

ОНТОЛОГОПОДІБНІ СИСТЕМИ АНАЛІЗУ ПРИРОДНОМОВНИХ ТЕКСТІВ

С.Л. Кривий, Н.П. Дарчук, О.І. Проватар

Розглянуто підхід до побудови онтологоподібних систем для аналізу текстів на природній мові (на разі тільки українською мовою). Такі системи будуються на основі формальних визначень онтології $O = (X, R, F, A(D, Rc))$ та синтаксичного і семантичного аналізу за допомогою елементів дескриптивної логіки. При побудові онтології вказується предметна область, до якої відносяться поняття з X і відношення із R . Фіксація області необхідна, оскільки сенс понять та відношень може бути різним в різних предметних областях. Специфікація області необхідна також для точнішого визначення інтерпретації F . Додаткові обмеження F описуються аксіоми A цієї області та обмеженнями Rc , які мають форму додаткових визначень (уточнення, обмеження на можливі значення і т. д.) властивостей області інтерпретації D . Вхідний текст T розглядається як область інтерпретації $D(T)$. З дескриптивної логіки використовуються визначення синтаксису понять та відношень. Використовуючи цей синтаксис, визначається тип логічної мови, яка в даному випадку є логікою предикатів першого порядку і стає можливим проводити логічне виведення та отримувати висновки з інформації, отриманої з вхідного тексту. Вхідний текст T розглядається як область інтерпретації F для понять X та відношень R , які з нього отримані. Результати синтактико-семантичного аналізу використовуються як вхідні дані для побудови онтології. В результаті такого аналізу відбувається розбиття об'єктів на групи, які відповідають частинам мови. Використовуючи цю інформацію та додаткові обмеження, будується онтологія даної предметної області. З точки зору користувача $D(T)$ виглядає як об'єкт реляційної бази даних, який складається лише з унарних та бінарних відношень. Різниця полягає в тому, що в класичних БД відношення замкнуті в рамках даної бази даних, в той час як $D(T)$, як база даних, є відкритою для даної ПО, тому що нормалізоване представлення знань виконується в ситуації, коли немає припущення про повноту цих знань. Крім того, $D(T)$ зв'язує семантичними відношеннями концепти C і відношення R із R , що немає аналогу в семантиці реляційних БД.

Ключові слова: онтологія, дескриптивні логіки, синтактико-семантичний аналіз.

Рассматривается подход к построению онтологоподобных систем для исследования естественных языковых текстов (на данное время только украинского языка) на основе синтактико-семантического анализа с использованием дескриптивных логик. Такие системы строятся на основе формального определения онтологии $O = (X, R, F, A(D, Rc))$ и синтаксического и семантического анализа с помощью элементов дескриптивной логики. При построении онтологии фиксируется предметная область, к которой относятся концепты из X и отношения из R . Фиксация области необходима, поскольку смысл концептов и отношений может быть разным в разных предметных областях. Спецификация области необходима ещё и для более точного определения интерпретации F . Дополнительные ограничения описываются аксиомами A этой области и ограничениями Rc , которые имеют вид дополнительных определений (уточнения, ограничения на возможные значения и т. д.) свойств области интерпретации D . Входной текст T рассматривается как область интерпретации $D(T)$. Из дескриптивной логики используются определения синтаксиса концептов и отношений. Используя этот синтаксис, определяется тип логического языка, который в данном случае является логикой предикатов первого порядка и появляется возможность проводить логический вывод и получать следствия из информации, полученной из входного текста. Входной текст T рассматривается как область интерпретации F для концептов из X и отношений R , которые из него получены. Результаты синтактико-семантического анализа используются как входные данные для построения онтологии. В результате такого анализа происходит разбиение объектов на группы, которые соответствуют частям языка. Используя эту информацию и дополнительные ограничения, строится онтология данной предметной области. С точки зрения пользователя $D(T)$ выглядит как объект реляционной базы данных, который состоит только лишь из унарных и бинарных отношений. Разница состоит в том, что в классических БД отношения замкнуты в рамках данной базы данных, в то время как $D(T)$, как база данных, является открытой для данной ПО, потому что нормализованное представление знаний используется в ситуации, когда нет предположения о полноте этих знаний. Кроме того, $D(T)$ связывает семантическими отношениями концепты C и отношения R из R , чего не имеет аналога в семантике реляционных БД.

