ПРОГРАМНО-ТЕХНІЧНІ КОМПЛЕКСИ

Г.І. ГОГЕРЧАК, Н.П. ДАРЧУК, С.Л. КРИВИЙ

УДК 004.822

ПРЕДСТАВЛЕННЯ, АНАЛІЗ ТА ВИДОБУВАННЯ ЗНАНЬ З НЕСТРУКТУРОВАНИХ ПРИРОДНОМОВНИХ ТЕКСТІВ

Анотація. Наведено огляд засобів дескриптивних логік для представлення знань з природномовних текстів, класифікацію дескриптивних логік за конструкторами концептів та ролей, а також основні концепції темпоральних дескриптивних логік. Розглянуто підхід до побудови систем аналізу природномовних текстів на основі задач визначення частин мови, пошуку граматичних залежностей та кореферентностей. Наведено приклади використання природномовних баз знань для розв'язання прикладних задач, зокрема для перевірки цілісності тексту, пошуку суперечностей.

Ключові слова: дескриптивні логіки, бази знань, алгоритм семантичного табло, видобування знань, оброблення природної мови, семантичний аналіз.

ВСТУП

Задача оброблення природних мов сьогодні є однією з основних у галузі комп'ютерних наук. Здебільшого це зумовлено прагненням людства до подолання мовних бар'єрів, а також великою кількістю прикладних задач, які тією чи іншою мірою дотичні до формалізації людського мовлення. Такими задачами є покращення сфери послуг, здешевлення певних процесів суспільного значення, які, в свою чергу, стимулюють розвиток методів автоматичного перекладу, реферування та анотування, розпізнавання (перетворення в текстовий формат) мовлення в режимі реального часу, в тому числі природномовних команд, автоматичного пошуку, конструювання відповідей на запитання, виявлення та корекції граматичних помилок, побудови діалогових систем природною мовою, перевірки цілісності тексту, сентимент-аналізу тощо. Сучасна галузь оброблення природних мов нараховує понад три десятки різних задач. Значною мірою вони є лінгвістичними, оскільки пов'язані з визначенням частин мови, лематизацією, токенізацією тексту, побудовою синтаксичних дерев залежностей, пошуком кореферентностей, розпізнаванням іменованих сутностей, відновленням структурної та семантичної неповноти речення, виявленням зв'язків і відношень між мовними одиницями тощо.

Важкість розв'язання таких проблем зумовлена складністю природної мови: багатозначністю, метафоричністю мовного знака, нечіткими множинами мовних одиниць, а отже, неможливістю опису за допомогою чіткого набору детермінованих правил. Найпоширенішим засобом розв'язання вказаних задач є машинне навчання, яке дає змогу в автоматичному режимі здійснювати пошук закономірностей на основі пар вхідних та вихідних даних, що складають корпус навчальної вибірки. Розв'язання задач, які можна чітко сформулювати (наприклад, частиномовне анотування тексту), залежить здебільшого від створення розмічених корпусів необхідного і достатнього обсягу для якісного навчання (наприклад, для якісного морфологічного анотування достатньо сформувати корпус обсягом в один

мільйон слововживань, а для синтаксичного розмічування — десятки і сотні мільйонів слововживань, щоб забезпечити достовірність і репрезентативність результатів аналізу). Проте не всі задачі оброблення природних мов потребують створення таких корпусів — на так званих проміжних етапах опрацювання мовної інформації можна застосовувати засоби математичної логіки тощо, що полегшує процес розв'язання більш складних задач на основі результатів розв'язання менш складних, але глибоко формалізованих і забезпечених корпусами достатнього обсягу.

До таких задач, зокрема, належить задача видобування знань з природномовних текстів, розв'язання якої відкриває шлях до потужного апарату математичних логік для аналізу текстів, написаних українською мовою, та розв'язання інших задач, дотичних до формальної логіки. Їхня особливість полягає у відсутності формальних вимог до формулювання правил представлення природної мови у вигляді певних формальних логічних структур — системи аксіом бази знань. Дійсно, представлення довільного тексту в подібній логічній структурі потребує апарату з достатньою виразною потужністю для представлення знань не тільки типу суб'єкт – дія – об'єкт, а й більш складних відношень, урахування часових, причинно-наслідкових характеристик тощо. Відсутність чіткої формальної постановки цієї задачі унеможливлює побудову корпусу навчальної вибірки для її розв'язання. З іншого боку, побудова такого корпусу у разі успішної формалізації становитиме труднощі через відсутність джерел для автоматичного збирання прикладів розв'язання задачі (як це можна зробити для задач розмічування частин мови, машинного перекладу чи передбачення наступного слова на базі наявних словників, енциклопедичних даних та текстової інформації в мережі Інтернет).

Це зумовлює актуальність досліджень у сфері побудови системи алгоритмів для виявлення та аналізу природномовних знань на базі якісно розв'язаних задач оброблення природної мови, зокрема задач розмічування частин мови, побудови дерева залежностей, пошуку кореферентностей та виявлення іменованих сутностей. Математичною основою такої системи є теорія баз знань та математична логіка.

У статті наведено теоретичні основи математичної логіки та комп'ютерної лінгвістики, а також алгоритми видобування та аналізу природномовних знань на основі якісно розв'язаних задач оброблення природної мови, зокрема задач розмічення частин мови, побудови дерева залежностей, пошуку кореферентностей. Розглянуто засоби представлення, аналізу та видобування знань з неструктурованих природномовних текстів, які становлять теоретичне значення статті.

Запропонований у статті комплекс алгоритмів має і практичне застосування в автоматизації задач перевірки текстів на несуперечність, цілісність тощо.

Побудова повноцінного комплексу для аналізу природномовних знань вимагає розв'язання таких задач:

- визначення засобів формального представлення природномовних знань;
- перетворення неструктурованого природномовного тексту у структурований вигляд;
- побудови алгоритму видобування знань зі структурованого представлення природномовних текстів та їхнього запису в обраній формальній системі;
- застосування алгоритмів математичної логіки та теорії графів до системи одержаних знань.

1. ОСНОВИ ТЕОРІЇ БАЗ ЗНАНЬ

Потреба опису знань за допомогою формальних засобів спричинила появу нового класу логік — дескриптивних (описових). Базові теоретичні поняття різних класів таких логік розглянуто в [1, 2].

1.1. Сімейство мов $\mathcal{AL}$. Нехай $CN = \{A_1, \ldots, A_m\}$ та $RN = \{R_1, \ldots, R_n\}$ — скінченні непорожні множини імен концептів (атомарних концептів) та ролей (атомарних ролей). Тоді синтаксис логіки $\mathcal{AL}$ (attribute language — атрибутивна мова) визначається так.

Означення 1. Множину концептів логіки $\mathcal{AL}$ задають індуктивно:

- символи $\top$ (універсальний концепт) та $\bot$ (порожній концепт) є концептами;
- довільне ім'я концепту $A \in CN$ є концептом;
- якщо A — ім'я концепту, то $\neg A$ (доповнення до A) — концепт;
- якщо C та D — концепти, то $C \sqcap D$ (перетин) — концепт;
- якщо C — концепт, а R — атомарна роль, то $\exists R.\top$ (обмежений квантор існування) та $\forall R.C$ (обмеження на значення) — концепти;
- жодних інших концептів не існує.

Наведене означення коротко можна записати так:

$$\top \mid \bot \mid A \mid \neg A \mid C \sqcap D \mid \exists R.\top \mid \forall R.C.$$

Семантику логіки $\mathcal{AL}$ задають за допомогою поняття інтерпретації.

Означення 2. Інтерпретацією називається пара $I = (\Delta^I, \cdot^I)$, що складається з непорожньої множини Δ^I (область інтерпретації) та функції $\cdot^I$, яка кожному атомарному концепту ставить у відповідність певну підмножину області інтерпретації, а кожній атомарній ролі — підмножину її декартового квадрата.

Таким чином, наведені операції над концептами можна визначити у такий спосіб:

$$\top^I = \Delta^I;$$
$$\bot^I = \varnothing;$$
$$(\neg A)^I = \Delta^I \setminus A^I;$$
$$(C \sqcap D)^I = C^I \cap D^I;$$
$$(\exists R.\top)^I = \{a \in \Delta^I \mid \exists b (a,b) \in R^I\};$$
$$(\forall R.C)^I = \{a \in \Delta^I \mid \forall b (a,b) \in R^I \rightarrow b \in C^I\}.$$

Означення 3. Концепти C та D еквівалентні ($C \equiv D$), якщо за довільної інтерпретації I справджується $C^I = D^I$. Концепт C включається в концепт D ($C \sqsubseteq D$), якщо за довільної інтерпретації I справджується $C^I \subseteq D^I$. Ролі R та S еквівалентні ($R \equiv S$), якщо за довільної інтерпретації I справджується $(a,b) \in R^I \leftrightarrow (a,b) \in S^I$. Роль R включається в роль S ($R \sqsubseteq S$), якщо за довільної інтерпретації I справджується $(a,b) \in R^I \rightarrow (a,b) \in S^I$.

Мови з більш виразною потужністю можна отримувати з мови $\mathcal{AL}$ додаванням нових конструкторів у визначення концептів і ролей (табл. 1).

