

УДК 51.681.3

*С.Л. Кривий*

## ПРИРОДНА МОВА І МАТЕМАТИЧНА ЛОГІКА В КОНТЕКСТІ ІДЕЙ В.М. ГЛУШКОВА

**Кривий Сергій Лук'янович**

Київський національний університет імені Тараса Шевченка,  
orcid: <https://orcid.org/0000-0003-4231-0691>

[sl.krivoi@gmail.com](mailto:sl.krivoi@gmail.com)

Розглядається підхід до побудови онтологоподібних систем для дослідження природомовних текстів з метою видобування з них знань, побудови бази знань для перевірки цих знань на суперечність/несуперечність на основі синтактико-семантичного аналізу текстів з використанням дескриптивних логік. Основна увага в роботі приділяється логічному аналізу добутих знань. Природномовний текст, з точки зору видобування знань, які в ньому знаходяться, та аналізу цих знань на сумісність (несуперечність), їх неповноту та істинність, складає одну з основних проблем побудови систем штучного інтелекту. Знання, акумульовані в природномовному тексті, представляються у формально-логічному вигляді, тобто у предикативній формі. Необхідність такого представлення полягає в тому, що знання може бути істинним, хибним, неповним або не мати ніякого зв'язку з реальністю. Основним завданням логічного аналізу є перевірка не істинності окремо взятих фактів, а їх несуперечності (сумісності) в цілому. Розглядаються загальні риси логічного аналізу знань, добутих з природномовного тексту. Ідея полягає в тому, щоб пов'язати результати і засоби лінгвістичного аналізу та формальної логічної мови. Цей зв'язок дозволяє розв'язати проблему перевірки сумісності знань, добутих з тексту, генерувати нові знання як наслідки з добутих фактів, поповнювати факти у разі їх неповноти тощо. На даний час одним з основних засобів реалізації такого підходу до аналізу виступають онтології та онтологічні бази знань. Розглядається окремий випадок онтологічної системи, у якій термінології є ациклічними, і алгоритм семантичного табло, орієнтований на такий тип онтології. Робота присвячується століттю з дня народження Віктора Михайловича Глушкова, ідеї якого знаходять своє втілення в житті.

**Ключові слова:** онтологія, дескриптивна логіка, синтактико-семантичний та логічний аналіз.

### Вступ

Вперше я зустрівся з Віктором Михайловичем Глушковым, будучи студентом Київського університету, у вересні 1969 року на лекції. У цей рік Віктор Михайлович починав читати студентам механіко-математичного факультету курс з теорії автоматів та її застосувань. Лекцію, прочитану Віктором Михайловичем, мені особисто записувати не було потреби, тому що після пари я міг повністю її записати з почутого на лекції. Лекція була прочитана так, що весь матеріал був викладений в такій логічній послідовності, що кожний факт був на своєму місці, зрозуміло було, чому цей факт повинен бути саме тут, яке прикладне значення він має. Далі моє знайомство з творчістю Віктора Михайловича продовжилось в 1975 році під час навчання в аспірантурі відділу «Теорії цифрових автоматів», яким він керував в Інституті кібернетики Академії наук України.

© С.Л. КРИВИЙ, 2023

Величезне навантаження і зайнятість Віктора Михайловича не дозволяли йому часто відвідувати наукові семінари, але коли він знаходив час, то це було надзвичайно цікаво. Особливо коли після доповіді Віктор Михайлович коментував результати виступів доповідачів, формулював нові задачі і проблеми даної тематики, підходи до розв'язання цих проблем та їх важливість для країни. У цих виступах вченого вражала ерудиція, широта мислення, обрії наукової фантазії та глибина розуміння проблематики. На одному з таких семінарів у 1976 році доповідалася робота групи лінгвістів, які представляли досягнення комп'ютерного перекладу текстів з однієї природної мови на іншу (це важлива область комп'ютерної лінгвістики і до сьогодні). Після закінчення виступу доповідачів Віктор Михайлович, характеризуючи результати доповіді, відзначав важливість розвитку даного напрямку досліджень для систем штучного інтелекту та величезну складність природної мови як об'єкта для формального аналізу змісту текстів природної мови. У чому складність такого аналізу та проблем, що виникають, і можливості автоматизації розглядаються в даній роботі, присвяченій світлій пам'яті Віктора Михайловича Глушкова.

Природномовний текст, з точки зору видобування знань, які в ньому знаходяться, та аналізу цих знань на сумісність (несуперечність), їх неповноту та істинність, складає одну з основних проблем побудови систем штучного інтелекту. Першочерговою проблемою є добування фактів (знань) з текстів, які стають об'єктом такого аналізу. Загальне означення поняття знання, яке наводиться в різних джерелах, формулюється у такому вигляді.

*Означення.* Знання — це філософська категорія для позначення упорядкованого в систему і підтвердженого практикою результату пізнання оточуючої нас дійсності.

Предметними формами існування знання, як відображення об'єктивної дійсності, є такі категорії: поняття (концепти), судження (міркування), гіпотези, теорії, факти тощо [1].

Знання поділяють на типи, серед яких розрізняють повсякденне знання, пов'язане з індивідуальним досвідом окремо взятої людини; художнє, що виражає специфіку художнього сприйняття людиною оточуючого світу; та наукове, яке відображає глибинні сутності процесів і явищ об'єктивної дійсності.

Важливою відмінністю наукового знання є його зв'язок з практикою, яка підтверджує або спростовує його істинність. У свою чергу цей тип знання поділяють на емпіричне та теоретичне. Обидва рівні відображають об'єктивні закономірні зв'язки процесів і явищ природи та суспільства. На емпіричному рівні ці зв'язки розкриваються шляхом аналізу безпосередніх даних спостереження, опису та експерименту. А на теоретичному розкривається загальний внутрішній характер, сутність і зміст пізнання процесів і явищ.

Що стосується знань, акумульованих в природномовному тексті, то вони повинні представлятися у формально-логічному вигляді, тобто у предикативній формі. Необхідність такого представлення полягає в тому, що знання може бути істинним, хибним, неповним (приклад 1) або не мати ніякого зв'язку з реальністю. Основним завданням логічного аналізу є перевірка не істинності окремо взятих фактів, а їх несуперечності (сумісності) в цілому. Розглянемо детальніше загальні риси логічного аналізу знань, добутих з природномовного тексту.

### **Формалізація понять**

Виходячи із завдань логічного аналізу, введемо певну формалізацію потрібних понять. Позначимо:

$T$  — сукупність природномовних текстів, вміст яких аналізується;

$CN$  — множина концептів (понять), добутих з  $T$ ;

$RN$  — множина (бінарних і унарних) відношень, заданих на множині  $CN$ ;

$TVox$  — множина термінологічних аксіом (термінологія)  $T$ ;

$ABox$  — множина фактів, добутих з  $T$ ;  
 $Ak$  — множина ознак елементів із  $CN$  і  $RN$ .

Зразу виникає низка питань:

- 1) яким чином добуваються знання із сукупності  $T$ ;
- 2) хто будує множини  $CN$  і  $RN$  та який вигляд вони мають;
- 3) яку логічну мову потрібно вибрати, щоб описати всі об'єкти;
- 4) які засоби логічного аналізу потрібно створити для перевірки сумісності (несумісності) множини фактів;
- 5) яким чином поповнювати факти в разі їх логічної неповноти;
- 6) звідки береться інтерпретація добутих знань.

Питання 1) і 2) відносяться здебільшого до системи семантико-синтаксичного і лінгвістичного аналізу, а питання 3)–6) — до системи логічного аналізу.