Ключевые слова: онтология, дескриптивные логики, синтактико-семантический анализ.

In this paper an approach to building of Ontological similar systems for analysis of texts in natural language (on this time only Ukrainian languages) is considered. Such systems are built on the base of formal definitions of ontology $O = (X, R, F, A(D, Rc))$ and syntactical and semantic analysis by using elements of Descriptive logics. When constructing an ontology, a subject domain is indicated and includes concepts from X and relations from R . Such indicates are necessary because of semantic sense of concepts and relations can be depend of the domain. Specification of domain is necessary to define more precision of the interpretation F . Additional adjustments of F are described by the axioms A of this domain and the restrictions Rc , which have the form of additional definitions (clarifications, constraints on possible values, etc.) of the properties of the domain of interpretation D . An input text T considered as domain of interpretation D .

From Descriptive logics we use definition of syntax of concepts and relations. Using this syntax of concepts and definitions we obtain first order logic and can do reasoning and obtain conclusions from information derived from input text. The input text T considered as the domain D of interpretation F for concepts X and relations R which are derived from it.

Syntactical and semantic result of analysis is used as input data for building of ontology. As a result of such analysis is the partition of objects on the groups which are correspond of the parts of speech. By using this information and additional adjustments we build data knowledge of the input domain.

In terms of user the domain D looks like an object of the relational database, which includes only unary and binary relations. The difference lies in the fact that classical database relationships are closed within this domain, while D as a database is open because the normalized representation of knowledge is executed in a situation where there is no assumption about the completeness of these knowledge. In addition, D relates the semantic relations of the concepts and the relation, which is not analogous to the relational database semantics.

Key words: Ontology, Description Logics, syntactical and semantic analysis.

1. Вступ

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

$$O = (X, R, F),$$

де X – скінченна множина концептів (понять), R – скінченна множина бінарних (семантичних) відношень, визначених на X , і F – функція інтерпретації на деякій області D елементів із X і R , тобто $F: X \cup R \rightarrow D$.

При побудові онтології вказується предметна область (ПО), до якої відносяться концепти із X і відношення із R . Конкретизація ПО необхідна для означення інтерпретації F . Зв'язок F з ПО може вносити додаткові корективи в означення F . Додаткові корективи описуються аксіомами A даної ПО і обмеженнями R_c , які мають вигляд додаткових дефініцій (уточнень, обмежень на можливі значення тощо) властивостей із області інтерпретації D даної ПО. Таким чином, приходимо до уточненого означення онтології для конкретної ПО:

$$O = (X, R, F, A(D, R_c)).$$

Різниця між означеннями (1) і (2) полягає в наступному [1].

а) множина концептів X в (1) орієнтована на проблему, яку потрібно розв'язати, а в (2) ця множина повинна бути максимально повною для даної ПО і повинна будуватися автоматизованими засобами;

б) множина R в (1) суб'єктивна, а в 2) повинна виконуватися на множині D , будуватися автоматизованими засобами та перевірятися на сумісність системою логічного виведення та, можливо, експертами в даній ПО;

в) функція інтерпретації F в (1) обирається користувачем відповідно до його професійної компетенції, власної або довідкової інформації, а в (2) ця функція формується, виходячи із загальнозначимих джерел текстової інформації – енциклопедій, тлумачних словників, результатів синтактико-семантичного аналізу тощо;

г) множина аксіом A описує додаткові специфічні означення концептів із D і обмеження інтерпретації R_c даної ПО. Множина D включає значення концептів із X і R , а також вводить концепти, які не ввійшли до X і R . Цим самим уточнюються і довізнаються специфічні властивості концептів і відношень даної ПО. Ці уточнення ґрунтуються на думці кола експертів в даній ПО. Множина R_c включає лише значимі обмеження на інтерпретацію і може бути, зокрема, пустою множиною. Наприклад, обробляється лише «діловий стиль» або «дискретна математика: комбінаторика».