Таблиця 1. Конструктори концептів та ролей для мов сімейства $\mathcal{AL}$

$\mathcal{AL}$	**Конструктор**	**Інтерпретація**
$\mathcal{U}$	$C \sqcup D$ (об'єднання)	$(C \sqcup D)^I = C^I \cup D^I$
$\mathcal{E}$	$\exists R.C$ (повний квантор існування)	$(\exists R.C)^I = \{a \in \Delta^I \mid \exists b [(a,b) \in R^I \wedge b \in C^I]\}$
$\mathcal{N}$	$\geq nR$, $\leq nR$ (кількісні обмеження)	$(\geq nR)^I = \{a \in \Delta^I \mid \mid \{b \mid (a,b) \in R^I\} \mid \geq n\}$, $(\leq nR)^I = \{a \in \Delta^I \mid \mid \{b \mid (a,b) \in R^I\} \mid \leq n\}$
$\mathcal{C}$	$\neg C$ (доповнення довільного концепту)	$(\neg C)^I = \Delta^I \setminus C^I$
$\mathcal{Q}$	$\geq nR.C$, $\leq nR.C$ (якісні обмеження)	$(\geq nR.C)^I = \{a \in \Delta^I \mid \mid \{b \mid (a,b) \in R^I \wedge b \in C\} \mid \geq n\}$, $(\leq nR.C)^I = \{a \in \Delta^I \mid \mid \{b \mid (a,b) \in R^I \wedge b \in C\} \mid \leq n\}$
$\mathcal{I}$	R^{-1} (обернена роль)	$(R^{-1})^I = \{(b,a) \in \Delta^I \times \Delta^I \mid (a,b) \in R^I\}$

Таким чином, можна визначити низку мов залежно від того, які конструктори будуть до них включені:

$$\mathcal{AL}:=\top\mid\bot\mid A\mid\neg A\mid C\sqcap D\mid\exists R.\top\mid\forall R.C;$$

$$\mathcal{ALU}:=\top\mid\bot\mid A\mid\neg A\mid C\sqcap D\mid C\sqcup D\mid\exists R.\top\mid\forall R.C;$$

$$\mathcal{ALE}:=\top\mid\bot\mid A\mid\neg A\mid C\sqcap D\mid\exists R.C\mid\forall R.C;$$

$$\mathcal{ALEN}:=\top\mid\bot\mid A\mid\neg A\mid C\sqcap D\mid\exists R.C\mid\forall R.C\mid\geq nR\mid\leq nR;$$

$$\mathcal{ALC}:=\top\mid\bot\mid A\mid\neg C\mid C\sqcap D\mid\exists R.\top\mid\forall R.C;$$

$$\mathcal{ALCQ}:=\top\mid\bot\mid A\mid\neg C\mid C\sqcap D\mid\exists R.\top\mid\forall R.C\mid\geq nR.C\mid\leq nR.C.$$

Теорема 1. Мають місце такі співвідношення:

а) $\mathcal{ALC}=\mathcal{ALUE}$;

б) $\mathcal{ALE}\subseteq\mathcal{ALQ}$;

в) $\mathcal{ALN}\subseteq\mathcal{ALQ}$;

г) $\mathcal{ALX}\subseteq\mathcal{ALCQ}$, $X\in\{\mathcal{U},\mathcal{E},\mathcal{N},\mathcal{C},\mathcal{Q}\}^*$.

Доведення. Наведені співвідношення випливають:

а) — з властивостей теоретико-множинних операцій;

б) — з означення якісних обмежень та повного квантора існування (дійсно, $(\exists R.C)=(\geq 1R.C)$);

в) — з означення якісних та кількісних обмежень (дійсно, $(\geq nR)=(\geq nR.\top)$, $(\leq nR)=(\leq nR.\top)$);

г) — з означення доповнення до концепту та якісних обмежень, для чого достатньо довести, що конструктори $\mathcal{U},\mathcal{E},\mathcal{N}$ виражаються через $\mathcal{C}$ та $\mathcal{Q}$, що випливає з а) та в). ■

1.2. Вкладення логіки $\mathcal{ALCQ}$ у логіку предикатів. Оскільки інтерпретація I ставить кожному атомарному концепту A у відповідність певну підмножину області інтерпретації $A^I\subseteq\Delta^I$, такому концепту можна поставити у відповідність одномісний предикат $P_A(x)$ належності індивіда x концепту A. Аналогічно можна визначити двомісний предикат $P_R(x,y)$ існування відношення R між індивідами x та y.

Отже, кожному концепту C у відповідність можна поставити формулу $\phi_C(x)$ таку, що для довільної інтерпретації I множина елементів Δ^I, що задовольняють $\phi_C(x)$, є точно C^I:

$$\phi_\top(x)=T;$$

$$\phi_\bot(x)=F;$$

$$\phi_A(x)=P_A(x);$$

$$\phi_{C\sqcap D}(x)=\phi_C(x)\wedge\phi_D(x);$$

$$\phi_{C\sqcup D}(x)=\phi_C(x)\vee\phi_D(x);$$

$$\phi_{\forall R.C}(x)=\forall y[R(x,y)\to\phi_C(x)];$$

$$\phi_{\exists R.C}(x)=\exists y[R(x,y)\wedge\phi_C(x)];$$

$$\phi_{\geq nR}(x)=\exists\, y_1,\ldots,y_n\left[R(x,y_1)\wedge\ldots\wedge R(x,y_n)\wedge\bigwedge_{i<j}y_i\neq y_j\right];$$

$$\phi_{\leq nR}(x)=\exists\, y_1,\ldots,y_{n+1}\left[R(x,y_1)\wedge\ldots\wedge R(x,y_{n+1})\to\bigvee_{i<j}y_i=y_j\right];$$

$$\phi_{\geq nR.C}(x)=$$

$$=\exists\, y_1,\ldots,y_n\left[R(x,y_1)\wedge\ldots\wedge R(x,y_n)\wedge\phi_C(y_1)\wedge\ldots\wedge\phi_C(y_n)\wedge\bigwedge_{i<j}y_i\neq y_j\right];$$

$$\phi_{\leq nR.C}(x)=$$

$$=\exists\, y_1,\ldots,y_{n+1}\left[R(x,y_1)\wedge\ldots\wedge R(x,y_{n+1})\wedge\phi_C(y_1)\wedge\ldots\wedge\phi_C(y_n)\to\bigvee_{i<j}y_i=y_j\right].$$

Таке вкладення дає змогу застосовувати метод резолюцій для перевірки включення чи еквівалентності концептів. Проте, попри можливість вкладення дескриптивної логіки в логіку предикатів першого порядку, необхідність окремого апарату таких логік зумовлена дещо стислішою формою подання тверджень, що, зокрема, у випадку кількісних та якісних обмежень надає можливість використовувати ефективніші алгоритми виведення, ніж у логіці предикатів.

1.3. Представлення знань. На базі наведених вище синтаксису та семантики можна визначити засоби подання тверджень щодо співвідношення концептів і ролей.

Означення 4. Термінологічною аксіомою називається твердження вигляду $C \equiv D, C \sqsubseteq D$, $R \equiv S$ або $R \sqsubseteq S$, де C, D — концепти, а R, S — ролі. Інтерпретація I задовольняє термінологічну аксіому T, якщо вона інтерпретується в ній як істина. При цьому її називають моделлю аксіоми T.

Термінологія (*TBox*) бази знань є множиною термінологічних аксіом.

Приклад 1. Нехай

$$CN = \{Котячий,\ Кіт,\ Тигр,\ Кішка,\ Кошеня,\ Стать\_жіноча,\ Тварина,\ Вік < 4\}$$

та

$$RN = \{має\_дитину\}$$

є множинами атомарних концептів та атомарних ролей відповідно. Розглянемо простий приклад термінології (*TBox*) на основі підмножини класифікації тварин:

$$Кіт \sqsubseteq Котячий;$$

$$Тигр \sqsubseteq Котячий;$$

$$Кішка \equiv Стать\_жіноча \sqcap Кіт;$$

$$Котячий \sqsubseteq Тварина;$$

$$Кошеня \equiv Кіт \sqcap Вік < 4;$$

$$Кіт \sqsubseteq \forall має\_дитину.Кіт;$$

$$Кіт \sqsubseteq =1\, має\_дитину^{-1}.Кішка;$$

$$Кіт \sqsubseteq =1\, має\_дитину^{-1}.\neg Стать\_жіноча \sqcap Кіт.$$

У вказаній термінології визначено певну ієрархію атомарних концептів, що віддзеркалює належність котів та тигрів до родини котячих, родини котячих до концепту тварин, визначають кішку як кота жіночої статі, а кошеня як кота віком менше чотирьох років. Два останні обмеження вказують, що коти народжують тільки котів і кожен кіт має тільки двох батьків: кішку та кота (чоловічої статі).

Окрім опису відношень між концептами та ролями, у базі знань також повинна бути інформація щодо окремих фактів та окремих об'єктів предметної області (індивідів) у термінах концептів і ролей.

Уведемо додатково нову множину $IN = \{a_1, \ldots, a_m\}$ імен індивідів.

Частина бази знань, що містить інформацію про окремі індивіди, називається *ABox* (assertion box) та складається з двох типів фактів:

- $a : A$ (належність індивіда $a \in IN$ до концепту A);
- aRb (зв'язок двох індивідів $a, b \in IN$ роллю R).

Означення 5. Інтерпретація I називається моделлю термінології $\mathcal{T}$, якщо вона є моделлю всіх її аксіом. Інтерпретація I називається моделлю системи фактів $\mathcal{A}$, якщо для довільних фактів $a : A$ та aRb має місце $a^I \in A^I$ та $(a^I, b^I) \in R^I$. Частина бази знань *ABox* $\mathcal{A}$ називається виконуваною (відносно термінології $\mathcal{T}$), якщо $\mathcal{A}$ має модель, яка є одночасно і моделлю $\mathcal{T}$.

1.4. Основні задачі. Задача наповнення бази знань пов'язана з перевіркою того, чи має новий її концепт зміст у межах вже наявних зв'язків або є навпаки суперечливим, чи включається він в інший концепт, чи є еквівалентним наявному концептові або диз'юнктним щодо нього. З огляду на це ключовими задачами

виведення для термінологій є:

• виконуваність — концепт C виконуваний у термінології $\mathcal{T}$, якщо існує модель I термінології $\mathcal{T}$ така, що C^I непусте;

• поглинання — концепт C поглинається концептом D у термінології $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$), якщо для кожної моделі I термінології $\mathcal{T}$ виконується $C^I \subseteq D^I$;

• еквівалентність — концепти C та D еквівалентні в термінології $\mathcal{T}$ ($C \equiv_{\mathcal{T}} D$), якщо для кожної моделі I термінології виконується $C^I = D^I$;

• диз'юнктність — концепти C та D диз'юнктні в термінології $\mathcal{T}$, якщо для кожної моделі I термінології виконується $C^I \cap D^I = \varnothing$.

Теорема 2. Мають місце такі твердження:

C — виконуваний $\Leftrightarrow C \sqsubseteq_{\mathcal{T}} \top$;

$C \equiv_{\mathcal{T}} D \Leftrightarrow C \sqsubseteq_{\mathcal{T}} D \wedge D \sqsubseteq_{\mathcal{T}} C$;

C, D — диз'юнктивні $\Leftrightarrow C \sqcap D \sqsubseteq_{\mathcal{T}} \bot$;

$C \sqsubseteq_{\mathcal{T}} D \Leftrightarrow C \sqcap \neg D$ — невиконуваний;

$C \equiv_{\mathcal{T}} D \Leftrightarrow C \sqcap \neg D,\ D \sqcap \neg C$ — невиконувані;

C, D — диз'юнктивні $\Leftrightarrow C \sqcap D$ — невиконуваний.