Перш за все нас цікавитиме логічний аналіз знань природномовного тексту, але коротко зупинимось і на засобах лінгвістичного аналізу, який у цьому аналізі відіграє ключову роль. Ідея полягає в тому, щоб пов'язати результати і засоби лінгвістичного аналізу та засоби формальної логічної мови. Цей зв'язок дозволяє розв'язати проблему перевірки сумісності знань, добутих з тексту, генерувати нові знання як наслідки з добутих фактів, поповнювати факти у разі їх неповноти тощо. На даний час одним з основних засобів реалізації такого підходу для аналізу виступають онтології та онтологічні бази знань (БЗ). Онтологічний підхід відповідає всім перерахованим вище завданням.

Створення онтологоподібної системи ґрунтується на понятті онтології, формальне означення якої має вигляд

$$O = (CN, RN, TBox, ABox, Ak), \quad (1)$$

де  $CN$  — скінченна множина концептів (понять), які фігурують в  $T$ ,  $RN$  — скінченна множина бінарних (семантичних) відношень, визначених на  $CN$ ,  $TBox$  — термінологічні аксіоми предметної області (ПО), до якої відноситься  $T$ ,  $ABox$  — множина фактів, які добуті з  $T$ ,  $Ak$  — специфічні властивості об'єктів із  $CN$  і  $RN$ .

При побудові онтології конкретизується ПО, до якої відносяться концепти із  $CN$  і відношення із  $RN$  та область інтерпретації  $\Delta$  з функцією інтерпретації  $f$ . Для інтерпретації вводиться позначення  $I = (\Delta, f)$ , і цю пару часто називають моделлю  $O$ . Конкретизація ПО необхідна для визначення функції інтерпретації  $f$  елементів із  $CN$  і  $RN$  на області  $\Delta$ .

Зв'язок функції інтерпретації  $f$  з ПО дозволяє вносити додаткові корективи до представлення об'єктів цієї ПО. Додаткові корективи описуються аксіомами  $Ak$  даної ПО і обмеженнями  $R_c$  та мають вигляд додаткових дефініцій (уточнень, обмежень значень тощо) властивостей із області інтерпретації  $\Delta$  даної ПО (які позначимо  $A(\Delta, R_c)$ ). Таким чином, приходимо до уточненого означення онтології для конкретної ПО:

$$O = (I = (\Delta, f), CN, RN, A(\Delta, R_c)). \quad (2)$$

В онтологічному підході інтерпретація  $I = (\Delta, f)$  орієнтована на теоретико-мнотинну модель, а означення (2) містить декларативні і процедурні засоби даної ПО, що дає можливість розв'язувати задачі користувача.

Розглянемо питання 1) — добування знань із сукупності  $T$ . Ця проблема розв'язується на етапі синтактико-семантичного лінгвістичного аналізу  $T$ . На цьому етапі формуються поняття (концепти), які фігурують в  $T$ , на їх підставі — термінологія ПО, до якої відноситься сукупність  $T$ , відношення між концептами, ознаки об'єктів ПО (а це множина  $Ak$ ) тощо.

Отже, результатом лінгвістичного аналізу, зокрема, повинні бути множини  $CN$  і  $RN$  та їх представлення у структурах даних, зручних для подальшого логічного аналізу.

З питанням 1) тісно пов'язане питання 2), оскільки складність  $T$  визначає формальну логічну мову, в якій записуються і обробляються знання, добуті з  $T$ . Як правило, мовою представлення знань та маніпуляції цими знаннями є мова предикатів першого порядку. Цей вибір не випадковий, підставою є те, що багато формальних логічних мов і, зокрема, мов сімейства дескриптивних логік (ДЛ) транслюються в мову предикатів першого порядку.

Звідси впливає відповідь на питання 3). Дійсно, якщо властивості концептів та відношень записуються на мові предикатів, то в цій мові застосовні метод резолюцій і метод семантичного табло для перевірки сумісності/несумісності знань та є можливість генерації наслідків з добутих знань.

Важливим і складним є питання 5). Справа в тому, що добуті з тексту знання, як правило, завжди неповні. Неповнота знань проявляє себе при генерації наслідків. Перевірка того, що деяка формула є наслідком заданої множини формул, не може бути проведена до кінця з огляду на брак деяких посилань. На разі немає загальних правил, яким чином ці посилання чи їх спростування знаходити. У деяких випадках такі посилання очевидні, в інших підказка може знаходитися в ПО (наприклад, посилання на якесь джерело, в якому даний факт доведено). Але загального розв'язання ця проблема не має.

Відповідь на питання 6) дає ПО, оскільки вона визначає об'єкти, які в ній фігурують, властивості цих об'єктів саме в цій ПО, а не в іншій (в іншій ПО ті самі назви об'єктів можуть мати зовсім іншу інтерпретацію).

Враховуючи теоретико-множинну орієнтацію інтерпретації об'єктів ПО, під концептом будемо розуміти зміст поняття (десигнат), об'ємом якого є множина таких об'єктів, які мають спільну властивість.

Судженням називається формулювання у вигляді предиката знань про об'єкти. У початковий момент аналізу суджень припускається, що всі наявні судження не обов'язково істинні, але на них можна опиратися як на аксіоми, сумнівність у правильності яких може бути встановлена тільки в результаті спільного розгляду.

У разі неповноти інформації додаткові судження, які називають гіпотезами, додаються експертом до заданої множини суджень. А коли встановлена суперечність початкової множини аксіом, то відбувається пошук джерела цієї суперечності, корекція множини аксіом і повторний аналіз.

Після етапу логічного аналізу настає етап побудови бази знань (БЗ), орієнтованої на дану ПО. Коли БЗ побудована, то вона розглядається як модель ПО, а логічна мова, в якій записані знання цієї БЗ, називається теорією БЗ. Тут уже маємо можливість генерувати запитання до БЗ і отримувати на них відповіді.

### Основні компоненти аналізу ПО

Зі сказаного вище впливає, що спочатку потрібно зафіксувати ПО. Необхідність цієї фіксації полягає в тому, що одні й ті самі поняття, як зазначалося, в різних ПО можуть мати різні семантичні значення.

На вхід системи аналізу подаються тексти із  $T$  (на даний час лише українською мовою, хоча передбачається подальший розвиток системи), які відносяться до даної ПО.

Областю інтерпретації концептів і відношень онтології виступає множина  $\Delta$ , в якій інтерпретується множина концептів  $CN$  і множина смислових відношень  $RN$ . Для множини  $CN$  область інтерпретації  $CN^f$  розбивається на класи (напри-

клад, імена власні, назви, фах тощо) [2]. Це розбиття є результатом синтактико-семантичного аналізу, який будує класи концептів за їх типами. Крім того, цей аналіз виконує побудову синтаксичних залежностей між членами речення у вигляді ациклічного орграфа. Синтаксичні залежності за певним відношенням між членами речення несуть деяку семантичну інформацію, яка використовується для виявлення семантичних ознак та потенційних смислових зв'язків між лексичними одиницями. Виявлення семантичних ознак відбувається не за єдиними правилами (їх просто не існує!), а залежить від мети аналізу, від дослідників, від уміння розробників.

