-онтологического представления знаний, содержащихся в естественноречевых текстах. В частности, представлены три фундаментальных метода генерирования онтологических категорий, описаны важнейшие моменты развития основных подмножеств классической логики и вариантов их реализации в прикладных системах; указано на возможности и ограничения этих подмножеств и систем. На основе описанных возможностей логических систем выполнен их сравнительный анализ, относительные оценки анализа сведены в таблицу. По данным таблицы обоснована целесообразность и предложена возможная схема использования рассмотренных систем логики.

**Ключевые слова:** знаниеориентированная информационная система, онтология, логика первого порядка, силлогистика, фреймовые системы, концептуальные графы.

### 1. Вступ

Особливістю сучасних інтелектуальних інформаційних систем (ІІС) є інтеграційні процеси, які впливають на розвиток їх архітектури і функціональних можливостей. Одним із напрямків є система інтеграція знань як таких, а також технологій їх ефективного формування, представлення, обробки та використання. Не менш важливим фактором є інтеграція двох областей штучного інтелекту (ШІ), які колись розвивались майже паралельно та незалежно: knowledge-engineering і комп'ютерна лінгвістика. В загальному випадку це відображає природну схему взаємодії людини з навколишнім світом. Вказану інтеграцію можна розглядати як закономірний еволюційний розвиток різних наукових теорій, започаткованих ще з давніх часів і відомих як логіка й онтологія. В науковій практиці відомі силогізми Аристотеля і онтологія Аристотеля; логіка Пірса і фундаментальний принцип Пірса розбудови онтології; математична теорія Лейбніца і його принцип генерування решіток онтологічних категорій довільного рівня тощо.

### 2. Постановка задачі

Розробка знанняорієнтованої інформаційної системи (ЗОІС) у будь-якій прикладній галузі включає розробку відповідної онтології, в тому числі і структуру доменно-незалежних категорій верхнього рівня, які повинні бути представлені експліцитно. Отже, має бути розроблена онтологічна

надбудова (або метаонтологія) для онтології прикладної галузі, що представляє об'єкт проектування. Вказана структура визначається множинами концептуальних відношень (або онтологічних зв'язувань концептів) і системою аксіом та фактів (або відмінних ознак). Слід зазначити, що в сучасних системах керування базами даних (СКБД) реляційного типу набори об'єктів бази даних і процедури маніпулювання ними вже задані розробниками системи, тобто задана в певному вигляді метаонтологія прикладного домену.

*Вибір онтологічних категорій* – перший крок у проектуванні бази даних, бази знань чи об'єктно-орієнтованої (ОО) системи [19]. Їх вибір визначає сукупність інформаційних структур, які будуть представлені у прикладній комп'ютерній програмі або сімействі прикладних комп'ютерних програм. Суттєво впливає експліцитна онтологія в комп'ютерних програмах, орієнтованих на обробку природної мови. Таку онтологію називають мовно-онтологічною картиною світу.

Загальновідомо, що не менш важливою ланкою в технологічному ланцюзі комп'ютерної обробки природномовних текстів (ПМТ) є формально-логічне представлення його структури та правил виведення. У своїй основі таке представлення припускає використання або відомої підмножини логіки першого порядку (First Order Logic або FOL), або суттєво зміненої. На сьогодні [19] представляються нові види FOL, з відмінним від класичного набором логічних операторів і кванторів.

Якщо логіка нам говорить, що дещо існує і надає логічні оператори маніпулювання сутностями, то онтологія, по-перше, надає сам словник цих сутностей, а, по-друге, формалізоване представлення усіх видів абстрактних і матеріальних сутностей, що становлять світ в їх взаємовідношеннях. Висловлювання на природній мові, будучи переведеним у формально-логічне представлення, вже може бути "зрозумілим" комп'ютеру та опрацьовано відповідно до конкретних потреб користувача. Тому авторам здалось доцільним прослідкувати еволюцію розвитку логіко-онтологічних формальних теорій та їх можливостей, зокрема, щодо *обробки природномовних текстів*. Зважаючи на статус класичної двозначної логіки [19], в даній роботі розглядаються підмножини лише класичної логіки.

### **3. Онтологія**

Предмет дослідження класифікаційно-категоріальної метаонтології походить від давньогрецьких філософів Геракліта, Платона, Аристотеля та їх сучасників в Індії і Китаї [3, 19]. У своєму першому трактаті "Категорії" Аристотель представив десять основних категорій для класифікації чогось, що може бути сказано або стверджено про щось. Метод Аристотеля утворення нових категорій за родом і його відмінними властивостями є фундаментальним і за своєю суттю представляє принцип дихотомії, тобто розділення на дві протилежні частини. Пізніше неоплатонік Порфирій упорядкував категорії Аристотеля у вигляді дерева, а середньовічні логіки розвинули його у більш деталізовану ієрархію, яку вони назвали "дерево Порфирія".

Наступний фундаментальний принцип формування категорій почав розробляти німецький філософ Кант, побудувавши свою таблицю, що складається із чотирьох груп, по три категорії в кожній. Він розглядав цю таблицю як принципову каркасну структуру, відмінну від структури Аристотеля. Аналогічну тріадичну структуру підтримував і німецький філософ Гегель.

Найбільш повний розвиток тріадичний принцип формування категорій одержав у роботах американського філософа-прагматика Пірса. Він взяв за основу структури тріад Канта й Гегеля і після розширеного аналізу усвідомив, що не всі тріади Канта відбивали три більш базові категорії. Останні він назвав *Первинність*, *Вторинність* і *Третинність*.

"Перше є концепція буття або існування *незалежно від чого-небудь* іще. Друге є концепція буття, *пов'язаного з чимось*, концепція реакції із чим-небудь іще. Третє є концепція *посередництва*, за допомогою чого перше й друге об'єднані у відношення" [19].

У розрізненні тріад Пірс підкреслював рівний статус усіх трьох категорій. Формально ці три види категорій характеризуються мінімальним числом сутностей, які повинні бути включені в їхнє визначення. *Первинність* може бути визначена одномісним предикатом  $P(x)$ , що описує сутність  $x$  її вбудованими якостями, незалежними від чого-небудь зовнішнього до  $x$ . *Вторинність* вимагає двомісного предиката  $Q(x, y)$ , що описує деяку реакцію між сутністю  $x$  і незалежною від  $x$  сутністю  $y$ . *Третинність* вимагає як мінімум тримісного предиката  $R(x, y, z)$ , що описує, як сутність  $x$  слугує сполучною ланкою між двома іншими сутностями:  $y$  і  $z$ . Пірс стверджував, що немає необхідності виходити далі за межі *Третинності*, тому що всі наступні "инності" можуть бути визначені в термінах тріад.