Доведення цих тверджень випливає з властивостей відповідних теоретико-множинних операцій.

Наслідком цих тверджень є те, що для розв'язання наведених вище чотирьох задач достатньо розв'язати задачу перевірки виконуваності концепту або перевірки включення концептів.

Означення 6. Алгоритм U розв'язує проблему виконуваності концептів у термінології $\mathcal{T}$ для дескриптивної логіки L, якщо виконуються такі умови:

• термінальність — для довільних концепту C і термінології $\mathcal{T}$ алгоритм U генерує відповідь $U(C, \mathcal{T})$ за скінченний час;

• коректність — для довільних C і $\mathcal{T}$, якщо C виконується в термінології $\mathcal{T}$, то $U(C, \mathcal{T}) = 1$;

• повнота — для довільних C і $\mathcal{T}$, якщо $U(C, \mathcal{T}) = 1$, то C виконується в термінології $\mathcal{T}$.

1.5. Алгоритм семантичного табло для $\mathcal{ALCQ}$ з термінологіями. Розглянемо алгоритм перевірки виконуваності концепту.

Нехай необхідно перевірити виконуваність концепту C. Сформуємо початковий *ABox* $\mathcal{A}$ бази знань з єдиним твердженням $x : C$.

Як підготовчий крок нормалізуємо концепт C, тобто одержимо еквівалентний концепт, в якому всі заперечення (доповнення) стоять тільки перед атомарними концептами. Для цього можна скористатися тотожностями:

$$\neg(C \sqcup D) \equiv \neg C \sqcap \neg D;\ \neg(C \sqcap D) \equiv \neg C \sqcup \neg D;$$

$$\neg \exists R.C \equiv \forall R.\neg C;\ \neg \forall R.C \equiv \exists R.\neg C;$$

$$\neg \geq nR.C \equiv \leq (n-1)R.C;\ \neg \leq nR.C \equiv \geq (n+1)R.C;\ \neg\neg C \equiv C.$$

Надалі вважатимемо, що концепти нормалізовані.

Означення 7. Індивід x блокує індивіда y, якщо x є предком y та для довільного концепту C має місце твердження $y : C \in \mathcal{A} \Rightarrow x : C \in \mathcal{A}$. Індивід x називається активним, якщо він не блокується жодним іншим індивідом.

На кожному кроці алгоритму застосовуємо до *ABox* одне з правил, наведених у табл. 2. Послідовність виконання правил у цьому алгоритмі довільна за винятком $\geq$-правила, яке виконується тільки, якщо жодне інше застосувати неможливо.

Алгоритм завершує свою роботу, якщо до наступного *ABox* неможливо застосувати жодного з правил, або якщо в ньому міститься суперечність (наявний факт $x : \bot$ або факти $x : A$, $x : \neg A$ одночасно, або досягнуто суперечності на $\leq$-правилі). Початковий концепт виконуваний, якщо під час роботи алгоритму трапляється несуперечливий *ABox*, до якого неможливо застосувати жодного з правил. У решті випадків концепт невиконуваний.

Таблиця 2. Правила алгоритму семантичного табло для $\mathcal{ALCQ}$

Правило	Умови застосування	Дія
$\sqcap$-правило	x — активний; $x:(C\sqcap D)\in\mathcal{A}$; $x:C\notin\mathcal{A}\vee x:D\notin\mathcal{A}$	$\mathcal{A}'=\mathcal{A}\cup\{x:C, x:D\}$
$\sqcup$-правило	x — активний; $x:(C\sqcup D)\in\mathcal{A}$; $x:C\notin\mathcal{A}\wedge x:D\notin\mathcal{A}$	$\mathcal{A}'=\mathcal{A}\cup\{x:C\}$, $\mathcal{A}''=\mathcal{A}\cup\{x:D\}$
$\forall$-правило	x — активний; $x:\forall R.C\in\mathcal{A}$; $\exists\, y: xRy\in\mathcal{A}\wedge y:C\notin\mathcal{A}$	$\mathcal{A}'=\mathcal{A}\cup\{y:C\}$
$\mathcal{T}$-правило	x — активний; $x:E\notin\mathcal{A}$, де $\top\sqsubseteq E\in\mathcal{T}$	$\mathcal{A}'=\mathcal{A}\cup\{x:E\}$
choose-правило	x — активний; $x:\geq nR.C\in\mathcal{A}$; $xRy\in\mathcal{A}$, $y:C\notin\mathcal{A}$, $y:\neg C\notin\mathcal{A}$	$\mathcal{A}'=\mathcal{A}\cup\{y:C\}$, $\mathcal{A}''=A\cup\{y:\neg C\}$
$\geq$-правило	x — активний; $x:\geq nR.C\in\mathcal{A}$; $\neg\exists\, y_1,\dots,y_n[$ $(\forall 1\leq i\leq n[\{xRy_i, y_i:C\}\subseteq\mathcal{A}])\wedge$ $\wedge\forall i<j[y_i\neq y_j\in\mathcal{A}]$ $]$	$y_1,\dots,y_n$ — нащадки x; $\mathcal{A}'=\mathcal{A}\cup\left\{\begin{array}{c} xRy_1,\ y_1:C \\ \dots \\ xRy_n,\ y_n:C \\ y_i\neq y_j,\ i<j \end{array}\right\}$
$\leq$-правило	x — активний; $x:\leq nR.C\in\mathcal{A}$; $\exists y_1,\dots,y_{n+1}[$ $(\forall 1\leq i\leq n+1[\{xRy_i, y_i:C\}\subseteq\mathcal{A}])$ $]$	Якщо $\exists i<j[y_i\neq y_j\notin\mathcal{A}]$, то $\forall i, j$ таких, що $y_i\neq y_j\notin\mathcal{A}$, $\mathcal{A}^{ij}=\mathcal{A}\mid x_j\to x_i$, де $\mathcal{A}\mid x_j\to x_i$ — *ABox*, в якому всі x_j замінено на x_i. Інакше — суперечність.

Приклад 2. Нехай існує база знань:

$$CN=\{Котячий, Кіт, Тигр, Кішка, Кошеня, Стать\_жіноча, Тварина, Вік<4\},$$
$$RN=\{має\_дитину\},$$
$$Кіт\sqsubseteq Котячий,$$
$$Тигр\sqsubseteq Котячий,$$
$$Кішка\equiv Стать\_жіноча\sqcap Кіт,$$
$$Котячий\sqsubseteq Тварина,$$
$$Кошеня\equiv Кіт\sqcap Вік<4,$$
$$Кіт\sqsubseteq\forall\, має\_дитину.Кіт,$$
$$Кіт\sqsubseteq=1\, має\_дитину^{-1}.Кішка;$$
$$Кіт\sqsubseteq=1\, має\_дитину^{-1}.\neg Стать\_жіноча\sqcap Кіт.$$

Розглянемо задачу перевірки виконуваності концепту:

$$Кіт\sqcap\neg\exists\, має\_дитину^{-1}.\top.$$

Зведемо концепт до нормальної форми та отримаємо ініціальний *ABox* вигляду

$$\{x:Кіт\sqcap\forall\, має\_дитину^{-1}.\bot\}.$$

Застосуємо $\sqcap$-правило, після чого одержимо новий *ABox* вигляду

$$\{x:Кіт\sqcap\forall\, має\_дитину^{-1}.\bot, x:Кіт, x:\forall\, має\_дитину^{-1}.\bot\}.$$

Згідно з $\mathcal{T}$-правилом, застосованим до факту $x:Кіт$, отримаємо *ABox*

$$\left\{\begin{array}{c} x:Кіт\sqcap\forall\, має\_дитину^{-1}.\bot, x:Кіт, x:\forall\, має\_дитину^{-1}.\bot, \\ x:=1\ має\_дитину^{-1}.Кішка,\dots \end{array}\right\}.$$

На основі ≥-правила отримаємо факти

$$\textit{має\_дитину}^{-1}(x, y),$$

$$y{:}\,\textit{Кішка}.$$

Згідно з ∀-правилом одержуємо факт

$$y{:}\,\bot,$$

а отже, початковий концепт невиконуваний. Це означає, що не існує котів, що не мають батьків.

Лема 1 (термінальність). Не існує нескінченної послідовності $\mathcal{A}_0, \mathcal{A}_1, \ldots$, в якій кожний *ABox* $\mathcal{A}_{i+1}$ одержано з $\mathcal{A}_i$ за деяким правилом алгоритму семантичного табло.

Доведення. Роботу алгоритму семантичного табло можна представити у вигляді дерева (рис. 1). Вершинами цього дерева є *ABox*, його коренем — ініціальний *ABox* $\mathcal{A}_0 = \{x_0 : C_0\}$. Листками цього дерева є ті *ABox*, до яких неможливо застосувати жодного з правил, а також ті, які містять суперечність. Максимальна кількість ребер, що виходять з кожної вершини цього дерева, визначається ≤-правилом та обмежена квадратом загальної кількості індивідів у відповідному *ABox*. Зазначимо, що повторне застосування ≥-правил, які є єдиним джерелом нових індивідів у цьому алгоритмі, неможливе через суперечність його результату умові його виконання. Таким чином, максимальна кількість індивідів у всіх створених за допомогою алгоритму *ABox* обмежена сумою кількостей індивідів в ініціальному *ABox* та числових характеристик конструкторів типу ≥ в усіх концептах ініціального *ABox*.

Отже, максимальна кількість ребер, що виходять з кожної вершини (її ще називають шириною дерева), обмежена сумою кількостей індивідів в ініціальному *ABox* та числових характеристик конструкторів типу ≥, що визначаються вхідними даними алгоритму.

Побудуємо множину підконцепців концепту C у такий спосіб:

$$Sub(\top) = \{\top\};$$

$$Sub(\bot) = \{\bot\};$$

$$Sub(A) = \{A\},\ \ A \in CN;$$

$$Sub(\neg C) = \{\neg C\} \cup Sub(C);$$

$$Sub(C \sqcap D) = \{C \sqcap D\} \cup Sub(C) \cup Sub(D);$$

$$Sub(C \sqcup D) = \{C \sqcup D\} \cup Sub(C) \cup Sub(D);$$

$$Sub(\forall R.C) = \{\forall R.C\} \cup Sub(C);$$

$$Sub(\geq nR.C) = \{\geq nR.C\} \cup Sub(C);$$

$$Sub(\leq nR.C) = \{\leq nR.C\} \cup Sub(C).$$

Тоді множина концептів кожного індивіда довільного *ABox* під час виконання алгоритму належить множині $Sub(C_0) \cup Sub(E)$, а отже скінченна.