**Приклад 1.** Нехай дано речення «Склянку борошна необхідно залити склянкою холодного молока». Синтаксичний аналіз дасть на виході такий ациклічний орграф (за відношенням «Хазяїн–Слуга») [4]:



Рис. 1

Семантичне представлення цього речення матиме такий вигляд:

1. КІЛЬК(ОДИН, СКЛЯНКА),                      2. КІЛЬК(ОДИН, БОРОШНО),
3. КІЛЬК(ОДИН, СКЛЯНКА),                      4. КІЛЬК(ОДИН, МОЛОКО),
5. ОЗНАК(ХОЛОДНИЙ, МОЛОКО),                6. МОДАЛ(НЕОБХІДНО, ЗАЛИТИ),
7. АГЕНТ(? , ЗАЛИТИ),                            8. ОБ'ЄКТ(БОРОШНО, ЗАЛИТИ),
9. КІНЦ-Т(ОДИН, ЗАЛИТИ),                      10. МЕТА(? , ЗАЛИТИ).

З цього семантичного представлення впливає розуміння речення приблизно такого змісту: «Борошно треба залити молоком, поки що невідомо ким і навіщо». Уточнення «навіщо» з тексту буде зрозумілим, але, можливо, потрібним буде його уточнення за допомогою відповідної дефініції.

Обмеження *Ак* якраз і служать для уточнення концептів із *CN* і вводять (наприклад, в діалозі з користувачем) нові концепти і відношення. Так, для атомарного концепта «молоко» уточнення «молоко холодне» потребує введення нового концепта «холодне», який виділяє специфічні властивості (ознаки) концепта «молоко» з множини значень його інтерпретації.

**Синтаксис мови концептів.** Для розробки представлення концептів із *CN* і *RN* пропонується такий синтаксис [3, 4]:

нехай *A* і *B* — довільні концепти із *CN* і *R* ∈ *RN*, тоді

*A, B* ::= *C* (атомарний концепт)

| *T* (універсальний концепт)

| *⊥* (пустий концепт)

| *¬A* (заперечення атомарного концепта)

| *C* ∩ *D* (перетин)

| *∀R.C* (обмежене значення)

| *∃R.T* (обмежений квантор існування).

Зауважимо, що заперечення застосовується лише до атомарних концептів, а для квантора існування областю його дії є універсальний концепт.

Даний синтаксис, як було сказано вище, орієнтується на теоретико-множинну семантику: нехай задана інтерпретація *I* = (*Δ*, *f*), тоді

- $\top^f = \Delta$ ,
- $\perp^f = \emptyset$ ,
- $(\neg C)^f = \Delta \setminus C^f$ ,
- $(C \cap B)^f = C^f \cap B^f$ ,
- $(\forall R.C)^f = \{a \in \Delta: \forall b (a, b) \in R^f \rightarrow b \in C^f\}$ ,
- $(\exists R.\top)^f = \{a \in \Delta: \exists b (a, b) \in R^f\}$ ,
- інтерпретація  $R^f \subseteq \Delta \times \Delta$ ,  $C^f \subseteq \Delta$ ,  $C$  — атомарний концепт,  $R$  — атомарне відношення.

Звідси випливають такі залежності :

- концепт  $A$  включається в концепт  $B$ , якщо  $A^f \subseteq B^f$ ;
- концепти  $A$  і  $B$  еквівалентні ( $A \equiv B$ ), якщо  $A^f = B^f$ ;
- об'єднання концептів  $C \cap B$  з інтерпретацією  $(C \cap B)^f = C^f \cap B^f$ ;
- повний квантор існування  $(\exists R.C)^f = \{a \in \Delta: \exists b (a, b) \in R^f \text{ і } b \in C^f\}$ ;
- числове обмеження  $\geq nR$  (не менше) і  $\leq nR$  (не більше) з інтерпретацією

$$\geq nR = \{a \in \Delta: |\{b:(a, b) \in R^f\}| \geq n\},$$

$$\leq nR = \{a \in \Delta: |\{b:(a, b) \in R^f\}| \leq n\};$$

- заперечення довільного концепта  $\neg C$  з інтерпретацією  $(\neg C)^f = \Delta \setminus C^f$ .

Зауважимо, що об'єднання і квантор існування можна було б і не вводити, а скористатися законами  $C \cup B = \neg(\neg C \cap \neg B)$  і  $\exists R.C = \neg \forall R.\neg C$ .

Зафіксувавши такий синтаксис і семантику, дістаємо дескриптивну логіку, яка отримала назву ALC (attributive language with complementation).

Розглянемо приклад термінології в ALC-логіці.

**Приклад 2.** Нехай атомарними концептами є «особа, житло, номер», а атомарним унарним відношенням є «має». Тоді «житло  $\cap$  особа» означає житло людини, а «житло  $\cap \neg$  особа» означає житло без людей.

Використовуючи ці атомарні об'єкти, будуюмо термінологію ПО «адреса особи»:

кімната-жила  $\equiv$  житло  $\cap$  особа  
 кімната-нежила  $\equiv$  житло  $\cap \neg$  особа  
 квартира  $\equiv \geq 1$  має.кімната-жила  
 номер  $\equiv 1...100$   
 будинок  $\equiv n$  має.номер  $\cap m$  має.квартира  
 жінка  $\equiv$  особа  
 чоловік  $\equiv$  особа  $\cap \neg$  жінка  
 дружина  $\equiv$  жінка  $\cap$  має.чоловік  
 чоловік-жонатий  $\equiv$  чоловік  $\cap$  має.дружину  
 подружжя  $\equiv$  чоловік-жонатий  $\cap$  дружина  
 мешкання-подружжя  $\equiv$  подружжя  $\cap$  має.квартира  
 адреса-подружжя  $\equiv$  (будинок  $\cap$  номер)  $\cap$  (поверх  $\cap$  номер)  $\cap$  (квартира  $\cap$  номер)

Це дещо спрощений варіант ПО «адреса особи», тому що не вказано місто, де живе подружжя чи окремо взята особа, вулиця та країна. Цю термінологію легко поповнити відповідними концептами.

### Об'єкти області і інтерпретації

Оскільки реальні об'єкти онтосистеми належать області інтерпретації  $\Delta$ , то необхідно описати світ, в якому вони інтерпретуються. Цей опис представляється в термінах концептів і бінарних відношень, які беруться з області інтерпретації  $\Delta$ . Деякі атомарні концепти і відношення можуть визначатися як термінологічні іме-

на, тоді можна вводити індивідууми шляхом присвоєння їм імен з описом властивостей цих індивідуумів. Нехай  $a, b, c, \dots$  — індивідууми, тоді, користуючись концептом  $C$  і відношенням  $R$ , можна побудувати властивості таких типів:

$$a : C \text{ і } R(b, c),$$

де  $a : C$  означає властивість « $a$  належить концепту  $C$ », а  $R(b, c)$  — властивість відношення, яке означає, що «елементи  $c$  — наповнення відношення  $R$  для  $b$ ». Наприклад, якщо атомарними концептами виступає  $B = \text{ІМ'Я}$ ,  $C = \text{ФАХ}$ , атомарним бінарним відношенням є  $R = \text{БАТЬКО}$ , а іменами індивідуумів в  $\Delta$  є Петро, Іван, Степан, то інтерпретацією концепта  $B^f$  є множина  $\{\text{Петро, Іван, Степан}\}$ ,  $C^f = \{\text{лінгвіст, лікар, програміст}\}$ , а  $R^f = \{(\text{Петро, Іван}), (\text{Петро, Степан})\}$ . Після цього «ЛІНГВІСТ(Петро)» означає, що Петро — фахівець з лінгвістики, а «БАТЬКО(Петро, Іван)» означає, що Петро — батько Івана.