Трохи пізніше, незалежно від Пірса, англійський математик і логік Уайтхед розробив свою метаонтологію, що включає вісім "категорій існування", які становлять дві тріади Пірса, доповнені двома категоріями для генерування комбінацій. Уайтхед визначив категорії для *дійсних сутностей*, *приєднань* (prehension) і *об'єднань* (nexus), які становлять тріаду матеріальних *Первинності*, *Вторинності* і *Третинності*. Дійсна сутність існує сама по собі. *Приєднання* є спрямоване відношення або реакція між двома сутностями. *Об'єднання* – зв'язування двох або більше приєднань.

Описаний тріадичний принцип формування категорій являє собою (у сучасному розумінні) принцип трихотомії або його ще називають принципом Пірса формування онтологічних категорій.

Крім відомих робіт з математики, німецький філософ-математик Лейбніц вніс значний вклад у розробку формальних математичних основ для формування онтологічних категорій і визначення операцій для їх опрацювання. Замість оперування термінами категорій він призначав категоріям певні прості числа і операцією множення генерував нові категорії, а операцією ділення визначав приналежність "тип-супертип". Добутки простих чисел формували свого роду ієрархію, названу решіткою. Згодом таке генерування категорій було названо комбінаційним методом Лейбніца.

Отже, описане вище представляє три фундаментальних методи генерування онтологічних категорій взагалі як для метаонтології верхнього рівня, так й онтологій прикладних галузей. Причому немає ніякого формального способу визначення пріоритетної значимості того або іншого методу. При розробці онтології домену пріоритет установлюється самим дослідником на основі його інтуїції і професійних знань. Прикладом є численні варіанти метаонтології, розроблені нашими великими попередниками й сучасними вченими при створенні сімейств комп'ютерних прикладних програм (системи Сус, Chat-80, Microcosmos, WordNet та ін.).

Загально визнано, що самим верхнім (нульовим) рівнем класифікаційно-категоріальної метаонтології є деякий універсальний супертип ( $U$ ), що формально є типом найвищого рівня категоризації, не має ніякої диференціації й задовольняє таким аксіомам:

- усякий  $x$  є елемент  $U$ ;
- кожна категорія є підкатегорією  $U$ ;
- усі підкатегорії визначаються додатковими розходженнями в  $U$ , щоб показати, як вони відрізняються від  $U$  й одна від іншої.

Відомі три базові групи категорій, що становлять перший рівень категорій метаонтології і відповідають основним законам філософії й принципам дихотомії й трихотомії. Першу групу представляють категорії *Матеріальне* й *Абстрактне*, другу – *Незалежне*, *Зв'язане* та *Проміжне*, і третю – *Тривале* та *Подійне*. Причому, як і у випадку з фундаментальними принципами, не існує теоретичної основи вважати кожну із зазначених груп більш пріоритетною.

Другий рівень категорій можна обчислити за комбінаційним методом Лейбніца, як таким, що генерує найбільш повне і симетричне число категорій. Тоді загальне число категорій другого рівня може бути обчислене як векторний добуток множин  $\{T, П\}$ ,  $\{M, A\}$  та  $\{H, Зв, Пр\}$ , де  $T, П, M, A, H, Зв$  і  $Пр$  – акроніми категорій із зазначених трьох базових груп, відповідно *Тривале*, *Подійне*, *Матеріальне*, *Абстрактне*, *Незалежне*, *Зв'язане* й *Проміжне*.

Метаонтологія, побудована таким способом, містить чотири рівні категорій і описана в [11].

Варто підкреслити, що незалежно від того, як названі згенеровані категорії, істотними є тільки множини (концептуальних) відношень, що зв'язують концепти в онтології, і система аксіом, правил і фактів, обраних для конкретної прикладної галузі.

#### 4. Логіка

Силогістика. Силогізми – "умовивід, в силу якого, признавши істинність посилок силогізму, не можна не погодитись з істинністю висновку, що витікає з посилок" [6]. Наприклад, "*Птахи літають за допомогою крил* (головна посилка); *Геша є птахом* (вторинна посилка), тоді *Геша літає за допомогою крил* (висновок)". Висновок зроблено на основі збіжності середніх термінів – в даному прикладі слів "*птахи*" та "*Геша*". Поняття силогізмів було введено Аристотелем. Також у нього вперше з'являються твердження, що оперують не з конкретними об'єктами, а з їх мнемоніками (літери латинського алфавіту). В такий спосіб він будує універсальні шаблони силогізмів. В період середньовіччя силогізми доповнюються. Позначаючи літерами А, Е, І та О відповідно універсальне твердження, окреме твердження, універсальне заперечення та окреме заперечення, силогізми отримують назви за логічним типом трьох їх частин: *Barbara* (AAA), *Celarent* (EAE), *Darii* (AII), *Ferio* (EIO) та ін. [14, 19]. Середньовічний філософ Адам, аналізуючи протирічні силогізми, ділить твердження на такі, що можуть стверджувати про себе, й такі, що не можуть (як то "*Це висловлювання неправдиве*"). Іспанський філософ Луллій узагальнює силогізми до правил виведення із загального – іншого загального, окремого, невизначеного та індивідуального. Однак, силогізми не придатні для представлення довільних наукових знань, бо символічний зв'язок не може повністю виразити смислового зв'язку, на що звернув увагу німецький філософ-математик Юнг і для чого навів приклад "*А є батьком. В є сином для А. Тоді А є батьком для В*" [14].

Силогізми також не дають змогу фіксувати окремі твердження, як то "*Птахи літають за допомогою крил*", а представляють собою знання, що виводяться. Хоча в силогічних виразах використовуються слова "*і*", "*якщо... то...*", "*кожен*", "*деякий*", "*обов'язково*" і т.п., проте вони не виділяються окремо, тому про логічні оператори чи квантори в силогістиці мови не йдеться. Формально силогізми представляються лінійними *текстовими шаблонами* з невизначеним середнім термом. Автоматичне виведення в системі силогізмів може проводитись шляхом порівняння структури висловлювання з шаблонами, та пошуку середніх термів у головній і вторинній посилках, та у вторинній посилці і висновку за одну ітерацію.

Логіка висловлювань та алгебра логіки. Логіка висловлювань представляє собою повну (визначення [4], доведення [10]) систему ітеративної побудови та розбору логічних висловлювань, де атомарною є конструкція, для складових якої не можна встановити істинність. Розділення мови на окремі висловлювання провів французький філософ Абеляр – він вперше у логіці запропонував "починати з вивчення простих, а потім складних висловлювань" [14], у зв'язку з чим виділив зв'язку (сорула) у твердженнях. *Центральним поняттям* у логіці висловлювань та у всій класичній логіці взагалі є поняття істинності.