З іншого боку, довжина будь-якого ланцюга в дереві від його кореня до листка не може перевищувати значення 2^n, $n = |\, Sub(C_0) \cup Sub(E)\,|$, а якщо ця умова не виконується, то існуватимуть два індивіди з однаковими концептами і один з них блокуватиме іншого, що зупинить зростання довжини ланцюга.

Отже, оскільки кількість ребер з однієї вершини, довжина ланцюга від кореня до листка, а також кожен *ABox* скінченні, то такого нескінченного ланцюга *ABox* не існує. ■

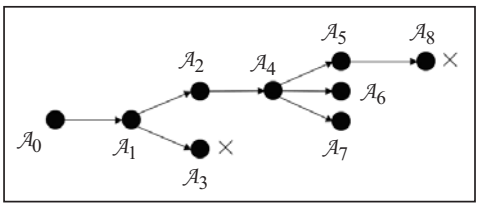

Рис. 1. Схема алгоритму семантичного табло

Лема 2 (коректність). Справедливі такі твердження:

1) концепт C виконуваний відносно $\mathcal{T}$ тоді і тільки тоді, коли *ABox* $\mathcal{A}_0 = \{x_0 : C_0\}$ виконуваний відносно $\mathcal{T}$;

2) нехай $\mathcal{A}'$ одержаний з $\mathcal{A}$ за допомогою одного з правил $\sqcap$, $\forall$, $\mathcal{T}$, $\geq$ алгоритму. Тоді, якщо $\mathcal{A}$ виконуваний відносно $\mathcal{T}$, то $\mathcal{A}'$ теж виконуваний відносно $\mathcal{T}$;

3) нехай $\mathcal{A}'$ та $\mathcal{A}''$ одержані з $\mathcal{A}$ за допомогою одного з правил $\sqcap$, *choose*. Тоді якщо $\mathcal{A}$ виконуваний відносно $\mathcal{T}$, то $\mathcal{A}'$ або $\mathcal{A}''$ теж виконувані відносно $\mathcal{T}$;

4) нехай $\mathcal{A}^1, \mathcal{A}^2, ..., \mathcal{A}^k$ одержані з $\mathcal{A}$ за допомогою правила $\leq$ алгоритму. Тоді якщо $\mathcal{A}$ виконуваний відносно $\mathcal{T}$, то принаймні один з $\mathcal{A}^1, \mathcal{A}^2, ..., \mathcal{A}^k$ теж виконуваний відносно $\mathcal{T}$.

Доведення. Твердження 1 очевидне. Для доведення тверджень 2 та 3 розглянемо кожне правило окремо:

• нехай $\mathcal{A}'$ одержаний з $\mathcal{A}$ за $\sqcap$-правилом. Тоді якщо $\mathcal{A}$ виконуваний, то існує його модель I. За умовою виконання $\sqcap$-правила $x^I \in (C \sqcap D)^I$. Отже, справедливо $x^I \in C^I \cap D^I \Rightarrow x^I \in C^I \wedge x^I \in D^I$. Таким чином, I — модель $\mathcal{A}'$;

• нехай $\mathcal{A}'$ одержаний з $\mathcal{A}$ за $\forall$-правилом. Тоді якщо $\mathcal{A}$ виконуваний, то існує його модель I. За умовою виконання $\forall$-правила $x^I \in (\forall R.C)^I$, $x^I R^I y^I$. Отже, для довільного елемента $d \in \Delta^I$ такого, що $x^I R^I d$, справедливо $d \in C^I$. Але це справедливо і для $d = y^I$, тому $y^I \in C^I$. Таким чином, I — модель $\mathcal{A}'$;

• нехай $\mathcal{A}'$ одержаний з $\mathcal{A}$ за $\mathcal{T}$-правилом. Тоді, якщо $\mathcal{A}$ виконуваний, існує його модель I. За умовою виконання $\mathcal{T}$-правила $\Delta^I \subseteq E^I \Rightarrow x^I \in E^I$. Таким чином, I — модель $\mathcal{A}'$;

• нехай $\mathcal{A}'$ одержаний з $\mathcal{A}$ за $\geq$-правилом. Тоді, якщо $\mathcal{A}$ виконуваний, існує його модель I. За умовою виконання $\geq$-правила $x^I \in (\geq nR.C)^I$. Отже, $x^I \in \{a \in \Delta^I : | \{b | (a,b) \in R^I \wedge b \in C\} | \geq n\}$. Звідси випливає, що існують попарно різні елементи $d_1, ..., d_n \in C^I$ такі, що $x^I R^I d_i$. З іншого боку, для $\mathcal{A}'$ наявні попарно різні $y_1^I, ..., y_n^I$ такі, що $x^I R^I y_1^I, ..., x^I R^I y_n^I$, $y_1^I \in C^I, ..., y_n^I \in C^I$. Таким чином, I — модель $\mathcal{A}'$;

• нехай $\mathcal{A}'$, $\mathcal{A}''$ одержані з $\mathcal{A}$ за $\sqcup$-правилом. Тоді, якщо $\mathcal{A}$ виконуваний, існує його модель I. За умовою виконання $\sqcup$-правила $x^I \in (C \sqcup D)^I$. Отже, $x^I \in C^I \cup D^I \Rightarrow x^I \in C^I \vee x^I \in D^I$. Таким чином, I — модель або $\mathcal{A}'$, або $\mathcal{A}''$;

• нехай $\mathcal{A}'$, $\mathcal{A}''$ одержані з $\mathcal{A}$ за *choose*-правилом. Тоді, якщо $\mathcal{A}$ виконуваний, існує його модель I. Отже, $x^I \in C^I \vee x^I \in \neg C^I \Rightarrow x^I \in C^I \vee x^I \in (\neg C)^I$. Таким чином, I — модель або $\mathcal{A}'$, або $\mathcal{A}''$;

• нехай $\mathcal{A}^1, \mathcal{A}^2, ..., \mathcal{A}^k$ одержані з $\mathcal{A}$ за $\leq$-правилом. Тоді, якщо $\mathcal{A}$ виконуваний, існує його модель I. За умовою виконання $\leq$-правила $x^I \in (\leq nR.C)^I$. Отже, $x^I \in \{a \in \Delta^I : | \{b | (a,b) \in R^I \wedge b \in C\} | \} \leq n\}$. Оскільки за умовою правила існує $n+1$ індивід, що задовольняє таким правилам, то для деяких індивідів d_1, d_2 буде справедливим $d_1^I = d_2^I$. Нехай $\mathcal{A}^j$ утворений заміною d_2 на d_1. Оскільки їхні інтерпретації збігаються, I — модель $\mathcal{A}^j$. ■

Лема 3 (повнота). Несуперечний *ABox*, до якого не можна застосувати жодного з правил алгоритму, виконуваний.

Доведення. Нехай $\mathcal{A}$ — несуперечний *ABox*, отриманий з $\mathcal{A}_0$, і до нього не можна застосувати жодного з правил алгоритму.

Побудуємо інтерпретацію I для $\mathcal{A}$ у такий спосіб:

$$\Delta^I := \{x | x \in \mathcal{A}\};$$

$$A^I := \{x \in \Delta^I \mid x:C \in \mathcal{A}\} \cup \{x \mid z \text{ блокує } x \wedge z:C \in \mathcal{A} \wedge z \text{ — активний}\};$$

$$R^I := \{(x, y) \mid xRy \in \mathcal{A} \wedge x \text{ — активний}\} \cup$$

$$\cup \{(x, y) \mid z \text{ блокує } x \wedge zRy \in \mathcal{A} \wedge z \text{ — активний}\}.$$

До $\mathcal{A}$ не можна застосувати $\mathcal{T}$-правило, а отже

$$x^I \in \Delta^I \to (x^I \in E^I \vee \exists z[z \text{ блокує } x \wedge z \text{ — активний}]).$$

З означення блокування $z^I \in E^I$, тому за побудовою $x^I \in E^I$. Звідси I є моделлю $\mathcal{T}$.

Розглянемо факт $xRy \in \mathcal{A}$. За побудовою I маємо $x^I R^I y^I$.

Для фактів $x:C \in \mathcal{A}$ скористаємось методом математичної індукції.

Необхідно довести таке твердження:

$$\forall x \in \Delta^I [x:C \in \mathcal{A} \to x \in C^I].$$

З урахуванням нормалізованості концепт C побудовано з концептів $\top$, $\bot$, A, $\neg A$ (де A — атомарний) за допомогою операторів $\sqcap, \sqcup, \forall, \geq, \leq$.

Також зауважимо, що для будь-якого неактивного індивіда x існує активний індивід y, що його блокує.

База індукції:

$$\neg\exists x \in \Delta^I [x:\bot \in \mathcal{A}] \Rightarrow \forall x \in \Delta^I [x:\bot \in \mathcal{A} \to x \in \varnothing];$$

$$\forall x \in \Delta^I [x \in \Delta^I] \Rightarrow \forall x \in \Delta^I [x:\top \in \mathcal{A} \to x \in \Delta^I];$$

$$\{x \in \Delta^I \mid x:A \in \mathcal{A}\} \subseteq A^I \Rightarrow \forall x \in \Delta^I [x:A \in \mathcal{A} \to x \in A^I];$$

$$x:\neg A \in \mathcal{A} \wedge x \notin (\neg A)^I \Rightarrow x:\neg A \in \mathcal{A} \wedge x \in A^I \Rightarrow x \in \neg A^I \wedge x \in A^I \Rightarrow \mathcal{A}' \text{ — суперечна.}$$

Крок індукції.

Випадок 1. Нехай $x:D \sqcap E \in \mathcal{A}$. Оскільки застосувати $\sqcap$-правило до $\mathcal{A}$ неможливо, справджується

$$\forall x[x:D \sqcap E \in \mathcal{A} \to (x:D \in \mathcal{A} \wedge x:E \in \mathcal{A}) \vee \exists z[z \text{ блокує } x \wedge z \text{ — активний}]].$$

За побудовою з $x:D \in \mathcal{A} \wedge x:E \in \mathcal{A}$ випливає $x \in D^I \wedge x \in E^I$.