Таким чином, з точки зору користувача,  $\Delta$  виглядає як об'єкт реляційної бази даних (БД), який складається лише з унарних та бінарних відношень. Різниця полягає в тому, що в класичних БД відношення замкнуті в рамках даної ПО, в той час як  $\Delta$  як база знань відкрита для даної ПО, тому що нормалізоване представлення знань виконується в ситуації, коли немає припущення про повноту цих знань. Крім того,  $\Delta$  зв'язує семантичними відношеннями концепти  $C \in CN$  і відношення  $R \in RN$ , що немає аналогу в семантиці реляційних БД.

#### Етап логічного аналізу знань

Для виконання логічного аналізу фактів заданої онтосистеми користуються, з одного боку, можливістю трансляції логіки ALC у деякий фрагмент мови предикатів першого порядку, а з другого боку розробляють алгоритми для самої логіки ALC. Розглянемо коротко кожен з можливостей.

**ALC як фрагмент логіки предикатів.** Семантика мови концептів, яка наведена вище, говорить про те, що вона є фрагментом мови предикатів першого порядку. Оскільки інтерпретація є теоретико-модельною семантикою, то на інтерпретовані концепти і відношення можна дивитися як на унарні і бінарні предикати, визначені на множині  $\Delta$ . А це дає можливість транслювати у формули логіки предикатів  $f_C(x)$  атомарні концепти з однією вільною змінною  $x$ , такою, що для довільної інтерпретації  $I = (\Delta, f)$  множина елементів із  $\Delta$ , які виконують  $f_C(x)$ , збігається з множиною  $C^f$ . А конструктори  $\cup, \cap, \neg$  — у логічні диз'юнкцію, кон'юнкцію та заперечення.

Якщо  $C$  вже транслюване у формулу  $\varphi_C(x)$  і  $R$  — атомарне бінарне відношення, то квантори існування і загальності виражаються формулами

$$\varphi_{\exists R.C}(y) = \exists x. R(y, x) \ \& \ \varphi_C(x), \quad \varphi_{\forall R.C}(y) = \forall x. R(y, x) \rightarrow \varphi_C(x),$$

де  $y$  — нова змінна, а числові обмеження значень виражаються формулами

$$\varphi_{\geq n R}(x) = \exists y_1, \dots, \exists y_n. R(x, y_1) \ \& \ \dots \ \& \ R(x, y_n) \ \& \ \bigwedge_{i < j} y_i \neq y_j,$$

$$\varphi_{\leq n R}(x) = \forall y_1, \dots, \forall y_{n+1}. R(x, y_1) \ \& \ \dots \ \& \ R(x, y_{n+1}) \rightarrow \bigwedge_{i < j} y_i = y_j.$$

Предикат рівності « $=$ » необхідний, оскільки за його допомогою виражаються числові обмеження. Але оскільки концепти транслюються в логіку предикатів, то вводити спеціальний синтаксис немає потреби.

Розглянемо ще одну можливість логічного аналізу БЗ алгоритмом семантичного табло (СТ).

**Алгоритм СТ для ALC.** Перш ніж навести алгоритм і довести, що він розв'язує проблему виведення в логіці ALC, дамо деякі формальні означення загального характеру.

Для довільної математичної логіки, після того як були визначені її синтаксис і семантика, намагаються будувати числення, в якому виводяться тільки істинні висловлювання в семантиці цієї логіки. Істинність формули  $\varphi$  позначається  $\models \varphi$ , а те, що ця формула виводиться в численні, —  $\vdash \varphi$ . Тоді числення називається коректним, якщо для  $\forall \varphi$  із  $\vdash \varphi$  випливає  $\models \varphi$ ; числення називається повним, якщо для  $\forall \varphi$  із  $\models \varphi$  випливає  $\vdash \varphi$ .

Словами це означає, що в коректному численні виводяться тільки істинні формули, а в повному — всі істинні формули.

Таке означення справедливе для класичних і деяких некласичних логік, але для ALC це означення не підходить тому, що для неї відсутнє числення (відсутні правила виведення)\*.

Замість цього для заданої ДЛ намагаються побудувати алгоритм, який перевіряє істинність аксіом і виконуваність концептів. Нехай зафіксовано ДЛ  $L$  і потрібно побудувати алгоритм  $U$ , який перевіряє включення заданих концептів  $C \subseteq D$  в заданій термінології  $T$ , де  $C, D, T$  формулюються в мові  $L$ . Нехай  $U(C, D, T)$  означає відповідь алгоритму  $U$  на вхідних даних  $C, D, T$ . Будемо вважати, що цей алгоритм працює в скінченному часі і видає лише відповіді 0 і 1. Тоді алгоритм  $U$  вважається коректним, якщо для довільних  $C, D, T$  із  $U(C, D, T)=1$  випливає  $T \models C \subseteq D$ ; повним, якщо для довільних  $C, D, T$  із  $T \models C \subseteq D$  випливає  $U(C, D, T)=1$ .

Оскільки в ДЛ розробляються алгоритми для перевірки виконуваності концептів, то включення концептів  $C \subseteq D$  еквівалентно невиконуваності концепта  $C \cap \neg D$ . У цьому разі відповіді алгоритму  $U$  будуть протилежні тим, які генерувалися при перевірці включення. Тому для проблеми виконуваності вище наведені поняття міняються місцями.

Алгоритм  $U$  розв'язує проблему виконуваності концептів в термінології  $TBox$  для ДЛ  $L$ , якщо виконані такі умови.

**Термінальність:** для довільних  $(C, T)$  алгоритм  $U$  генерує відповідь  $U(C, T)$  в скінченному часі.

**Коректність:** для довільних  $(C, T)$ , якщо  $C$  виконується в термінології  $T$ , —  $U(C, T) = 1$ .

**Повнота:** для довільних  $(C, T)$ , якщо  $U(C, T) = 1$ , концепт  $C$  виконується в  $T$ .

Термінології можуть бути ациклічними і з циклами. Для обох типів термінологій розроблені алгоритми СТ [4]. Далі мова йтиме про ДЛ з ациклічними термінологіями, а інтерпретація називається базовою, якщо вона інтерпретує лише концепти із  $CN$  та ролі із  $RN$ .

Найважливішою властивістю ациклічних термінологій є можливість їх вилучення: проблема виконуваності концептів відносно ациклічної термінології зводиться до проблеми виконуваності концептів без термінологій.

**Теорема 1** (вилучення ациклічних термінологій). *Якщо  $TBox$  ациклічна, то для довільного концепта  $C$  справедлива еквівалентність:  $C$  виконується в  $TBox$  тоді і тільки тоді, коли концепт  $C_T$  виконується.*

**Доведення.** Нехай  $I$  — модель, в якій  $C^I \neq \emptyset$ . Звідси маємо  $\models C \equiv C_T$ , і тому  $C_T^I = C^I \neq \emptyset$ . А це означає виконуваність концепта  $C_T$ .

---

\* Для логіки ALC існує аксіоматична система, наведена в [5]. Вона включає схеми аксіом числення висловлювань, дві додаткові схеми аксіом  $\forall R.(\alpha \cap \beta) \equiv \forall R.\alpha \cap \forall R.\beta$  і  $\forall R.\top \equiv \top$  та одне додаткове правило виведення  $\alpha \vdash \forall R.\alpha$ . Отримана аксіоматична система несуперечна та повна [6].