Розділення висловлювань на істинні й неправдиві вперше провів Луллій. Сучасне застосування логіки, зокрема, в інженерії знань, неодмінно передбачає автоматичну обробку, що може проводитись математичним обчисленням. Так, ще англійський філософ-матеріаліст Гоббс вперше висловив думку, що "...мислення є ніщо інше, як обчислення: при цьому додавати й віднімати можна і величини, і тіла, рухи, часи, якості..." [14], а пізніше Юнг зробив наголос саме на *математичній формі обчислення висловів*. Лейбніц на основі цих течій у науці про логіку та своїй тези, "що всі поняття можуть бути зведені до елементарних неподільних понять" [14], а складні поняття утворюються за допомогою операції перемноження (логічного) і перетину об'ємів понять та що набір простих понять має бути несуперечливим, розробляє систему математичного представлення тверджень. За прості твердження він узяв прості числа, а за основні операції – арифметичні множення й ділення, маючи машину, що виконувала ці операції. Такий принцип можна віднести як до побудови ієрархій категорій, так і до логічного представлення природномовних висловлювань (ПМВ). Так, якщо поняття "*Геша*" позначити як  $G$ , "*птаха*" – як  $B$ , то вираз "*Деякий Геша – птаха*" буде мати вигляд  $G = B \cdot x$ , а при позначенні, де  $T$  – "*пінгвін*" і  $F$  – "*літати*", посилка "*Жоден пінгвін не літає*" буде представлена як  $T \cdot x \neq F \cdot y$ , де  $x$  та  $y$  – деякі числа. Але завжди можливо підібрати такі  $x$  і  $y$ , щоб остання нерівність стала рівнянням, тому Лейбніц вводить позитивні числа для ствердження, але негативні для заперечення. Проте в такому варіанті неможливим є комбінування тверджень і заперечень: вираз "*Птахи літають за допомогою крил, але жоден пінгвін не літає*" неможливо представити в системі Лейбніца. Подальші його доповнення у систему, щоб запобігти такої неповноти, не дали бажаного результату. Тим більше, що твердження для досить складних об'єктів являли собою числа значного для обчислення порядку.

Однак результатом праці Лейбніца стала перша логічна система, заснована виключно на символах та математичних операціях і тому здатна для обчислення. Німецький філософ-математик Ламберт створює числення з операціями додавання, віднімання, множення та ділення і

відношеннями рівності і нерівності, позначаючи їх "+", "-", "×", "÷", "=", "≠" відповідно. При цьому він вводить аксіоми для цих операцій, серед яких  $A \times A = A$  (принцип ідемпотентності);  $1 + A = 1 - A$ , з якої витікає  $A + A = 0$  (виключаюче "або"). Така аксіоматика повністю узгоджується із бінарною алгеброю, прийнятою зараз за основу логіки. Але розробником такої алгебри став ірландський математик Буль. На основі принципу ідемпотентності у логіці  $AA = A$  він шукав корені рівняння  $AA = A$  або  $A^2 - A = 0$ . В такий спосіб Буль встановив, що в алгебрі логіки можуть існувати лише два значення: 1 та 0. Тому однією з аксіом його арифметики є  $1+1=0$ . І лише з часом 1 та 0 почали трактуватись як "істина" та "неправда" (як пропонувалось Луллієм), множення як логічне "і", додавання як "або" і т.п., інтегруючи досягнення схоластиків середньовіччя і математиків з епохи просвітництва до XIX ст. Найбільший внесок у таке об'єднання зробив Пірс. Він, зокрема, змінює виключаюче "або" на включаюче. Вершиною розвитку логіки висловлювань і алгебри логіки стала *повна аксіоматика* німецького філософа-математика Г. Фреге. Отже, логіка висловлювань представляється множиною формул. Для автоматичного виведення вона потребує арифметичного обчислення (з правилами арифметики логіки) значень істинності складових у формулах. Для природномовного тексту (ПМТ), оперуючи досить великими за обсягом частинами тексту, автоматичне виведення у логіці висловлювань є ефективним механізмом. Однак кожне висловлювання має свою внутрішню структуру, яка не може бути представлена за допомогою логіки висловлювань. Наприклад, атомарні речення "Птахи літають за допомогою крил" та "Пінгвіни не літають" не можуть бути семантично зв'язані навіть за наявності того факту, що "Пінгвіни є птахами".

Числення предикатів. Предикат (лат. Praedicatum) – присудок судження; те, що висловлюється (стверджується чи заперечується) у судженні про суб'єкт [6]. Предикативна форма ПМТ була помічена ще Аристотелем. Однією з найголовніших віх в історії предикатного представлення ПМВ стало введення Лейбніцем поняття символічного відношення зі змінними: вирішуючи проблему неспроможності силогічного представлення заключення про батька й сина (див. вище), він пропонує розглядати відношення  $R(A, B)$  – "A є батьком для B" та  $\tilde{R}(A, B)$  – "B є сином для A" такі, що  $R(A, B) = \tilde{R}(A, B)$ . Вперше частини висловлювань як одномісні предикати представляє німецький філософ Плуке. Але аналогічно логіці висловлювань найвагомійший внесок у числення предикатів зробив Фреге. "Він вважав її універсальною мовою, в якій можна б було представити систематично і математично точним способом будь-яку можливу форму раціонального мислення, яка могла б стати частиною дедуктивного мислення" [12]. Фреге вводить в логічне числення *квантор загальності* (разом з операціями імплікації та заперечення), причому його система є функціонально повною. Проте, вважаючи природномовну форму представлення знань причиною логічних протиріч, він розробляє власну двовимірну систему означень, яка, однак, не мала продовження, бо ускладнювала запис залученням другого виміру, не маючи при цьому наочності інших двовимірних систем представлення логічних знань (семантичні мережі, концептуальні графи тощо). Нижче представлено приклад зображення ПМВ "Птахи літають за допомогою крил" у нотації Фреге (рис. 1).