У протилежному випадку за означенням блокування $z:D \sqcap E \in \mathcal{A}$, а з активності z випливає, що $z:D \in \mathcal{A} \wedge z:E \in \mathcal{A}$. За побудовою інтерпретації одержуємо $x \in D^I \wedge x \in E^I$. Отже, $x \in D^I \cap E^I = (D \sqcap E)^I$.

Випадок 2. Нехай $x:D \sqcup E \in \mathcal{A}$. Оскільки застосувати $\sqcup$-правило до $\mathcal{A}$ неможливо, справджується

$$\forall x[x:D \sqcup E \in \mathcal{A} \to (x:D \in \mathcal{A} \vee x:E \in \mathcal{A}) \vee \exists z[z \text{ блокує } x \wedge z \text{ — активний}]].$$

За побудовою з $x:D \in \mathcal{A} \vee x:E \in \mathcal{A}$ випливає $x \in D^I \vee x \in E^I$.

У протилежному випадку за означенням блокування $z:D \sqcup E \in \mathcal{A}$, а з активності z випливає, що $z:D \in \mathcal{A} \vee z:E \in \mathcal{A}$. За побудовою інтерпретації одержуємо $x \in D^I \vee x \in E^I$. Отже, $x \in D^I \cup E^I = (D \sqcup E)^I$.

Випадок 3. Нехай $x:\forall R.D \in \mathcal{A}$. Оскільки застосувати $\forall$-правило до $\mathcal{A}$ неможливо, маємо

$$\forall y[xRy \in \mathcal{A} \to y:D \in \mathcal{A} \vee \exists z[z \text{ блокує } x \wedge z \text{ — активний}]]. \qquad (1)$$

Нехай x — активний. Розглянемо довільний $y \in \Delta^I$ такий, що $xR^I y$. Тоді за побудовою $xRy \in \mathcal{A}$ і з урахуванням (1) маємо $y:D \in \mathcal{A}$. Звідси випливає, що $y \in D^I$.

Нехай z — активний і блокує x. Виберемо довільний $y \in \Delta^I$ такий, що $xR^I y$.

Тоді за побудовою $zRy \in \mathcal{A}$ і з урахуванням (1) для активного z справедливо $y:D \in \mathcal{A}$. Звідси отримуємо $y \in D^I$.

Таким чином, $x \in (\forall nR.D)^I$.

Випадок 4. Нехай $x:\geq nR.D \in \mathcal{A}$. Оскільки застосувати ≥-правило до $\mathcal{A}$ неможливо, справджується

$$\exists y_1,\ldots, y_n[(\forall 1 \leq i \leq n[\{xRy_i, y_i:D\} \subseteq A]) \wedge \forall i< j[y_i \neq y_j \in \mathcal{A}]] \vee$$
$$\vee \exists z[z \text{ блокує } x \wedge z \text{ — активний}]. \quad (2)$$

Нехай x — активний. Тоді справедливо $xR^I y_i, y_i \in D^I$ за побудовою інтерпретації.

Нехай z — активний і блокує x. Тоді з урахуванням (2) для активного z справедливо $zRy_i \in \mathcal{A}$, $y_i:D \in \mathcal{A}$. Звідси випливає, що $xR^I y_i, y_i \in D^I$ за побудовою інтерпретації. Отже, існує n різних елементів з Δ^I, для яких $xR^I y_i$ та $y_i \in D^I$. Таким чином, $x \in (\geq nR.D)^I$.

Випадок 5. Нехай $x:\leq mR.D \in \mathcal{A}$. Оскільки застосувати ≤-правило до $\mathcal{A}$ неможливо, справджується

$$\neg\exists y_1,\ldots, y_{n+1}[\forall i[\{xRy_i, y_i:D\} \subseteq \mathcal{A}]] \vee$$
$$\vee \exists z[z \text{ блокує } x \wedge z \text{ — активний}].$$

Перетворимо цей вираз відповідно до правил де-Моргана та двоїстості:

$$\forall y_1,\ldots, y_n[\exists i[xRy_i \notin \mathcal{A} \vee y_i:D \notin \mathcal{A}]] \vee$$
$$\vee \exists z[z \text{ блокує } x \wedge z \text{ — активний}]. \quad (3)$$

Припустимо, що існує $n+1$ таких $y_i \in D$, що $xR^I y_i \in \mathcal{A}$.

Нехай x — активний. Тоді для довільного i справедливо $y_i:D$ та $xRy_i \in \mathcal{A}$, що суперечить (3).

Нехай z — активний і блокує x. Тоді для довільного i справедливо $z:D \in \mathcal{A} \vee x:D \in \mathcal{A}$ та $zRy_i \in \mathcal{A}$. За означенням блокування $x:D \in \mathcal{A} \to z:D \in \mathcal{A}$, отже, має місце $z:D \in \mathcal{A}$ та $zRy_i \in \mathcal{A}$, що суперечить (3) для активного z.

Отже, існує не більше n різних елементів з Δ^I, для яких $xR^I y_i$ та $y_i \in D^I$. Таким чином, $x \in (\leq nR.D)^I$. ■

Теорема 3 (розв'язуваність $\mathcal{ALCQ}$). Алгоритм семантичного табло розв'язує проблему виконуваності концептів логіки $\mathcal{ALCQ}$.

Доведення. Термінальність. З леми 1 випливає, що дерево пошуку не має нескінченних ланцюгів, а оскільки ступінь його розгалуження обмежений, дерево пошуку скінченне. Отже, для довільних вхідних даних алгоритм семантичного табло поверне відповідь за скінченний час.

Коректність. Якщо $\mathcal{A}_0$ виконуваний, то за лемою 2 хоча б один з кінцевих *ABox* $\mathcal{A}$ виконуваний. Він не може бути суперечним, а тому є несуперечним *ABox*, до якого не можна застосувати жодного з правил. За побудовою алгоритму в цьому випадку буде одержано 1.

Повнота. Нехай на виході алгоритму одержано 1. Тоді серед його кінцевих *ABox* існує такий $\mathcal{A}$, що є несуперечним і до нього не можна застосувати жодного з правил. За лемою 3 $\mathcal{A}$ виконуваний. Вочевидь, $\mathcal{A}_0 \subseteq \mathcal{A}$, оскільки алгоритм семантичного табло лише додає факти, але не вилучає їх. Таким чином, $\mathcal{A}_0$ теж виконуваний. ■

1.6. Темпоральні дескриптивні логіки. Дескриптивні логіки не мають достатньо виразної потужності для представлення знань про поведінку індивідів у часі. Дійсно, якщо розглянути твердження «Я виконував домашнє завдання вчора, проте сьогодні ні», то за класичною дескриптивною логікою воно містить два суперечних

факти: *виконував* (x, y), $\neg$ *виконував* (x, y), хоча семантично ці факти мали місце в різний часовий проміжок, а тому їх не можна вважати суперечними.

У такому разі допоміжною математичною моделлю може бути логіка лінійного часу *LTL*, для якої передбачається наявність темпоральних операторів: O — у наступний момент, $\Diamond$ — колись, $\Box$ — у будь-який момент у майбутньому та $\mathcal{U}$ — поки. Ці оператори дають змогу розширити класичні дескриптивні логіки часовим виміром.

Розглянута в [2] концепція гібридної логіки $LTL_{\mathcal{ALC}}$ додає до розглянутих раніше конструкторів концептів ще два темпоральні (табл. 3). До того ж інтерпретація фактів у цій логіці отримує додатковий вимір — часовий.

За допомогою вказаних конструкцій твердження «Для того щоб стати успішним, потрібно вчитися» можна навести у вигляді аксіоми

$$\neg \textit{успішний} \sqcap \Diamond \textit{успішний} \sqsubseteq \Diamond ((\exists \textit{вчитися}) \,\mathcal{U}\, \textit{успішний}).$$

Алгоритм семантичного табло для темпоральних дескриптивних логік представлено в [3].

Таблиця 3. Темпоральні конструктори концептів для мов сімейства $\mathcal{AL}$

$\mathcal{AL}$	**Конструктор**	**Інтерпретація**
O	$\mathrm{O}C$ (у наступний момент)	$(\mathrm{O}C)^I = \{(n,x) \mid (n+1,x) \in C^I\}$
$\mathcal{U}ntil$	$C\mathcal{U}D$ (поки)	$(C\mathcal{U}D)^I = \{(t,x) \mid \exists u \geq t[(u,x) \in D^I \wedge$ $\wedge \forall v[t \leq v < u \rightarrow (v,x) \in C^I]]\}$

2. ВИДОБУВАННЯ ЗНАНЬ З ПРИРОДНОМОВНОГО ТЕКСТУ

Задача виявлення відкритої інформації (open information extraction) полягає у представленні природномовного тексту в формалізованому вигляді: зазвичай у вигляді бінарних відношень, а також відношень більших розмірностей, у термінах базової математичної логіки тощо. Якісне розв'язання цієї задачі свідчило б про наявність автоматизованих методів наповнення бази знань з природномовних даних, зміст яких і складається з атомарних концептів та ролей — відношень між ними.

Складність цієї задачі, окрім спільної для всіх задач оброблення природної мови проблеми неоднозначності мовлення людини, полягає у труднощах представлення довільного неструктурованого тексту у формалізованому вигляді. Значні результати наразі досягнуто в окремих звуженнях постановки цієї задачі. Так, засобами машинного навчання досягаються непогані результати [5] для задачі виявлення відношень, яка звужує розгляд до видобування з тексту трійок суб'єкт–дія–об'єкт для обмеженого переліку доступних дій, а також для задачі видобування відкритої інформації [6].

Проте в повному обсязі ця задача не має чітко сформульованих та загальноприйнятих стандартів результату, тобто не визначено, які саме відношення потрібно одержати та яким чином вони повинні оформлюватися. Не сформовано також стандарту оцінювання моделей та корпусів прийнятного обсягу для якісної побудови моделей машинного навчання, як це прийнято для багатьох задач оброблення природномовних текстів.