Нехай  $C_T$  виконується, і нехай  $J$  — така інтерпретація, що  $C_T^J \neq \emptyset$ . Можна вважати, що  $J$  — базова інтерпретація, оскільки в  $C_T$  зустрічаються лише базові концепти. Оскільки  $T$  ациклічна, то вона коректна, і тому існує єдине розширення  $I$  інтерпретації  $J$ , яка є моделлю для  $T$ . Оскільки  $I$  і  $J$  збігаються на базових концептах, то  $C_T^J \neq \emptyset$ . Очевидно, що  $T \models C \equiv C_T$ , а це означає, що  $C^I = C_T^I \neq \emptyset$ . Отже, концепт  $C$  виконується в  $T$ .

Ця теорема дає можливість стверджувати, що із розв'язуваності якої-небудь ДЛ автоматично випливає розв'язуваність цієї ж ДЛ з ациклічними термінологіями. Але обчислювальна складність ДЛ може стати експоненціальною від довжини початкового концепта, що негативно впливає на ефективність. Проте практика показує, що для багатьох логік складність ациклічних термінологій така сама, як і без термінологій.

**Алгоритм СТ для ALC без термінологій.** Нехай задано концепт  $C_0$  логіки ALC, для якого необхідно перевірити виконуваність. Без обмеження загальності вважатимемо, що  $C_0$  нормалізований, тобто всі символи заперечення в ньому стоять перед атомарними концептами. Алгоритм, який називатиметься SAT, будує початковий ABox  $A_0 = \{x_0 : C_0\}$ .

Далі на кожному кроці до поточного ABox  $A$  застосовується правило, в результаті дістаємо один або два нових ABox. Список правил наведений в таблиці. Порядок застосування правил довільний, оскільки для логіки ALC правильність роботи алгоритму не залежить від порядку застосування правил (це дає широкий вибір стратегії застосування правил з метою оптимізації). Правила можна поділити на два класи: правила типу кон'юнкції і правила типу диз'юнкції. Перші породжують один новий ABox  $A'$ , другі — два нових ABox  $A'$  і  $A''$ . Далі правила застосовуються до кожного з них незалежно.

Таблиця

| Правило            | Умови застосування                                                                 | Дія                                                          |
|--------------------|------------------------------------------------------------------------------------|--------------------------------------------------------------|
| $\cap$ -правило    | якщо 1. $x : (C \cap D) \in A$<br>2. $x : C \notin A$ або $x : D \notin A$         | то $A' = A \cup \{x : C, x : D\}$                            |
| $\cup$ -правило    | якщо 1. $x : (C \cup D) \in A$<br>2. $x : C \notin A$ і $x : D \notin A$           | то $A' = A \cup \{x : C\}$<br>$A'' = A \cup \{x : D\}$       |
| $\exists$ -правило | якщо 1. $x : \exists R.C \in A$<br>2. $\neg \exists y : xRy \in A$ і $y : C \in A$ | то створити нову точку $y$ і<br>$A' = A \cup \{xRy, y : C\}$ |
| $\forall$ -правило | якщо 1. $x : \forall R.C \in A$<br>2. $\exists y : xRy \in A$ і $y : C \notin A$   | то $A' = A \cup \{y : C\}$                                   |

Як впливає з алгоритму SAT, він будує бінарне дерево пошуку моделі, на якій виконується  $A_0$ . Робота алгоритму закінчується, якщо до отриманого ABox  $A$  або не застосовне жодне правило, або є суперечність і продовжувати роботу немає сенсу. Введемо терміни:

- ABox  $A$  називається несуперечним, якщо не має фактів  $x : A$  і  $x : \neg A$  для деякого індивіда  $x$  і атомарного концепта  $A$  або включає факт  $x : \perp$  для деякого індивіда  $x$ ;
- ABox  $A$  називається повним несуперечним, якщо він несуперечний і жодне з правил алгоритму SAT не застосовне.

Якщо алгоритм побудував повний несуперечний ABox  $A$ , то він генерує відповідь 1, що означає існування моделі, а отже, і виконуваність  $A_0$ . Після цього алгоритм закінчує роботу і решту гілок дерева не обходить. Якщо він побудував суперечний ABox, то це означає лише, що на цій гілці моделі не існує і алгоритм

продовжує свою роботу далі. Коли алгоритм обійшов всі гілки дерева і виявилось, що всі листки дерева суперечні, то алгоритм видає відповідь 0, що означає невиконуваність  $ABox A_0$ .

#### Алгоритм SAT(A) для логіки ALC

Вхід: нормалізований  $ABox A$ .

Вихід: значення 1 (виконується) або 0 (суперечний).

Метод:

якщо в  $A \in x : \perp$  для деякого  $x$ , то return(0)\*.

якщо в  $A \in x : A$  і  $x : \neg A$  для деякого  $x$  і  $A$ , то return(0).

якщо до  $A$  не застосоване жодне правило, то return(1).

якщо застосоване правило  $\cap$ ,  $\exists$  або  $\forall$ ,

то застосувати довільне з них, діставши  $A'$ , і return(SAT( $A'$ )).

якщо застосоване правило  $\cup$ , то застосувати його, діставши  $A'$  і  $A''$ , і виконати

(якщо SAT( $A'$ ) = 1, то return(1);

якщо SAT( $A''$ ) = 1, то return(1));

return(0).

Алгоритм SAT(A) отримує на вхід довільний нормалізований  $ABox A$  і генерує відповідь 0 або 1. Покажемо, що цей алгоритм розв'язує проблему виконуваності концептів для логіки ALC. Кожний  $ABox A$  дамо у вигляді розміченого графа  $(G, L)$ . Вершинами графа  $G$  є індивіди  $x$ , які зустрічаються в  $A$ , а ребрам відповідають факти в  $A$  типу  $xRy$ . Позначкою  $L(x)$  вершини  $x$  є множина таких концептів  $C$ , що в  $A$  є факт  $x:C$ , тобто  $L(x) = \{C \mid x:C \in A\}$ . Для повного несуперечного  $ABox$  цей граф представляє шукану модель.

**Лема 1.** *Не існує нескінченного ланцюга  $A_0, A_1, \dots$ , в якому кожний  $ABox A_{i+1}$  отриманий з  $A_i$  за яким-небудь правилом алгоритму SAT.*

Доведення випливає з монотонності правил, які застосовуються в процесі пошуку моделі. Ці правила тільки додають елементи до  $ABox$  і нічого не вилучають. Покажемо обмеженість росту  $ABox$ . Для цього розглянемо декілька тверджень.

- Граф  $(G, L)$  є деревом з коренем  $x_0$ , оскільки на початку граф складається з єдиної вершини, а нові вершини породжуються лише  $\exists$ -правилом. Це правило з'єднує нову вершину з однією з існуючих вершин. Отже, з кореня  $x_0$  в довільну іншу вершину  $x \neq x_0$  веде єдина послідовність дуг. Довжина цієї послідовності називається рівнем вершини  $x$ .

- Ширина  $w$  дерева  $(G, L)$  обмежена числом кванторів  $\exists$  в початковому концепті. Дійсно, дуги, які виходять з якої-небудь вершини, виникають у результаті застосування  $\exists$ -правил, які відповідають концептам типу  $\exists R.C$  і зустрічаються в  $C_0$ .