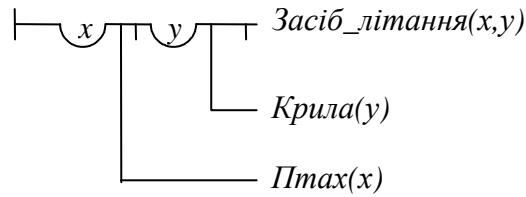


Рис. 1. ПМВ "Птахи літають за допомогою крил" у нотації Фреге

Як видно з рисунку, мала кількість первинних відношень потребує перефразування висловлювань у досить складну для запису і читання форму, які лише тоді можуть бути представлені у нотації Фреге. Форма представлення предикатів, як і логічних висловлювань, множина формул зі змінними, записаними лінійно. Спираючись на арифметичні закономірності, Пірс виділив *квантори загальності та існування*: взявши грецький символ  $\Pi$  для позначення багатократного логічного множення та символ  $\Sigma$  – для додавання, він помітив, що  $\Pi = 1$  при істинності (значення 1) всіх операндів, а  $\Sigma = 1$  – при наявності хоча б одного істинного (дорівнюючого 1) операнда. Таким чином, символ  $\Pi$  Пірс назвав квантором загальності, а символ  $\Sigma$  – квантором існування. Крім того, Пірс використовує знак " $<$ " (модифікований " $\leq$ ") для слідування, бо значення антицидента завжди менше чи дорівнює значенню консеквентна. Сучасне числення логіки предикатів (і висловлювань) наслідує систему операцій Буля, Фреге і Пірса, квантори Пірса та нотацію італійського математика Джузеппе Піано. Останній для уникнення плутанини при сумісному використанні логіки й арифметики замінив знаки логічних операцій: квантор існування (від лат. "Est") – на " $\exists$ ", слідування (від лат. "Consequentia") – на " $\supset$ ", диз'юнкцію (від лат. "Vel") – на " $\vee$ ", кон'юнкцію (від диз'юнкції) – на " $\wedge$ ", заперечення (від "-") – на " $\sim$ ", еквівалентність (від "=") – на " $\equiv$ " [19]. Квантор загальності " $\forall$ " (від нім. "Alle") був введений пізніше. Але тепер в основному використовується типова логіка предикатів, де замість конструкцій

$$(\forall x)(Птах(x) \supset Літає_за_допомогою_крил(x))$$

$$(\exists x)(Птах(x) \wedge Геша(x))$$

використовуються коротші та більш читабельні конструкції

$$(\forall x:Птах)(Літає_за_допомогою_крил(x))$$

$$(\exists x:Птах)(Геша(x)),$$

де запис " $(\forall x:Птах)$ " читається "для всіх  $x$  типу  $Птах$ ", або просто "для всіх птахів", а " $(\exists x:Птах)$ " – "деякий птах".

Отже, числення предикатів, як і логіка висловлювань, є повною [4, 15] системою ітеративної побудови та розбору логічних висловлювань. Однак атомарність предикатів, взагалі, є довільною, бо крім можливості повністю розібрати будь-яке ПМВ, наприклад, "Птахи літають за допомогою крил" (тобто, якщо птах літає, то знаяррдям польоту є крила):

$$(\forall b:Птах f:Полім)(\exists w:Крила)(Здійснює(b,f) \supset Знаяррдя(f,w))$$

існує можливість подальшої деталізації:

$$(\forall b:Птах f:Полім)(\exists s:Множина)(Належить(w,s) \supset Крило(w)) \wedge (Елементів_у_множині(s,2)) \wedge (Здійснює(b,f) \supset Знаяррдя(f,w)),$$

де пояснюється, що під словом "крила" слід розуміти наявність двох крил. Тобто предикатна модель представлення знань у ПМВ здатна представити будь-яке стверджувальне речення і навіть деякі модальності і часові відношення [13]. Автоматичне виведення знань передбачає попереднє переведення формул у специфічну форму (виведення методом резолюції, наприклад, потребує каузальної форми логічних формул [2]). Тому ефективність обчислення знань, представлених предикатною системою, нижче, ніж у висловлювань, і швидко зменшується при деталізації знань.

Екзистенційально-кон'юнктивна логіка. Подальший опис стосується варіантів описаних вище підмножин FOL. Порівняння цих варіантів підмножин FOL проводиться багатьма авторами робіт у галузі ІІС та ШІ ([2, 3, 19] тощо), проте специфікою даного порівняльного аналізу є орієнтація на ПМВ як вхідну форму представлення знань. Першою з підмножин, що розглядаються, є екзистенційально-кон'юнктивна логіка (ЕК логіка). Її назва пов'язана з логічним оператором кон'юнкції і квантором існування, що в ній використовуються. Найширше застосування ЕК логіка отримала із розробкою та втіленням реляційних баз даних (БД). Так, кожна БД служить для фіксації наявності деякого об'єкта (квантор існування) та переліку невід'ємних його властивостей (з'єднаних між собою кон'юнкцією). Усе, що не входить в БД, розглядається з логічної точки зору як неіснуюче. Таке обмеження дозволило створити одну з найуживаніших мов запитів – SQL (Structured Query Language) з відповідними ефективними механізмами виконання цих запитів, що зводяться до простого однопрохідного перебору. Однак очевидно є неповнота ЕК логіки як системи представлення знань. Так, загальний факт, що *"Птахи літають за допомогою крил"*, позначуваний в логічній системі з повним набором кванторів та операцій квантором загальності, потребував би внесення всіх птахів у базу з указанням для кожного із них факту спроможності літати за допомогою крил. Така ресурсоємна операція мала б повторюватись для всіх записів з кожною появою нової інформації про клас птахів. Більш того, номінально ЕК логіка не має можливості заявити заперечення, наприклад, що пінгвіни не літають. В останніх діалектах SQL доповнено можливість написання тригерів (процедури, що виконуються при істинності деякої події), й існує можливість перед доповненням в БД порції нової інформації, наприклад про пінгвінів, перевірити її та не включати як таку, що суперечить (тобто заперечує) існуючій інформації. Проте використання тригерів, по-перше, розширює SQL за межі ЕК логіки, по-друге, залишається неможливим представити інформацію з виключеннями експліцитно: *"Птахи літають за допомогою крил, але пінгвіни не літають"*, і по-третє, написання тригерів як виконуваної процедури відноситься до розробки БД, коли внесення факту *"пінгвіни не літають"* в систему з повною логікою є дією на рівні користувача, що потребує меншої професійності і, відповідно, менших матеріальних витрат.