Таким чином, означення (2) містить декларативні і процедурні засоби даної ПО, що дає можливість розв'язувати завдання користувача.

1. Основні компоненти системи

Зі сказаного у вступі, завдання, які впливають з означення (2), зводяться до наступних.

1.1. Визначити область інтерпретації онтології ПО. Обов'язково потрібно зафіксувати предметну область. Необхідність цієї фіксації полягає в тому, що одні й ті самі поняття в різних ПО можуть мати різні значення. На вхід системи подаються тексти (на разі лише українською мовою, хоча передбачається подальший розвиток системи шляхом включення інших мов, зокрема, російської та англійської), які відносяться до даної ПО.

Областю інтерпретації концептів і відношень онтології виступає множина текстів T , за якою будується термінологія ПО, в якій інтерпретується множина концептів X і множина смислових відношень R . Для множини X область інтерпретації X^F розбивається на класи (наприклад, імена власні/невласні, назви осіб, назви абстрактні/конкретні, 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 певну 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 (рисунок).

Приклад 1. Проаналізуємо речення "Склянку борошна необхідно залити склянкою холодного молока". Синтаксичний аналіз дасть на виході такий ациклічний орграф (за відношенням "Хазяїн-Слуга"):

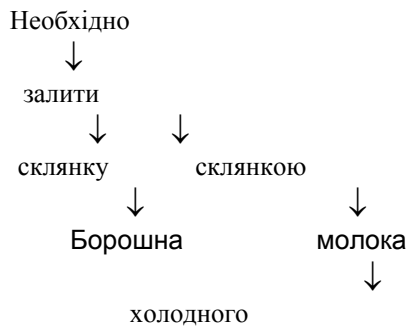


Рисунок. Результат синтаксичного аналізу речення

Семантичне представлення цього речення матиме такий вигляд:

1. КІЛЬК(ОДИН, СКЯНКА),
2. КІЛЬК(ОДИН, БОРОШНО),
3. КІЛЬК(ОДИН, СКЯНКА),
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 випливає 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 $A(D, R_c)$ уточнюють концепти із X і вводять (наприклад, в діалозі з користувачем) нові концепти і відношення. Так, для атомарного концепту «молоко» уточнення «молоко холодне» потребує введення нового концепту «холодне», який виділяє специфічні властивості (ознаки) концепту «молоко» із множини значень його інтерпретації. Детальніше про це далі.

Область інтерпретації ПО будемо позначати $D(T)$.

1.2. Визначення синтаксису мови концептів. Для розробки представлення концептів із X і R пропонується такий синтаксис [3, 4]: нехай A і B довільні концепти із X і R із R , тоді

- $A, B ::= C$ (атомарний концепт)
- | \square (універсальний концепт)
 - | \square (пустий концепт)
 - | $\neg A$ (заперечення атомарного концепту)
 - | $C \square D$ (перетин)
 - | $\forall R. C$ (обмежене значення)
 - | $\exists R. \square$ (обмежений квантор існування).

Зауважимо, що заперечення застосовується лише до атомарних концептів, а для квантора існування областю його дії є універсальний концепт. До цього синтаксису додається можливість опису індивідуальних імен, Ці імена називають *номіналами* і вводяться в дескриптивну мову за допомогою конструктора «множини» $\{a_1, a_2, \dots, a_n\}$. Цей конструктор перетворює сукупність індивідуальних імен у концепт.