- Глибина  $h$  дерева  $(G, L)$  обмежена кванторною складністю, тобто максимальною величиною вкладеності в ньому кванторів. Це випливає з того, що кванторна складність концептів в позначках вершин зменшується зі збільшенням рівня вершин. Формально це виглядає так: якщо  $x : C \in A$  і  $x$  має рівень  $L$ , то  $h(C) \leq h - 1$ . Це доводиться індукцією за числом  $l$ . База індукції очевидна. Крок індукції випливає з того, як сформульовані правила  $\exists$  і  $\forall$ : якщо на деякому кроці були концепти типу  $\exists R.C$  і  $\forall R.C$ , то ці правила на наступний рівень переносять лише концепт  $C$ , складність якого на одиницю менша. Отже, загальне число вершин в графі  $(G, L)$  не перевищує величини  $1 + w + w^2 + \dots + w^h$ .

- Позначка кожної вершини обмежена:  $L(x) \subseteq Sb(C_0)$ , де  $Sb(C_0)$  — множина підконцептів концепта  $C_0$ . Дійсно, довільне правило додає до позначки довільної вершини лише підконцепти вже існуючих там концептів.

Отже, розмір графа  $(G, L)$  обмежений зверху, а це означає його скінченність.

\* Виконання оператора return призводить до зупинки роботи алгоритму.

**Лема 2.** *Мають місце такі твердження:*

а) *концепт  $C_0$  виконуваний тоді і тільки тоді, коли  $ABox A_0 = \{x_0 : C_0\}$  виконуваний;*

б) *нехай  $A'$  отриманий з  $A$  за одним із правил  $\cap$ ,  $\exists$ ,  $\forall$ . Тоді коли  $A$  виконується, то  $A'$  виконується;*

в) *нехай  $A'$  і  $A''$  отримані за  $\cup$ -правилом. Тоді коли  $A$  виконується, то  $A'$  або  $A''$  виконується.*

*Доведення.* Перший пункт очевидний. А доведення пунктів б) і в) досить прості, тому доведемо лише твердження для  $\exists$ - і  $\forall$ -правил.

Нехай  $A'$  отриманий з  $A$  за  $\exists$ -правилом. Якщо  $ABox A$  виконується, то існує його модель  $I$ . На підставі семантики  $\exists$ -правила маємо  $x^I \in (\exists R.C)^I$ , а це на підставі першої частини правила означає, що існує  $d \in \Delta$ , такий, що  $(x^I, d) \in R^I$  і  $d \in C^I$ . Щоб побудувати модель для  $A'$ , розширимо  $I$ , покладаючи  $y^I := d$ . Тоді  $x^I, y^I \in R^I$  і  $y^I \in C^I$ . Отже,  $I \models xRy$  і  $I \models y : C$ , тобто  $I$  — модель для  $A'$ .

Нехай  $A'$  отриманий з  $A$  за  $\forall$ -правилом. Якщо  $ABox A$  виконується, то існує його модель  $I$ . Тоді на підставі другої частини  $\forall$ -правила маємо  $x^I \in (\forall R.C)^I$  і  $(x^I, y^I) \in R^I$ . Звідси випливає, що для довільного елемента  $d \in \Delta$ , такого, що  $(x^I, d) \in R^I$ , має місце  $d \in C^I$ . Зокрема, це має місце і для  $d = y^I$ , тому  $y^I \in C^I$ . Отже,  $I \models y : C$ , а це означає, що  $I$  є моделлю для  $A'$ .

**Лема 3.** *Довільний повний несуперечний  $ABox$  виконуваний.*

*Доведення.* Нехай  $ABox A$  повний і несуперечний. Побудуємо його канонічну модель  $I$ :

- $\Delta := \{x \mid \text{індивід } x \text{ зустрічається в } A\}$ ;
- $A^I := \{x \in \Delta \mid x:A \in A\}$  для кожного атомарного концепта  $A$ ;
- $R^I := \{(x, y) \in \Delta \times \Delta \mid xRy \in A\}$  для кожної атомарної ролі  $R$ .

Покажемо, що  $I \models A$ , тобто  $I \models \alpha$  для кожного факту  $\alpha \in A$ . Для фактів вигляду  $xRy$  це випливає безпосередньо з побудови  $R^I$ . Для фактів вигляду  $x : C$  індукцією за побудовою концепта  $C$  доведемо справедливість імплікації

$$\forall x \in \Delta (x : C \in A \rightarrow x \in C^I).$$

Оскільки всі концепти в  $ABox$  нормалізовані, то довільний концепт побудований з концептів вигляду  $\top$ ,  $\perp$ ,  $A$ ,  $\neg A$ , де  $A$  — атомарний концепт, за допомогою зв'язок  $\cap$ ,  $\cup$ ,  $\exists$ ,  $\forall$ .

**База індукцій:**

• очевидно, для  $\perp$  посилка імплікації хибна, а для  $\top$  висновок імплікації істинний;

• якщо  $x : A \in A$ , то за побудовою  $A^I$  маємо  $x \in A^I$ ;

• якщо  $x : \neg A \in A$ , то  $x \in (\neg A)^I$ , оскільки в протилежному випадку, якщо  $x \in A^I$ , то з означення  $A^I$  випливало б  $x:A \in A$ , а разом з  $x : \neg A \in A$  це означало б, що  $A$  суперечний. Дістаємо суперечність з умовою леми.

**Крок індукції.** Випадки зв'язок  $\cap$  і  $\cup$  досить прості, тому розглянемо лише квантори.

• Випадок  $C = \exists R.D$ . Нехай  $x:C \in A$ , тобто  $x : \exists R.D \in A$ . Але тоді виконується перша посилка  $\exists$ -правила. Оскільки  $ABox A$  повний, то  $\exists$ -правило до нього не застосовне, а це означає невиконуваність його другої посилки. Тобто існує такий  $y \in \Delta$ , що  $xRy \in A$  і  $y : D \in A$ . Звідси дістаємо  $(x, y) \in R^I$  за побудовою  $R^I$ , а також  $y \in D^I$  за припущенням індукції для  $D$ . Отже, на підставі означення семантики для квантора  $\exists$  отримуємо  $x \in (\exists R.D)^I$ .

• Випадок  $C = \forall R.D$ . Нехай  $x : C \in A$ , тобто  $x : \forall R.D \in A$ . Але тоді виконується перша посилка  $\forall$ -правила. Щоб довести, що  $x \in (\forall R.D)^I$ , візьмемо довільний  $y \in \Delta$ , такий, що  $(x, y) \in R^I$ , і покажемо, що  $y \in D^I$ . За припущенням індукції для  $R$  маємо  $xRy \in A$ . Оскільки  $ABox A$  повний, то  $\forall$ -правило до нього не застосовне, а це означає невиконуваність його другої посилки, тобто має місце  $y : D \in A$ . Звідси за припущенням індукції для  $D$  впливає  $y \in D^I$ , що і потрібно було показати.

Отже, побудовано модель  $I ABox A$ , звідки впливає виконуваність  $A$ .

**Теорема 2.** Проблема виконуваності концептів логіки  $ALC$  розв'язувана. Алгоритм  $SAT$  є процедурою розв'язуваності для цієї проблеми.

Доведення теореми впливає з вищенаведених лем.

**Теорема 3.** Проблема виконуваності  $ABox$  в логіці  $ALC$  розв'язувана.

**Приклад 3.** Перевірити справедливості включення атомарних концептів і ролей над областю інтерпретації, де множинами атомарних концептів і атомарних ролей виступають

$CN = \{C_1 = \text{унів}, C_2 = \text{факт}, C_3 = \text{каф}, C_4 = \text{студенти}, C_5 = \text{жін-ст}, C_6 = \text{рік}\};$

$RN = \{R_1 = \text{маєу}, R_2 = \text{маєф}, R_3 = \text{маєкф}, R_4 = \text{маєст}, R_5 = \text{рік-вп}\}.$

Концепти із  $CN$  мають природну інтерпретацію: унів — множина університетів Києва, факт — множина факультетів, каф — множина кафедр, студенти — множина студентів, жін-ст — особи жіночої статі, рік — рік випуску.