Продукційні системи. За своєю формою продукційні системи дещо наслідують силогізми: система продукцій складається з правил  $A \supset B$  (*"якщо A, то B"*), де в загальному вигляді на місці  $A$  та  $B$  можуть стояти довільні логічні вирази без імплікації. Однак в автоматичних системах продукцій як  $B$  використовують одиночний факт чи посилання для простішої реалізації механізму прямого виведення (від антицедента до консеквента) [2]. Саме системи прямого виведення, в основному, називають продукційними, тоді як більшість системи зі зворотним виведенням є Хорновими (див. далі). Складається система продукцій (прямого і зворотного виведення) з фактів (предикати з константними предметами) та правил-продукцій. Механізмом обробки продукцій з прямим



виведенням можуть бути просте обчислення формул з предикатами (в т.ч. і продукцій) або ж специфічні процедури, орієнтовані на сталість структури  $A \supset B$ . Вагомим недоліком при використанні лише специфічних підходів для виведення є ускладнення перевірки несуперечності продукцій [2, 3] – для автоматизації цього процесу необхідне повноцінне обчислення предикатних формул, бо методи виведення не призначені для цього.

Диз'юнкти Хорна. Як уже зазначалось, продукційні системи зі зворотним виведенням у своїй більшості є Хорновими системами. Причиною цьому є використання диз'юнктивів Хорна – імплікативних формул  $A \supset B$ , де  $B$  є атомарним предикатом. Такий підхід дає низку переваг перед повною системою логіки предикатів: ефективне виведення (в загальному вигляді зі зростанням бази фактів часова складність зростає експоненціально, але в реальних оптимізованих системах варіативність пошуку значно обмежена [5]), проста схема виведення на основі загальнологічного правила уніфікації зведена до пошуку відповідностей (детальніше див. [2]), наділення логічних імплікацій більш зрозумілим для користувача процедурним смислом: імплікація  $(Має\_крила(x) \wedge Літає(x)) \supset Птаха(x)$

буде читатися не "З того, що  $x$  має крила й  $x$  літає, слідує, що  $x$  – птах", а " $x$  є птахом – це коли  $x$  має крила й  $x$  літає". Такі принципові переваги зробили мову логічного програмування Prolog, що застосовує логіку диз'юнктивів Хорна, однією з найвідоміших мов проектування й розробки баз знань по всьому світу. Попереднє ПМВ може бути записано у правилі Prolog як

```
bird(X) <- has_wings(X) & can_fly(X)
```

або в іншому діалекті

```
bird(X):- has_wings(X) , can_fly(X).
```

Проте, несуттєве для питально-відповідних систем та баз знань обмеження даної логіки [5] є визначальним при побудові систем представлення ПМВ з автоматизованою обробкою природномовних структур: вираз логіки предикатів

$$(Має\_крила(x) \wedge Літає(x)) \supset (Птаха(x) \wedge \sim Пінгвін(x))$$

відповідний природномовному "Птахи, окрім пінгвінів, мають крила й літають" є неприпустимим в Хорнових системах, отже далеко не всі природномовні структури можуть бути представлені в системах з логікою Хорна.

Фреймові системи та об'єктно-орієнтовані системи. "Термін *фрейм* (від англ. "frame") був запропонований Марвіном Мінським [9]... Ця модель ... має глибоке психологічне обґрунтування. Фрейм – це абстрактний образ для представлення деякого стереотипу сприйняття" [3]. Пропозиція Мінського стосувалась саме представлення знань. В загальному вигляді формально фрейм можна описати як структуру, яка включає ім'я фрейма, множину слотів зі своїми іменами і множину приєднаних процедур, зв'язаних із фреймом чи зі слотами [2]. Такі фрейми називають фреймами-прототипами. Фрейми-прикладі є фреймами-прототипами зі значеннями слотів та повинні відображати певні реальні об'єкти, ситуації тощо [2, 3]. З поняття фреймів виросло поняття об'єктно-орієнтованих (ОО) систем, що звузили трактування фреймів-прототипів до класів, а фреймів-прикладів – до об'єктів класу, зберігши повністю структуру й можливості фреймів. Наприклад, речення "Птахи літають за допомогою крил" може бути представлено фреймовою структурою як

Птах	
Спосіб пересування	Літати
Засіб пересування	Крила

чи у вигляді об'єкта класу в ОО мові програмування (наприклад, C++):

```
class Bird {
    const char *moving_method;
    const char *moving_tool;};
```

```
Bird some_bird;
```

```
some_bird.moving_method = "літати";
```

```
some_bird.moving_tool = "крила";
```

Крім того, сутність "*літати*" може розглядатись як процедура. В такому разі фрейм буде містити інші слоти й процедуру польоту, а клас – інші поля й метод польоту. У фреймових системах просто реалізується функція наслідування. Так, додавання до прикладу інформації про те, що "*Геша є птах*", може бути представлено як

Геша	
Фрейм_предок	Птах

чи

```
class Gesha:Bird {
    const char *name;};
Gesha gesha;
gesha.name = "Геша";
```

Проте, незважаючи на широке застосування фреймового представлення знань завдяки інтуїтивній зрозумілості принципу прототипів та прикладів, простоті реалізації наслідування атрибутів, високого ступеня структурованості й інших плюсів фреймових систем, логічною основою фреймів (без використання слотів-процедур) є ЕК логіка з усіма витікаючими з цього обмеженнями. Семантичні мережі й концептуальні графи. Семантичною мережею називається орієнтований граф з іменованими вершинами, що представляють об'єкти, та дугами – відношеннями між об'єктами. Як граф семантичні мережі використовують як мінімум два виміри для своєї нотації. Першою семантичною мережею було невелике дерево понять із відношеннями рід-вид, побудоване Порфірієм за категоріями Аристотеля [19]. Пізніше неодноразово будувались дерева категорій, проте вони використовували дуже обмежений набір відношень. Семантичні мережі отримали небувалого розвитку з можливістю їх автоматичної обробки на ЕОМ, з'являється величезна кількість використаних відношень: лінгвістичні (відмінкові, атрибутивні), логічні (кон'юнкція, диз'юнкція, заперечення, імплікація), квантифіковані (*усі, деякий*) [2]. Логічною основою семантичних мереж є повна логіка предикатів першого порядку – кожне відношення можна розглядати як предикат від суміжних йому (відношенню) вершин-об'єктів. З усього розмаїття семантичних мереж найбільшої уваги заслуговують *концептуальні графи* (КГ) [17]. Про це свідчить хоча б той факт, що для КГ комітетом NCITS T2 по інформаційному обміну й інтерпретації були прийняті ANSI-стандарти. КГ призначені для представлення знань на рівні людина-машина, хоча існує форма запису КГ, призначена для обміну графами між машинами. Для КГ визначені базові універсальні примітиви [17] для побудови семантичних мереж із довільними відношеннями. Вони мають можливість включати зображення, аудіо-інформацію та інші КГ як об'єкти (вершини графа).