Дослідження проблеми видобування знань з природномовних текстів для подальшого наповнення ними онтологоподібних систем проводяться наразі як зарубіжними [7, 8], так і вітчизняними [9–13] вченими. Зокрема, серед наявних аналогів слід зазначити системи FRED [7] та SHELDON [8], що здійснюють побудову OWL-онтологій на основі природномовних текстів і є у відкритому доступі. Попри загальне призначення цих систем, їхніми основними недоліками є відсутність підтримки часових зв'язків та низька якість роботи з неангломовними текстами, оскільки будь-який такий текст попередньо перекладають

англійською за допомогою автоматичних засобів, через що часто втрачається корисне змістовне навантаження.

Розглянемо підхід до розв'язання цієї задачі за допомогою проміжних структурно-лінгвістичних представлень природномовного тексту, базові засади якого було сформульовано в [9]. Основною метою цього підходу є побудова цілісної конвеєрної системи видобування знань з текстів, яка дає змогу легко здійснювати заміну окремих її компонентів з метою інкорпорування допоміжних моделей машинного навчання з кращими результатами.

2.1. Засоби побудови проміжних лінгвістичних структур. Неструктурований, звичайний текст є складним для безпосереднього алгоритмічного аналізу через багатогранність та неоднозначність людського мовлення. Саме тому для задач оброблення природної мови часто послуговуються додатковими, більш інформативними поданнями тексту у вигляді структур даних, які обробляють за допомогою алгоритмів. Таке представлення тексту називатимемо структурним.

З огляду на потужний апарат алгоритмів над структурами на кшталт дерев та графів у переважній більшості представлень тексту в різних аспектах використовують або деревовидну, або графову структуру. Розглянемо деякі поширені структурні представлення лінгвістичної інформації.

Найменшою змістовою одиницею мови в сучасних засобах її оброблення є токен — послідовність символів речення, що являє собою певний зміст. Зазвичай речення поділяють на токени за розділювачами на кшталт пунктуаційних знаків та пробілів, проте існують й винятки. Наприклад, вираз «і.е.» англійською означає «іншими словами» та є одним єдиним токеном.

Кожен токен у результаті лексичного аналізу може мати одну чи кілька характеристик, зокрема, частину мови, рід, відмінок тощо. За синтаксичним розбором мови токени пов'язують одне з одним за допомогою синтаксичних граматичних зв'язків.

Одне з найбільш змістовних представлень речення — це дерево (в деяких випадках — граф) залежностей (рис. 2).

За таким структурним представленням текстова інформація має вигляд дерева, вершинами якого є токени (найменші синтаксично значущі одиниці речення), коренем є присудок (зазвичай дієслово, в окремих випадках — іменник, прислівник чи прикметник), а ребра позначають залежність одних токенів від інших.

Приклад 3. У наведеному на рис. 2 прикладі токен «повідомлення» пов'язаний з токеном «записують», який є коренем дерева, залежністю obj (об'єкт), а токен «замінюючи» — залежністю advcl (прислівниковий зворот).

Цей універсальний формат дерев залежностей для різних природних мов запропоновано у [14]. Для порівняння якості (відповідності еталону) різних підходів до побудови такого дерева використовують дві метрики: UAS та LAS.

Означення 8. Оцінка непозначеного приєднання (UAS) — це відсоткове відношення токенів, що мають коректно визначеного предка. Оцінка позначеного приєднання (LAS) — це відсоткове відношення токенів, що мають коректно визначеного предка та коректну позначку.

Наразі найкращі значення наведених вище метрик демонструє модель Label Attention Layer + HPSG + XLNet, запропонована в [15]. Ця модель також базується на нейромережевому підході та досягає UAS 97.33 % і LAS 96.29 % для англійської мови. Проте на останніх наукових конференціях основну увагу приділено побудові єдиних моделей синтаксичного розбору для великої кількості мов. Так, модель HIT-SCIR [16] дає змогу досягти LAS у 92 %, 88 % та 87 % для російської, української та англійської мов відповідно.

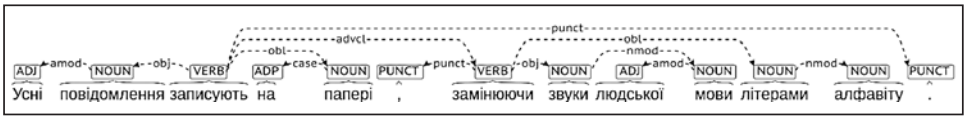

Рис. 2. Дерево залежностей речення (формат Universal Dependencies)

Існують й інші формати представлення дерев синтаксичних залежностей, зокрема наведений на рис. 3 з [17].

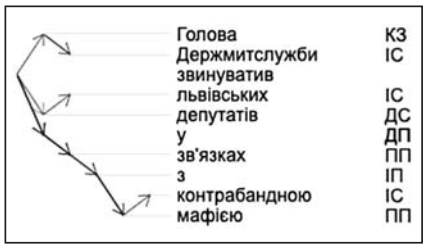

Рис. 3. Дерево залежностей речення (формат mova.info)

Наявність кількох моделей, які будують подібні представлення речення у вигляді дерев, а також різних форматів представлень одного речення дає змогу використовувати для дерев алгоритми виявлення потенційних помилок у представленнях та їхньої корекції.

Зазначимо, що дерева залежностей демонструють графічно лише зв'язки в межах одного речення, а співвідношення між сутностями в різних реченнях залишаються невідомими. Для їхнього представлення використовують іншу структуру — групи кореферентностей (рис. 4).

Множина кореферентних слів та словосполучень зазвичай має вигляд лісу — множини дерев, кожне з яких позначає множину кореферентних вузлів. Дуга кореферентності зазвичай спрямовується до найбільш конкретного позначення об'єкта реального світу.

Порівняння підходів до розв'язання цієї задачі здійснюють за допомогою середнього арифметичного трьох метрик: MUC, B^3 та $CEAF_{\varphi 4}$.

В оцінці MUC [18] враховується найменша кількість дуг між сутностями, які потрібно додати або вилучити, щоб з одержаного результату отримати еталонний:

$$MUC = \frac{\sum_{S_i \in T} |S_i| - |p(S_i)|}{\sum_{S_i \in T} |S_i| - 1},$$

де T — множина еталонних груп кореферентних сутностей, $p(S)$ — кількість груп кореферентних сутностей у результаті, що відповідають еталонній групі S.

Метрика B^3 [19] базується на порівнянні розмірів відповідних груп кореферентності:

$$B^3_{\text{prec}} = \frac{1}{|M|} \sum_{m \in M} \frac{G_m \cap P_m}{P_m},$$

$$B^3_{\text{rec}} = \frac{1}{|M|} \sum_{m \in M} \frac{G_m \cap P_m}{G_m},$$

$$B^3 = \frac{2B^3_{\text{prec}} B^3_{\text{rec}}}{B^3_{\text{prec}} + B^3_{\text{rec}}},$$

де M — множина сутностей, G_m — еталонна група, до якої належить сутність m, P_m — група результату, до якої належить сутність m.

Метрику $CEAF_{\varphi 4}$ обчислюють спеціалізованим алгоритмом порівняння, запропонованим у [21].

Наразі найкращі показники для цієї задачі демонструють модифікації моделі BERT [22], яка базується на машинному навчанні (модель розроблена командою Google AI Language). Модель BERT пропонує спільний підхід до подання природномовної інформації для сукупності задач оброблення текстів та запроваджує врахування ліво- та правобічного контексту слова, на відміну від оброблення тексту

Рис. 4. Ліс кореферентностей для речення українською мовою

зліва направо чи справа наліво у попередніх ефективних моделях. Кращий результат середнього $F1 = 80.2$ досягає модель машинного навчання, запропонована в [20].

2.2. Видобування знань зі структурного подання природномовного тексту. Деякі підходи до аналізу природної мови для видобування знань представлені в [9, 12, 13]. Розглянемо основні правила видобування фактів та аксіом на основі дерева універсальних залежностей.

Дерево універсальних залежностей (див. рис. 2) надає важливі вхідні дані для алгоритмів видобування знань за форматом своєї побудови. Так, оскільки наведені в ньому залежності позначають не суто синтаксичні, а й семантико-синтаксичні зв'язки і відношення, деякі правила видобування знань на основі цієї структури є тривіальними.

Проте деревовидна структура подання інформації в дереві універсальних залежностей дещо погіршує засоби оброблення у випадках наявності сурядних токенів. Аби перетворити деревовидну структуру у графову, яка зручніша для оброблення, використовують алгоритм розширення залежностей. Перетворення базового дерева залежностей у розширений граф залежностей потребує, зокрема, розв'язання таких проблем:

- відновлення слів, яких у тексті немає, але їхня наявність мається на увазі, шляхом створення фіктивних токенів;
- поширення зв'язків (об'єктів, суб'єктів, означень) через кон'юнкцію;
- поширення суб'єктів на підпорядковані дієслова складного предиката;
- оброблення підрядного речення, що уточнює певний об'єкт, як дії, виконаної цим об'єктом (може призводити до утворення циклів);
- додавання допоміжного слова в назву залежності.

Дерево та граф залежностей наведені на рис. 5 і 6 відповідно.

Такий граф може містити множину кореневих токенів (на рис. 6 їх два, оскільки корінь початкового дерева залежностей «ловлять» має сурядний з ним токен «харчуються»). Інші токени можуть мати більш ніж одного предка (наприклад, токен «щурів» є нащадком також і токена «ловлять» в наслідок поширення цього зв'язку через сурядний йому токен «мишей»).

На основі розширеного графу залежностей здійснюють аналіз щодо наявності аксіом та фактів. Розглянемо декілька базових правил видобування знань такого типу.

Якщо коренем дерева залежностей є дієслово, для якого наявні дуги obj та subj, можна згенерувати факт для *ABox* вигляду $x_{\text{subj}} R_{\text{root}} x_{\text{obj}}$, де x_{subj} — ім'я індивіда, що відповідає токену з залежністю subj, x_{obj} — ім'я індивіда, що відповідає токену з залежністю obj, R_{root} — ім'я ролі, що відповідає кореню речення. Також кожен індивід пов'язується з відповідним йому концептом за допомогою факту $x_{\text{word}} : A_{\text{word}}$.