Ролі із множини  $RN$  мають інтерпретацію:  $R_1 = \{\text{маєу}\}$  — це бінарне відношення, яке включає пари (факультет, університет),  $R_2 = \{\text{маєф}\}$  — пари (університет, факультет),  $R_3 = \{\text{маєкф}\}$  — пари (кафедра, факультет),  $R_4 = \{\text{маєст}\}$  — пари (студенти, кафедра),  $R_5 = \{\text{рік-вп}\}$  — пари (рік, студенти).

Нехай дана інтерпретація  $I = (\Delta, f)$ , де областю інтерпретації  $\Delta$  є така множина:

•  $\Delta = \{\text{унів}^f = \{\text{КНУ, КПП, МОГ, НАУ}\};$

$\{\text{факт}^f = \{\text{ек, мат, кіб, іт, інф, інзм, юр, пмат, фіз, хім, гео, іст, біо, мв}\};$

$\{\text{каф}^f = \{\text{мі, іс, ан, геом, алг, мех, крп, оснпр, зф, яф, оргх, норгх, тк, мсс, ймв, екіб, фізх, прав, ген, істу, екп, макр}\};$

•  $\text{студенти}^f = \{\text{Петров П.П., Петрова П.П., Іванов І.І., Сидоров С.С., Сидоров П.С., Петренко П.П., Семенов С.С., Степанов С.С., Сивий О.О., Опанасов О.О., Стасов С.С., Климів К.К., Жигулін Ж.Ж., Ликов Л.Л., Яровий Я.Я., Старов І.І., Варавва В.В., Семенова С.С., Степанова С.С., Сидорова С.С., Варга А.А., Арапов А.А., Степовий С.С., Асанов С.С., Кримов К.К., Критий Л.Л., Протар О.І., Іванов Є.О.}\};$

•  $R_1^f = R_2^-$  — відношення, обернене до  $R_2$ ;

•  $R_2^f = \{(\text{КНУ, ек}), (\text{КНУ, кіб}), (\text{КНУ, мат}), (\text{КНУ, іт}), (\text{КНУ, юр}), (\text{КНУ, фіз}), (\text{КНУ, хім}), (\text{КНУ, гео}), (\text{КНУ, іст}), (\text{КНУ, біо}), (\text{КНУ, мв}), (\text{КПП, мех}), (\text{КПП, інф}), (\text{КПП, хім}), (\text{КПП, юр}), (\text{КПП, ек}), (\text{КПП, мв}), (\text{КПП, пмат}), (\text{МОГ, інф}), (\text{МОГ, іст}), (\text{МОГ, юр}), (\text{МОГ, гум}), (\text{МОГ, ек}), (\text{НАУ, інф}), (\text{НАУ, ек}), (\text{НАУ, мех}), (\text{НАУ, інзм}), (\text{НАУ, юр})\};$

•  $R_3^f = \{(\text{кіб, іс}), (\text{кіб, мі}), (\text{кіб, тк}), (\text{кіб, мсс}), (\text{мат, алг}), (\text{мат, геом}), (\text{мат, ан}), (\text{мат, ймв}), (\text{фіз, зф}), (\text{фіз, яф}), (\text{фіз, рф}), (\text{хім, оргх}), (\text{хім, норгх}), (\text{хім, фізх}), (\text{юр, крп}), (\text{юр, оснпр}), (\text{біо, зб}), (\text{біо, ген}), (\text{біо, бот}), (\text{іст, істу}), (\text{іст, іссв}), (\text{гео, геоу}), (\text{ек, екіб}), (\text{ек, екп}), (\text{ек, макр})\};$

•  $R_4^f = \{(\text{Іванов І.І., тк}), (\text{Протар О.І., тк}), (\text{Петров П.П., мі}), (\text{Семенова С.С., мсс}), (\text{Семенов С.С., оргх}), (\text{Яровий Я.Я., алг}), (\text{Арапов О.О., ан}), (\text{Климів К.К.,}$

оснпр), (Петров С.С., істу), (Петренко П.П., мсс), (Варга В.В., оргх), (Яровий Я.Я., алг), (Варава В.В., алг), (Протар О.І., тк), (Петрова П.П., мі), (Степовий С.С., мех), (Сивий О.О., оргх), (Старов І.І., алг), (Кримов К.К., яф), (Опанасов О.О., ймв), (Іванов І.І., екіб), (Жигулін Ж.Ж., ймв)};

•  $R_5^f = \{(1972, \text{Іванов І.І.}), (1972, \text{Сидоров С.С.}), (1973, \text{Степанов С.С.}), (1971, \text{Петрова П.П.}), (1973, \text{Климів К.К.}), (1971, \text{Петренко П.П.}), (1980, \text{Протар О.І.}), (2018, \text{Арапов О.О.}), (2016, \text{Яровий Я.Я.}), (2016, \text{Старов І.І.}), (2015, \text{Сивий О.О.}), (2015, \text{Степовий С.С.}), (2015, \text{Кримов К.К.}), (2011, \text{Опанасов О.О.}), (2010, \text{Старов О.О.})\}$ .

Визначимо таку термінологію в логіці *ALC*:

студентка  $\equiv$  студенти  $\cap$  жін-ст

студент  $\equiv$  студенти  $\cap \neg$ жін-ст

факт-прир  $\equiv$  мат:факт  $\cup$  кіб:факт  $\cup$  фіз:факт  $\cup$  хім:факт

каф-прир  $\equiv$  мі:каф  $\cup$  іс:каф  $\cup$  ан:каф  $\cup$  мех:каф  $\cup$  тк:каф

рік-вп  $\equiv \{1972, 1973, 1974, \dots, 2017\}$

факт-гум  $\equiv$  факт  $\cap \neg$  факт-прир

Чи виконується наведений нижче концепт в даній інтерпретації?

$\exists \text{маєст.каф} \cap \neg \forall \text{маєст.}(\neg \text{каф-прир} \cap \text{каф}) \subseteq \exists \text{маєст.каф-прир}$

Це включення означає «Чи є студенти, які навчаються ні на яких інших кафедрах, крім природничих?». Перепишемо це включення в термінах позначень відношення «маєст» =  $R_4$ :

$\exists R_4.\text{каф} \cap \neg \forall R_4.(\neg \text{каф-прир} \cap \text{каф}) \subseteq \exists R_4.\text{каф-прир}$ .

Отже, потрібно перевірити виконуваність концепта

$C := \exists R_4.\text{каф} \cap \neg \forall R_4.(\neg \text{каф-прир} \cap \text{каф}) \subseteq \exists R_4.\text{каф-прир}$ .

Якщо відповідь 1, то відповідь на початковий запит буде 0, і навпаки.

**Нормалізація концепта C.** Переносимо заперечення до атомарних концептів, що завжди можна виконати завдяки законам де Моргана. У результаті нормалізації дістаємо такий концепт:

$C_0 = \exists R_4.\text{каф} \cap \exists R_4.(\text{каф-прир} \cap \neg \text{каф}) \cap \forall R_4. \neg \text{каф-прир}$ .