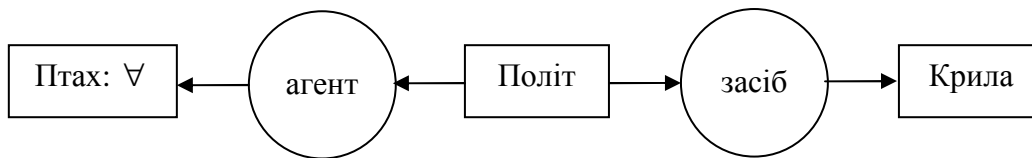


Рис. 2. КГ для речення "Птахи літають за допомогою крил"

Це робить їх максимально наближеними до природної мови, збільшуючи наочність представлення знань, простоту формування КГ з природномовних висловлювань, але й віддаляє таке представлення від формального, яке може бути обчислено. Можливим виходом є деяка модифікація КГ до потреб конкретної системи введенням обмежень на використовувані об'єкти й відношення та їх представлення. Як універсальна система представлення знань КГ не мають штатних механізмів виведення, парсерів та редакторів, що є вагомим недоліком їх використання як базової системи представлення й аналізу знань.

Мова формалізованого представлення та обміну знаннями KIF. Одночасно з КГ розроблено мову, спеціально призначену для обміну знаннями між гетерогенними системами баз знань і баз даних. Вона носить назву Формат обміну знаннями (Knowledge Interchange Format або KIF) і також регулюється ANSI-стандартами [18]. Семантика і логічна база мови KIF ідентична семантиці й логіці КГ, але описова виразність її менша: "щоб спростити аналіз комп'ютером, KIF має спрощений синтаксис та обмежену множину символів" [19]. Природномовне речення "Птахи літають за допомогою крил" у KIF буде мати вигляд

```

(forall (?b "Птах") (?f "Політ")
(exists (?w "Крила")
(=> ("Здійснює" ?b ?f)
("Знаряддя" ?f ?w))))
  
```

Як і КГ, KIF є універсальною мовою представлення та обробки знань, тому її доцільно адаптувати для вирішення конкретної задачі, дещо модернізуючи, але не вносячи змін, які суперечили б стандартам для KIF, що збереже можливість обміну представленими таким чином знаннями між системами з підтримкою KIF. Перевагою KIF над КГ є його лінійна текстова форма запису, подібна до LISP – інтерпретатори останнього мають змогу обробляти KIF-програми за наявності відповідних модулів, що роблять синтаксис KIF "зрозумілим" для LISP-інтерпретаторів [16].

Порівняльний аналіз підмножин FOL для представлення ПМВ. На основі зробленого опису підмножин FOL, у т.ч. з їх реалізаціями в діючих системах і нотаціях, можна зробити висновок, що вони дають змогу в тій чи іншій мірі представляти складні стверджувальні речення та питання до їх частин, замінюючи деякі компоненти представлення (силогізму, формули, графа) невідомою змінною, відповіді на які можуть бути отримані або рішенням рівняння, де значення формул із зміною має бути істиною, або пошуком фрагмента графа-питання в загальному графі [3], та встановленням значення невідомих вузлів чи дуг. ПМВ з модальними операторами або просторовими і часовими відношеннями можуть бути представлені в деяких з розглянутих підмножинах FOL але з певними змінами підходу до формалізації смислу ПМВ [13], або залученням інших, *некласичних логік*: модальна, часова, багатозначна, нечітка, інтуїціоністська тощо [4, 7]. Як механізм, описаний у [13], так і спеціальні квантори, операції, аксіоми та механізми виведення

некласичних логік і їх можливості й обмеження щодо представлення ПМВ потребують додаткового ґрунтового дослідження, що виходить за межі даної роботи.

На основі описаних вище переваг і недоліків розглянутих логічних систем та систем представлення знань можна провести їх *нестроге ранжування* [8] (від системи з мінімальними можливостями до системи з максимальними можливостями) за найвагомішими характеристиками з авторськими оцінками можливостей даних систем щодо використання для аналізу ПМВ. Список параметрів оцінки складений авторами виключно для вказаної задачі і не претендує на повноту й універсальність для задач представлення й аналізу знань у цілому.

1. *Описова виразність.* Найменше можливостей для представлення довільних ПМВ (лише певна множина логічних виведень) дають силогізми. Далі йде найбільш обмежена підмножина логіки предикатів – ЕК логіка та фреймові й ОО системи, що мають за основу цю підмножину (перелік характеристик довільного числа існуючих об'єктів). Повною за функціональністю є система логіки висловлювань, однак неможливість зв'язування частин висловлювань сильно обмежує можливості даної системи. Більш потужними є логічні системи предикатів з можливістю застосування імплікації (але з певними обмеженнями) – продукційні системи і диз'юнкти Хорна. Знімаючи обмеження на застосування операцій і додаючи квантор загальності, числення предикатів і, відповідно, KIF мають найбільшу описову виразність. Такі ж можливості представлення ПМВ надають КГ.

2. *Наочність представлення.* Маючи великий ступінь деталізації, значну кількість змінних для зв'язування формул і при цьому використовуючи математичні означення для операцій і кванторів, числення предикатів є найменш наочною системою представлення ПМВ. Зменшуючи кількість операцій і кванторів, ЕК логіка має більшу наочність, ніж повна система числення предикатів. Імплікативна форма представлення є більш природною для людини, ніж просте перелічення об'єктів з їх характеристиками, тому продукційні системи є наступними за наочністю. Можливість надання продукціям процедурного смислу збільшує наочність диз'юнктив Хорна. Завдяки застосуванню природномовних слів (з обмеженнями), KIF є більш наочним. Логіка висловлювань, на відміну від KIF, дозволяє застосовувати довільні слова і символи, тому її наочність є ще більша.

Маючи "глибоке психологічне обґрунтування" [3] стосовно людської схильності "представлення стереотипу сприйняття" [3], наступними за наочністю є фреймові системи. Будучи, фактично, набором природномовних речень, силогізми є найбільш наочними стосовно представлення ПМВ. Однак, оскільки графічне представлення не поступається мовному, КГ мають той же ступінь наочності.