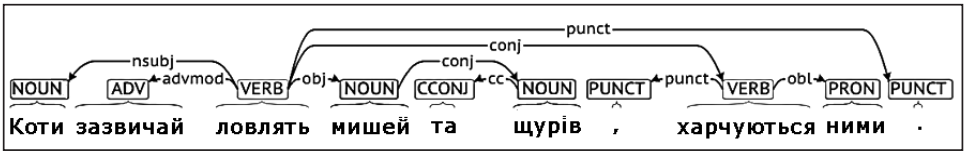

Рис. 5. Базове дерево залежностей

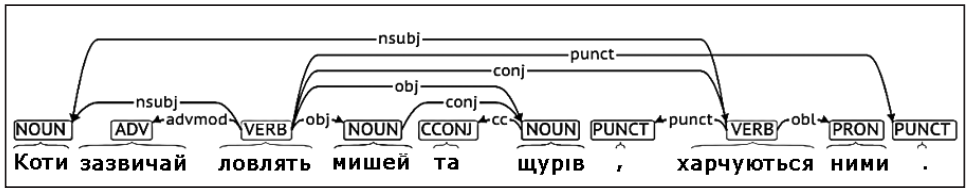

Рис. 6. Розширення дерева залежностей

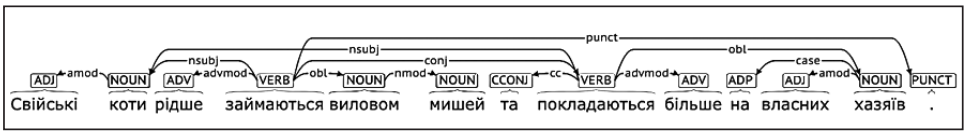

Рис. 7. Розширений граф залежностей

Це правило ускладнюється, якщо об'єктів чи суб'єктів більш ніж один. Нехай S — множина токенів-суб'єктів, O — множина токенів-об'єктів. Тоді $ABox$ бази знань поповнюється фактами з множини $\{x_s R_{\text{root}} x_o \mid \forall s \in S, \forall o \in O\}$.

Аналогічно джерелом фактів є дуги obl, які позначають додаток. У таких випадках більш виразним буде збереження пов'язаних цим типом залежностей токенів як додаткової характеристики ролі, що потребує підтримки багатомісних ролей відповідними дескриптивними логіками.

Отримати факти з розширеного графу залежностей складніше для модифікаторів: amod (прикметниковий), nmod (іменниковий), compound (частина цілого), flat (частини імен, дат тощо). Джерелом аксіом включення є речення з іменниками та прикметниками, які являють собою корені дерев залежностей.

Так, для речення, наведеного на рис. 7, можна сформувати таку базу знань:

$$CN = \{кіт,\ вилов,\ вилов\_мишей,\ хазяїн,\ свійський,\ власний\};$$

$$RN = \{займатися,\ покладатися\_на\};$$

$$IN = \{x_{свійський\_кіт}, x_{вилов\_мишей}, x_{власний\_хазяїн}\};$$

$$\mathcal{T} = \{вилов\_\ мишей \sqsubseteq вилов\};$$

$$\mathcal{A} = \{x_{свійський\_кіт}\ займатися\ x_{вилов\_мишей},$$

$$x_{свійський\_кіт}\ покладатися\_\ на\ x_{власний\_хазяїн}\}.$$

Після видобування знань із окремих речень виникає питання щодо об'єднання фактів в єдину пов'язану базу знань. Для цього кожному індивіду ставиться у відповідність токен з усіма його модифікаторами (amod, nmod, compound, flat тощо). Зрештою всі індивіди з однаковими мітками ототожнюються, як й індивіди, мітки яких пов'язані кореферентними зв'язками. Решта кореферентних зв'язків позначають аксіоми тотожності $A_{\text{parent\_coref}} \equiv A_{\text{child\_coref}}$.

Для якісної побудови баз знань з природномовних текстів необхідні розширення і специфікація правил виведення знань з розширеного дерева залежностей, у тому числі видобування та формалізація темпоральних характеристик, розв'язання проблеми пропущених токенів, наявність яких передбачається в реченні, зняття неоднозначностей у модифікаторах сурядних токенів та поповнення бази знань додатковими аксіомами, вилученими з інших джерел. Як такі джерела, зокрема, можна розглянути тезауруси та тлумачні словники.

Означення 9. Тезаурус — це семантичний словник певної природної мови, в якому слова пов'язані між собою лексико-семантичними відношеннями (наприклад, відношеннями рід–вид, частина–ціле, синонімією, кореляцією, асоціацією тощо) [23]. Тлумачний словник — це словник, що подає лексико-фразеологічний склад мови з поясненням значення, граматичних та стилістичних особливостей уживання його одиниць.

Запропонований підхід до видобування знань з природномовних текстів дає змогу поповнювати базу знань для будь-якої мови, використовуючи відповідні моделі для розв'язання задач розмічування частин мови, аналізу залежностей, пошуку кореферентностей тощо (рис. 8). Також подібний підхід дає змогу покращувати результат роботи простою заміною моделей машинного навчання для підзадач іншими, з кращими показниками.

Оскільки для розв'язання деяких задач (аналіз залежностей та розмічування частин мови) існує багато різноманітних моделей, їхні результати можна поєдну-

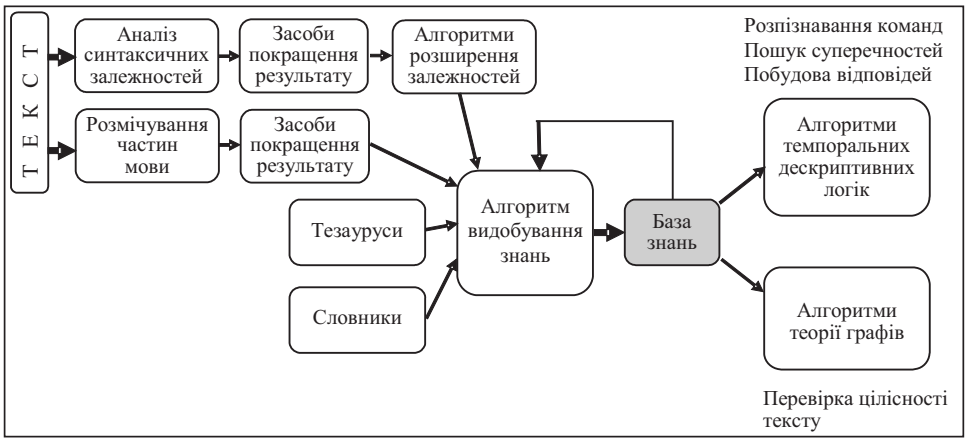

Рис. 8. Схема аналізу текстової інформації на основі баз знань

вати, щоб покращити якість вхідних даних для алгоритму видобування знань, зокрема, за допомогою теорії графів та алгоритмів над деревами.

Підхід тестувався з використанням трьох мов: англійської, української та російської. Наразі для російської мови наявний корпус кореферентностей, проте відсутні у вільному доступі моделі машинного навчання для розв'язання цієї задачі. Для української мови відсутній у вільному доступі також і корпус кореферентностей. Для повноцінної роботи системи необхідно побудувати якісні моделі пошуку кореферентностей для вказаних мов на основі кращих результатів розв'язання задачі для англомовних текстів.

3. АНАЛІЗ ЯКІСНИХ ХАРАКТЕРИСТИК ТЕКСТУ ЗА ДОПОМОГОЮ БАЗ ЗНАНЬ

Побудована на основі запропонованого підходу природномовна база знань дає змогу розв'язувати низку задач оброблення природної мови як за допомогою алгоритмів та методів дескриптивних логік різного рівня, так і аналізуючи її представлення з використанням апарату теорії графів. Як приклад типових задач, які можна розв'язувати за допомогою таких систем знань, наведемо, зокрема, розпізнавання команд, перевірку цілісності тексту, пошук суперечностей, побудову на основі бази знань діалогової системи для відповідей на запитання користувача, перевірку відповідності твердження поданому тексту тощо.

Якщо розглянути представлення бази знань, побудованої на основі природномовного тексту, у вигляді графу, де концепти зв'язані ребрами у разі участі в одній ролі їхніх індивідів чи наявності між ними ієрархічних зв'язків, задача перевірки тексту на цілісність зводиться до задачі перевірки графу на k-реберну зв'язність.

Означення 10. Граф $G = (E, V)$ називається k-реберно зв'язним, якщо для довільної підмножини ребер $X \subseteq E$ потужності $|X| < k$ граф $G' = (E \setminus X, V)$ є зв'язним.

Задачі пошуку суперечностей та перевірки відповідності текстів є прямими наслідками алгоритму семантичного табло для дескриптивних логік.

Так, на рис. 9 наведено найпростіший приклад нецілісного тексту. Йому відповідає графічне зображення фактів та аксіом дескриптивної логіки (див. рис. 9, *а*) та відповідний граф зв'язаних концептів (див. рис. 9, *б*). Граф має дві компоненти зв'язності, які відповідають зоологічній та комп'ютерній тематиці. Таким чином, можна зробити висновок щодо двох змістовних ліній у відповідному природномовному тексті, а отже, щодо його нецілісності.

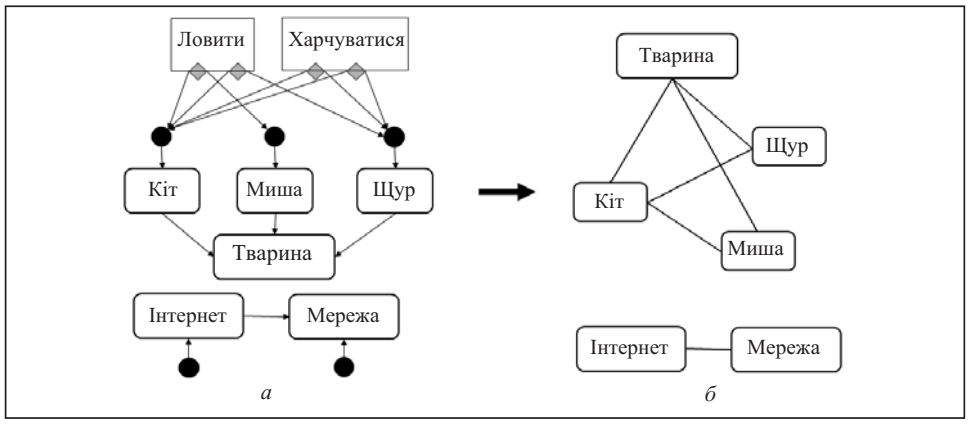

Рис. 9. База даних та її графове представлення

ВИСНОВКИ

У результаті розв'язання проблем оброблення природних мов та представлення неструктурованих текстів у вигляді графових чи деревовидних структур наразі можна отримати якісні вхідні дані для задачі наповнення баз знань фактами з текстів природної мови. Це дає поштовх до побудови систем алгоритмів для виявлення та аналізу природномовних знань на базі якісно розв'язаних задач оброблення природної мови, зокрема задач розмічування частин мови, побудови дерева залежностей, пошуку кореферентностей та виявлення іменованих сутностей.