Створимо початкову точку  $x$  з умовою (0)  $x:C_0$ . Застосуємо алгоритм SAT до початкового  $ABox A := \{x:C_0\}$ . Оскільки  $C_0$  — кон'юнкція концептів, то точка  $x$  повинна належати всім цим концептам, і тому до  $A$  додаються такі факти: (1)  $x:\exists R_4.\text{каф}$ , (2)  $x:\exists R_4.\text{каф} \cap (\text{каф-прир} \cap \neg \text{каф})$ , (3)  $x:\forall R_4. \neg \text{каф-прир}$ .

За умовою (1) повинна існувати точка  $u$ , така що  $xR_4u$ , в якій виконується концепт  $\{\text{каф}\}$ . Тому в  $A$  додаються такі факти: (4)  $xR_4u$ , (5)  $u:\text{каф}$ .

За умовою (2) повинна існувати точка  $z$ , така що  $xR_4z$ , в якій виконується концепт  $\text{каф-прир} \cap \neg \text{каф}$ . Тому в  $A$  додаються такі факти: (6)  $xR_4z$ , (7)  $z:\text{каф-прир} \cap \neg \text{каф}$ .

За умовою (3) у всіх точках, зв'язаних з точкою  $x$  відношенням  $R_4$  (а це точки  $u$  і  $z$ ), повинен виконуватися концепт  $\neg \text{каф-прир}$ . Тому в  $A$  додаються такі факти: (8)  $u:\neg \text{каф-прир}$ , (9)  $z:\neg \text{каф-прир}$ .

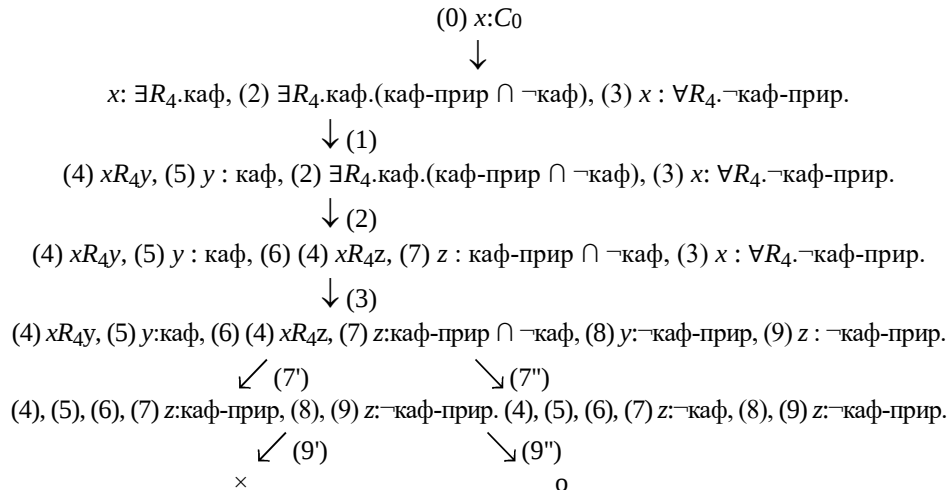
Розглянемо умову (7). Оскільки ця умова є диз'юнкцією, то вона породжує дві гілки, і тому потрібно розглянути два випадки. Перший випадок (7')  $z:\text{каф-прир}$  суперечить умові (9)  $z:\neg \text{каф-прир}$ . Другий випадок (7'')  $z:\neg \text{каф}$  не приводить до суперечності.

Отже, в результаті всі умови зведені до елементарних і знайдена несуперечна модель  $ABox A$ , яка має вигляд:

$$I=(\Delta, f), \text{ де } \Delta=\{x, y, z\}, \text{ каф}^I=\{y\}, \text{ каф-прир}^I=\emptyset, R_4^I=\{(x, y), (x, z)\}.$$

У цій моделі всі факти виконуються, і тому концепт  $C_0$  виконується, а початковий концепт  $C$  хибний. Дійсно, елементом  $x$  в даній моделі за інтерпретації  $I$  виступає студент К.К. Климів, який навчається на кафедрі «Основи права» та студент С.С. Петров, який навчається на кафедрі «Історія України», а ці кафедри не є природничими.

Граф  $(G, L)$ , який відповідає побудові алгоритмом SAT даної моделі, набуває вигляду:



### Висновок

Отже проблема комп'ютерного аналізу природномовного тексту пов'язана з двома областями — комп'ютерною лінгвістикою та математичною логікою. Повна автоматизація процесу такого аналізу на разі виглядає проблематичною, оскільки певні кроки цього процесу не підлягають автоматизації. Зокрема, переклад суджень-фактів з природної мови на мову формальної логіки потребує втручання спеціаліста з даної ПО. А поповнення додатковими гіпотезами в разі неповноти фактів теж вимагає діалогу зі спеціалістом в онтологоподібній системі. Отанням часом на цьому шляху є певний прогрес, який демонструє система GPT.

*S. Kryvyi*

### NATURAL LANGUAGE AND MATHEMATICAL LOGIC IN THE CONTEXT OF THE IDEAS OF V.M. GLUSHKOV

**Sergii Kryvyi**

Taras Shevchenko National University of Kyiv,

*sl.krivoi@gmail.com*

It is considered an approach to the construction of ontology-like systems for the study of natural language texts with the aim of extracting knowledge from them, building a knowledge base for checking this knowledge for contradiction/non-contradiction based on syntactic-semantic analysis of texts using descriptive

logics. The main attention in the work is given to the logical analysis of the acquired knowledge. The natural language text from the point of view of extracting the knowledge contained in it and analyzing this knowledge for compatibility (non-contradiction), their incompleteness and truth is one of the main problems of building artificial intelligence systems. The knowledge accumulated in the natural language text is presented in a formal-logical form, that is, in a predicative form. The need for such a representation is that knowledge can be true, false, incomplete or having no connection with reality. The main task of logical analysis is to check the truth of not individual facts, but to check their consistency (compatibility) as a whole. General features of logical analysis of knowledge obtained from natural language text are considered. The idea is to link the results and tools of linguistic analysis and the tools of formal logical language. This connection allows solving the problem of checking the compatibility of knowledge obtained from the text, generating new knowledge as a consequence of obtained facts, supplementing facts in case of their incompleteness, etc. Currently, one of the main means of implementing such an approach to the analysis is ontologies and ontological knowledge bases. A special case of an ontological system, in which the terminology is acyclic, and a semantical tableau algorithm is focused on this type of ontology are considered. The work is dedicated to the centenary of the birthday of Viktor Mykhailovych Glushkov, whose ideas are being implemented in life.

**Keywords:** ontologies, descriptive logics, logical and syntactical and semantic analysis.

#### ПОСИЛАННЯ

1. Українська радянська енциклопедія. Том 4 (2-е вид.). Стаття «Знання». Київ : Головна редакція УРЕ. 1979. 286 с.
2. Дарчук Н.П. Комп'ютерне анотування українського тексту: результати і перспективи. Київ : Освіта України, 2013. 543 с.
3. Baader F., Calvanese D., McGuinness D.L., Nardi D., Patel-Schneider P.F. The Description logic handbook. Theory, implementation and applications. Cambridge : University Press, 2010. 601 p.
4. Kryvyi S.L., Hoherchak G.I. Analyzing natural language knowledge in uncertainty. 2022 *IEEE International Conference on Advanced Trends in Information Theory (ATIT 2022)*. Kyiv, 2022. Dec. 14-16. P. 268–272.
5. Rademacher A. A proof theory for description logics. Springer Briefs in Computer Science, 2012. 109 p.
6. Schild K. A correspondence theory for terminological logics: preliminary report. *In Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCA'91)*. 1991. P. 466–471.

Отримано 14.06.2023