3. *Простота трансляції з природної мови.* За причинами, вказаними для наочності представлення (п. 2), трансляція ПМВ у числення предикатів є найскладнішою. Практично такі ж проблеми присутні при переведенні ПМВ у KIF, окрім написання природною мовою кванторів ("forall", "exists") та операцій ("and", "or" тощо) – тому KIF займає наступне місце за простотою трансляції. Обмеження на операції і квантори спрощує задачу перекладу, звужуючи коло можливих варіантів представлення, тому наступним є ЕК логіка. Також, для ПМВ, що мають вигляд імплікацій, продукційні системи мають такі ж характеристики простоти трансляції, що й ЕК логіка. Завдяки

можливості розглядати диз'юнкції Хорна в процедурному сенсі, Хорнові системи мають додатковий механізм переведення ПМВ у формули і тому займають наступне за ЕК логікою і продукційними системами місце. З точки боку синтаксису, фрейми можуть представляти структуру речення за правилами взаємного розташування частин речення у ПМВ. Таким чином, маючи фреймове представлення можливих схем речень, задача переведення ПМВ у фреймовій системі спрощується. Трансляція у КГ може проводитись за правилами синтаксичного розбору речення, але без вимоги наявності попередньо складених схем речень. Обмеженість логіки висловлювань дає ще простіший механізм представлення у ній ПМВ – заміна зворотів на складнопідрядні речення і подальша заміна союзів відповідними логічними операторами. Відповідні за структурою силогізмам ПМВ можуть бути переведені в силогізми заміною середніх термінів на змінні, що є найпростішим з усіх можливих способів трансляції ПМВ у підмножини FOL, які розглядаються.

4. *Простота представлення в оперативній та постійній пам'яті ЕОМ.* Найскладнішими для представлення в пам'яті ЕОМ є структури, не передбачені архітектурою ЕОМ та можливостями мови програмування. Такими структурами є, зокрема, графові. Представлення ускладнюється відсутністю обмеження на типи вершин і зв'язків. Доволі складним є представлення силогізмів – просте текстове представлення не дає змоги пов'язувати вислови між собою та виокремлювати силогізми із множини інших. Використання текстової форми продукційних системи і систем диз'юнктів Хорна практично нівелює їх переваги стосовно простоти й ефективності виведення, тому питання про вибір структури представлення є відкритим (у Prolog, наприклад, використовуються специфічні графи [1]). Більш простою є задача представлення у пам'яті знань, що містяться у системі з ЕК логікою. Слідуючи принципам БД, таким представленням можуть бути блоки записів фіксованої довжини. Фрейми і ОО системи прямо відображаються структурами (*structure* в термінах С-подібних мов програмування) чи записами (*record* у Pascal) і класами в ОО мовах відповідно. Найпростішою є форма представлення у вигляді тексту, що є притаманною для логіки висловлювань, числення предикатів і KIF, проте відсутність в алфавіті KIF спеціальних символів (логічних кванторів і операцій) робить його представлення найпростішим з усіх.

5. *Ємність представлення в оперативній та постійній пам'яті ЕОМ.* Виходячи із об'єму знань, представлених силогізмами (одиничний висновок), маючи при цьому форму як мінімум трьох речень, силогізми є найбільш ємним способом представлення знань. Системи з ЕК логікою (переважно мова йдеться про БД), що використовують записи фіксованої довжини, також є доволі ємними. Логіка висловлювань зберігає тільки текст, необхідний для встановлення істинності ПМВ, проте, як зазначалось раніше, атомарні елементи логіки висловлювань є досить великими і потребують значних обсягів пам'яті для зберігання. Продукційні системи і системи диз'юнктів Хорна більш деталізовані, ніж логіка висловлювань, тому мають змогу виражати ПМВ у менш ємній формі. За можливості використання всіх операторів і кванторів у будь-якій формулі, числення предикатів є більш короткою нотацією, ніж продукції. Наступним у послідовності стоїть KIF, оскільки включає більшу кількість операцій (арифметика, робота зі списками тощо). Фреймові структури та КГ, маючи пряме відображення у пам'ять (впорядковані записи та дані вершин із посиланнями на дуги відповідно), потребують найменше пам'яті ЕОМ.

6. *Ефективність виведення.* Як уже зазначалось, виведення в системах з ЕК логікою зводиться до методу прямого повного перебору, який є найгіршим за своєю ефективністю. KIF потребує обчислення предикатних формул, що є ефективнішим за повний прямий перебір, але, на відміну від логіки предикатів, KIF має більше можливостей представлення (функції, визначення, мета-засоби тощо), а це потребує додаткових обчислень. Тому наступним є KIF, а далі – числення предикатів. Застосовуючи перебір фактів, використання продукційних систем і систем диз'юнктив Хорна дає змогу скоротити час виведення оптимізацією пошуку, в основі якої лежить обмеження даних логічних систем. Логіка висловлювань не включає кванторів, на відміну від числення предикатів, і оперує значними за обсягом елементами (цілісні елементарні ПМВ), тому містить менше операцій, що підлягають обчисленню. Виведення на КГ може бути зведено до обробки структури графа (пошук найкоротшого шляху, Ейлерового циклу тощо), для чого розроблені ефективні алгоритми. Система фреймів у загальному вигляді є графовою [2], проте більшість систем використовують ієрархічні структури фреймів (зокрема, ОО системи), що спрощує виведення. Виведення у силогізмах, як зазначалося раніше, може бути зроблено за одну ітерацію.

7. *Можливість декомпозиції.* Силогізми є цілісною і неподільною структурою, тому взагалі не підлягають декомпозиції. Логіка висловлювань може розбивати складні ПМВ на прості, не маючи змоги представлення їх внутрішньої структури. КГ, як і будь-які семантичні мережі, можуть бути розбиті на підграфи, однак виокремлення кожного підграфа розриває усі зовнішні для нього зв'язки, що значно ускладнює перевірку цілісності графа після декомпозиції. Декомпозиція систем продукцій і диз'юнктив Хорна простіша, але структура  $A \supset B$  є нерозривною. Фреймові структури мають аналогічні з КГ проблеми щодо декомпозиції, однак представлення у вигляді дерева практично знімає їх. Наступними ідуть числення предикатів і KIF, де декомпозиція здійснюється розбиттям на елементарні предикати, що можуть бути далі деталізовані записом їх як лямбда-виразів. Найпростішою є декомпозиція тверджень в ЕК логіці, де єдиним механізмом є розбір описів уже розділених між собою об'єктів по операції кон'юнкції.

8. *Масштабованість бази знань.* Додаючи новий об'єкт до КГ, як і до будь-якої семантичної мережі, необхідно проаналізувати можливість його зв'язування з усіма іншими об'єктами за усіма відношеннями, присутніми у графі, що є досить складною і ресурсоемною операцією. Масштабованість бази знань систем з ЕК логікою простіша, але потребує перевірки на повноту і несуперечність, що також вирішується виключно повним перебором серед наявних записів і існуючих відношень. Однак кількість останніх значно менша, ніж у КГ, тому ефективність і простота розширення бази знань систем з ЕК логікою вища, ніж у КГ. Наступними йдуть продукційні системи та системи диз'юнктив Хорна, недоліком яких при доповненні бази знань є складність перевірки несуперечності серед правил. Фреймові та ОО системи доповнюються додаванням нового фрейма у дерево ієрархії або додаванням нового поля до фрейма-предка. Найпростішим є доповнення бази знань систем силогізмів, логіки висловлювань, числення предикатів та KIF, бо єдина проблема суперечності нових знань із вже існуючим вирішується єдиним універсальним методом обробки знань у цих системах (силами користувача у силогізмах і прямим обчисленням в інших).