У статті розглянуто теоретичні основи дескриптивних логік $\mathcal{ALCQ}$, які слугують мовою формального запису природномовних знань, наведено базові засади комп'ютерної лінгвістики для перетворення неструктурованого природномовного тексту у структурований вигляд. Запропоновано підхід до видобування знань зі структурованого подання природномовних текстів та їхнього запису засобами мови логіки $\mathcal{ALCQ}$. Розглянуто застосування алгоритмів логік $\mathcal{ALCQ}$ та теорії графів для аналізу добутих природномовних знань щодо цілісності та наявності суперечностей.

Наведений підхід до видобування знань з лінгвістичної інформації, добутої з тексту на основі задач аналізу залежностей, кореферентностей та частин мови, дає змогу здійснювати аналіз знань довільної мови за умови якісного розв'язання для неї відповідних лінгвістичних задач. Покращення запропонованого підходу можливе за рахунок розширення і специфікації правил виведення знань з розширеного дерева залежностей, в тому числі видобування та формалізації темпоральних, локативних, каузальних характеристик, розв'язання проблеми пропущених токенів, наявність яких передбачено в реченні, зняття неоднозначностей у модифікаторах сурядних токенів та поповнення бази знань додатковими аксіомами, вилученими з тезаурусів, тлумачних словників тощо.

СПИСОК ЛІТЕРАТУРИ

1. Baader F., Calvanese D., McGuinness D., Nardi D., Patel-Schneider P. The description logic handbook. Cambridge University Press, 2007. 578 p.
2. Кривий С.Л., Гогерчак Г.І. Логіка в математиці і інформатиці. *Праці першої української конференції «Логіка та її застосування»* (Київ, 26–28 листопада 2019 р.). Київ: АВАНПОСТ-ПРИМ, 2019. С. 47–55.
3. Lutz C., Wolter F., Zakharyaschev M. Temporal description logics: A survey. *Proc. of the 15th International Symposium on Temporal Representation and Reasoning* (Montreal, Canada, June 16–18, 2008). IEEE Computer Society, 2008. P. 3–14. https://doi.org/10.1109/TIME.2008.14.
4. Lutz C., Sturm H., Wolter F., Zakharyaschev M. Tableaux for temporal description logic with constant domains. *Proc. of First International Joint Conference, IJCAR 2001: Automated Reasoning* (Sienna, Italy, June 18–22, 2001). Springer, 2001. P. 121–136. https://doi.org/10.1007/3-540-45744-5_10.

5. Lai S., Leung K. S., Leung Y. SUNNYNLP at SemEval-2018 Task 10: A Support-Vector-Machine-based method for detecting semantic difference using taxonomy and word embedding features. *Proc. of The 12th International Workshop on Semantic Evaluation* (New Orleans, USA, June 5–6, 2018). 2018. P. 741–746. http://doi.org/10.18653/v1/S18-1118.
6. Zhan J., Zhao H. Span model for open information extraction on accurate corpus. *Proc. of the AAAI Conference on Artificial Intelligence.* 2020. Vol. 34, Iss. 5. P. 9523–9530. https://doi.org/10.1609/aaai.v34i05.6497.
7. Gangemi A., Presutti V., Reforgiato Recupero D., Nuzzolese A., Draicchio F., Mongiovi M. Semantic Web machine reading with FRED. *Semantic Web.* 2017. Vol. 8, Iss. 6. P. 873–893. https://doi.org/10.3233/SW-160240.
8. Reforgiato Recupero D., Nuzzolese A., Consoli S., Presutti V., Mongiovi M., Peroni S. Extracting knowledge from text using SHELDON, a Semantic Holistic framEwork for LinkeD ONtology data. *Proc. of the 24th International Conference on World Wide Web (WWW'15 Companion)* (Florence, Italy, May 2015). Association for Computing Machinery, 2015. P. 235–238. https://doi.org/10.1145/2740908.2742842.
9. Hoherchak H. Knowledge bases and description logics applications to natural language texts analysis. *Problems in Programming*. 2020. N 2–3. P. 259–269. https://doi.org/10.15407/pp2020.02-03.259.
10. Кривий С.Л., Дарчук Н.П., Провотар О.І. Онтологоподібні системи аналізу природномовних текстів. *Проблеми програмування.* 2018. № 2–3. С. 132–139.
11. Палагин А.В., Крывый С.Л., Петренко Н.Г. Знание-ориентированные информационные системы с обработкой естественно-языковых объектов: основы методологии и архитектурно-структурная организация. *УСиМ*. 2009. № 3. С. 42–55.
12. Палагин А.В., Крывый С.Л., Петренко Н.Г. Об автоматизации процесса извлечения знаний из естественно-языковых текстов. *Natural and Artificial Intelligence Intern. Book Series. Inteligent Processing*. Sofia: ITHEA, 2012. N 9. P. 44–52.
13. Палагин А.В., Крывый С.Л., Бибиков Д.С. Обработка предложений естественного языка с использованием словарей и частоты появления слов. *Natural and Artificial Intelligence Intern. Book Series. Inteligent Processing*. Sofia: ITHEA, 2010. N 9. P. 44–52.
14. McDonald R., Nivre J., Quirmbach-Brundage Y., Goldberg Y., Das D., Ganchev K., Hall K., Petrov S., Zhang H., Täckström O., Bedini C., Castelló N.B., Lee J. Universal dependency annotation for multilingual parsing. *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics* (Sofia, Bulgaria, August 4–9, 2013). Association for Computational Linguistics, 2013. (Vol. 2: Short Papers) P. 92–97.
15. Mrini K., Dernoncourt F., Bui T., Chang W., Nakashole N. Rethinking self-attention: An interpretable self-attentive encoder-decoder parser. *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 2020. P. 731–742. http://doi.org/10.18653/v1/2020.findings-emnlp.65.
16. Che W., Lui Y, Wang Y., Zheng B., Liu T. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. *Proc. of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (Brussels, Belgium, October 31 – November 1, 2018). Association for Computational Linguistics, 2018. P. 55–64. http:// doi.org/10.18653/v1/K18-2005.
17. Дарчук Н. Автоматичний синтаксичний аналіз текстів корпусу української мови. *Українське мовознавство*. 2013. № 43. С. 11–19.
18. Vilain M., Burger J., Aberdeen J., Connolly D., Hirschman L. A model-theoretic coreference scoring scheme. *Proc. of the 6th Message Understanding Conference (MUC-6) (*Maryland, USA, November 6–8, 1995). Association for Computational Linguistics, 1995. P. 45–52. https://doi.org/10.3115/1072399.1072405.
19. Stoyanov V., Gilbert N., Cardie C., Riloff E. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. *Proc. of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing* (Singapore, August 2–7, 2009). Association for Computational Linguistics, 2009. P. 656–664. http://doi.org/10.3115/1690219.1690238.
20. Luo X. On coreference resolution performance metrics. *Proc. of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05* (Vancouver, Canada, October, 2005). Association for Computational Linguistics, 2005. P. 25–32. http://doi.org/10.3115/1220575.1220579.

21. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Minneapolis, USA, June 2–7, 2019). Association for Computational Linguistics, 2019. Vol. 1 (Long and Short Papers). P. 4171–4186. http://dx.doi.org/10.18653/v1/N19-1423.

22. Xu L., Choi J.D. Revealing the myth of higher-order inference in coreference resolution. *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP) (online, November 16–20, 2020). Association for Computational Linguistics, 2020. P. 8527–8533. http://dx.doi.org/10.18653/v1/2020.emnlp-main.686.

23. Лукашевич Н.В. Тезаурусы в задачах информационного поиска. Москва: Изд-во Моск. ун-та, 2011. 512 с.

Надійшла до редакції 10.11.2020

Г.И. Гогерчак, Н.П. Дарчук, С.Л. Крывый

ПРЕДСТАВЛЕНИЕ, АНАЛИЗ И ИЗВЛЕЧЕНИЕ ЗНАНИЙ ИЗ НЕСТРУКТУРИРОВАННЫХ ЕСТЕСТВЕННОЯЗЫЧНЫХ ТЕКСТОВ

Аннотация. Приведен обзор средств дескриптивных логик для представления знаний из естественноязычных текстов, классификация дескриптивных логик по конструкторам концептов и ролей, а также описаны основные концепции темпоральных дескриптивных логик. Рассмотрен подход к построению систем анализа естественноязычных текстов на основе задач определения частей речи, поиска грамматических зависимостей и кореферентностей. Приведены примеры использования естественноязычных баз знаний для решения прикладных задач, в частности для проверки целостности текста, поиска противоречий.

Ключевые слова: дескриптивные логики, базы знаний, алгоритм семантического табло, извлечение знаний, обработка естественного языка, семантический анализ.

H. Hoherchak, N. Darchuk, S. Kryvyi

REPRESENTATION, ANALYSIS AND EXTRACTION OF KNOWLEDGE FROM UNSTRUCTURED NATURAL LANGUAGE TEXTS

Abstract. The article provides an overview of the means of descriptive logics for knowledge representation in natural-language texts. Descriptive logics are classified by constructors of concepts and roles, and the basic concepts of temporal descriptive logics are considered. The approach to construction of systems of the analysis of natural-language text based on problems of parts of speech tagging, dependency parsing, coreference resolution is considered. Examples of using natural-language knowledge bases to solve applied problems, in particular to check the integrity of the text and to reveal contradictions, are provided.

Keywords: description logics, knowledge bases, tableau algorithm, knowledge extraction, natural language processing, semantic analysis.

Гогерчак Григорій Іванович,
аспірант Київського національного університету імені Тараса Шевченка,
e-mail: gogerchak@gmail.com.

Дарчук Наталя Петрівна,
доктор філол. наук, професор, професор Київського національного університету імені Тараса Шевченка, e-mail: NataliaDarchuk@gmail.com.

Кривий Сергій Лук'янович,
доктор фіз.-мат. наук, професор, професор Київського національного університету імені Тараса Шевченка, e-mail: sl.krivoi@gmail.com.