9. *Наявність загальних методів і засобів аналізу.* Єдина форма представлення знань, для якої не існує загальних методів і засобів аналізу, є КГ, бо при аналізі має враховуватись семантика відношень системи представлення, що неможливо через відсутність будь-яких обмежень для КГ у цьому плані.

На основі представленої вище ранжування можна поставити у відповідність кожній системі логічного представлення відповідні порядкові номери, починаючи з 1, для системи з мінімальними можливостями і до 9 – з максимальними, заносючи отримані відносні оцінки в таблицю.

Таблиця. Оцінки можливостей підмножин FOL

	Описова виразність для ПМВ	Наочність представлення	Простота трансляції з природної мови	Простота представлення в оперативній та постійній пам'яті EOM	Ємність представлення в оперативній та постійній пам'яті EOM	Ефективність виведення	Простота декомпозиції	Масштабованість бази знань	Наявність загальних методів і засобів аналізу
Силогізми	1	8	8	2	1	8	1	4	2
Логіка висловлювань	3	6	7	5	3	5	2	4	2
Числення предикатів	5	1	1	5	5	3	7	4	2
ЕК логіка	2	2	3	4	2	1	8	2	2
Продукційні системи	4	3	3	3	4	4	4	3	2
Диз'юнкти Хорна	4	4	4	3	4	4	4	3	2
Фреймові та ОО системи	2	7	5	6	7	7	5	5	2
Концептуальні графи	5	8	6	1	7	6	3	1	1
KIF	5	5	2	9	6	2	7	4	2

## 5. Висновки

Сукупний формальний аналіз табличних оцінок є проблематичним через складність встановлення кількісних взаємозв'язків між відносними оцінками, занесеними в кожний окремих стовпчик. Однак, за даними таблиці, можуть бути запропоновані певні рекомендації стосовно задачі представлення й аналізу знань, що містяться у ПМТ, а саме: найбільшою виразністю володіють три системи: числення предикатів, концептуальні графи та KIF. Це може слугувати критерієм вибору саме цих систем логіки як основних для представлення й аналізу знань. За іншими характеристиками можна запропонувати, що:

- представляти знання для користувача доцільно у КГ;
- трансляцію знань потрібно виконувати у три етапи (з ПМТ в КГ, з КГ в KIF, з KIF у числення предикатів);

- зберігати БЗ в оперативній та постійній пам'яті доцільно у KIF (об'єднуючи оцінки простоти і ємності представлення в пам'яті);
- вибір форми представлення для аналізу знань залежить від задачі аналізу (деякі задачі можна ефективно виконувати навігацією по КГ за алгоритмами роботи з графами, але низка задач не має методів і засобів їх вирішення на графах. В такому разі мають застосовуватись універсальні загальні методи логічного виведення на системі числення предикатів чи на KIF);
- обробку БЗ (доповнення й декомпозицію) доцільно проводити у формі числення предикатів або KIF.

Таким чином, у роботі описані найважливіші аспекти еволюційного розвитку формальних теорій логіко-онтологічного представлення знань, що містяться у природномовних текстах. Зокрема, представлено три фундаментальні методи генерування онтологічних категорій; описано найвагоміші віхи розвитку основних підмножин класичної логіки та варіантів їх реалізації у прикладних системах; вказано на можливості й обмеження даних підмножин і систем. На основі описаних можливостей логічних систем проведено їх порівняльний аналіз; відносні оцінки аналізу зведено в таблицю. За даними таблиці зроблено висновки щодо доцільності та можливої схеми використання систем логіки, які розглядаються.

Ефективний вибір та інтеграція логіко-онтологічного представлення знань суттєво впливають на результати, одержувані користувачем ЗОІС.

## СПИСОК ЛІТЕРАТУРИ

1. Братко И. Программирование на языке Пролог для искусственного интеллекта / Пер. с англ. – М.: Мир, 1990. – 560 с.
2. Гаврилов А.В. Системы искусственного интеллекта: методические указания для студентов заочной формы обучения АВТФ. – Новосибирск: Новосибирский государственный технический университет, 2004. – 59 с.
3. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2001. – 384 с.
4. Гуц А.К. Математическая логика и теория алгоритмов: Учебное пособие. – Омск: Издательство Наследие. Диалог-Сибирь, 2003. – 108 с.
5. Искусственный интеллект: история развития. – [www.library.by/shpargalka](http://www.library.by/shpargalka).
6. Кондаков Н.И. Логический словарь. – М.: Наука, 1999. – 656 с.
7. Логика. – <http://ru.wikipedia.org/wiki/логика.htm>.
8. Макаров И.М., Виноградская Т.М., Рубчинский А.А., Соколов В.Б. Теория выбора и принятия решений. – М.: Наука, 1982. – 326 с.
9. Минский М. Фреймы и представление знаний. – М.: Энергия, 1979. – 150 с.
10. Новиков Ф.А. Дискретная математика для программистов. – СПб.: Питер, 2001. – 301 с.
11. Палагін О.В., Петренко М.Г. Модель категоріального рівня мовно-онтологічної картини світу // Математичні машини і системи. – 2006. – № 3. – С. 91–104.
12. Робинзон Дж. Логическое программирование – прошлое, настоящее и будущее // Логическое программирование. – М.: Мир, 1988. – 368 с.
13. Рубашкин В.Ш. Представление и анализ смысла в интеллектуальных информационных системах. – М.: Наука, 1989. – 191 с.
14. Стяжин И.И. Становление идей математической логики. – М.: Наука, 1964. – 304 с.
15. Чёрч А. Введение в математическую логику. – М.: ИЛ, 1960. – 485 с.
16. <http://www-ksi.stanford.edu/htw/dme/thermal-kb-tour/kif-ontology.lisp.html>.
17. NCITS T2 (1998) Conceptual Graphs. A Presentation Language for Knowledge in Conceptual Schemas, Working draft of proposed American national standard, Document No. X3T2/96-008.
18. NCITS T2 (1998) Knowledge Interchange Format. Working draft of proposed American national standard, document (or available at <http://logic.stanford.edu/kif/dpans.html>).
19. Sowa, John F. Knowledge Representation: Logical, Philosophical and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove, CA, © 2000. – 594 p.