Наведений синтаксис орієнтується на теоретико-множинну модель семантики: нехай $D(T)$ – область інтерпретації, тоді

- $\square^F = D(T)$,
- $\perp^F = \emptyset$,
- $\{a_1, a_2, \dots, a_n\}^F = \{a_1^F, a_2^F, \dots, a_n^F\}$,

- $(\neg C)^F = D(T) \setminus C^F$,
- $(C \square B)^F = C^F \cap B^F$,
- $(\forall R.C)^F = \{a \in D(T) : \forall b (a,b) \in R^F \rightarrow b \in C^F\}$,
- $(\exists R.\square)^F = \{a \in D(T) : \exists b (a,b) \in R^F\}$,

де $R^F \subseteq D(T) \times D(T)$, $C^F \subseteq D(T)$, C – атомарний концепт, R – атомарна роль (бінарне відношення).

Крім того, будемо використовувати наступні додаткові конструктори концептів:

- об'єднання концептів $C \square B$ з інтерпретацією $(C \square B)^F = C^F \cup B^F$;
- повний квантор існування $(\exists R.C)^F = \{a \in D(T) : \exists b (a,b) \in R^F \wedge b \in C^F\}$;
- числове обмеження $\geq nR$ (не менше) і $\leq nR$ (не більше) з інтерпретацією

$$(\geq nR) = \{a \in D(T) : |\{b:(a,b) \in R^F\}| \geq n\}, \quad (\leq nR) = \{a \in D(T) : |\{b:(a,b) \in R^F\}| \leq n\}.$$

- заперечення довільного концепту $\neg C$ з інтерпретацією $(\neg C)^F = D(T) \setminus C^F$.

Зауважимо, що об'єднання і квантор існування можна було б і не вводити, а скористатися законами

$$C \square B = \neg(\neg C \square \neg B) \text{ і } \exists R.C = \neg \forall R.\neg C.$$

Єдиними термінологічними аксіомами для концептів і відношень є аксіоми вигляду:

$$B \square C (R \square Q) \text{ або } B \equiv C (R \equiv Q)$$

з наступною інтерпретацією. Кажуть, що аксіома $B \square C (R \square Q)$ виконується в інтерпретації F , якщо $B^F \subseteq C^F (R^F \subseteq Q^F)$. Аналогічно, аксіома $B \equiv C (R \equiv Q)$ виконується в інтерпретації F , якщо $B^F = C^F (R^F = Q^F)$.

Обмежимося введеними конструкторами визначення нових концептів та термінологічними аксіомами і розглянемо, що можна виразити за їх допомогою.

Приклад 2. Нехай атомарними концептами є «особа, стать-ж, вмістилище», а атомарними бінарними відношеннями є «має» і «складається-із».

Використовуючи ці атомарні об'єкти, можна побудувати термінологію ПО «адреса особи»

кімната-жила	\square вмістилище \square \exists має.особа
номер	$\equiv \{1, \dots, 100\}$
квартира	$\square \geq 1$ складається-із.кімната-жила
квартира	$\square \exists$ має.номер
будинок	$\square \exists$ має.номер \square має.квартира
жінка	\square особа \square стать-ж
чоловік	\square особа $\square \neg$ жінка
заміжня-жінка	\square жінка $\square \exists$ має.чоловік
чоловік-жонатий	\square чоловік $\square \exists$ має. жінка
особа-з-адресою	\square особа $\square \exists$ має.будинок

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

1.3. Мова концептів як фрагмент логіки предикатів. Семантика мови концептів, яка наведена вище, говорить про те, що вона являє собою фрагмент 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 $D(T)$. А це дає можливість транслювати у формули логіки предикатів

$f_{C(x)}$ атомарні концепти з однією вільною змінною x такою, що для довільної інтерпретації F множина елементів із $D(T)$, які виконують $f_{C(x)}$, збігається з множиною C^F . При цьому атомарний концепт C

транслюється у формулу $C(x)$, а конструктори \square, \square, \neg – у логічні диз'юнкцію, кон'юнкцію та заперечення відповідно.

Якщо C вже странсьоване у формулу $\psi_{C(x)}$ і R – атомарне бінарне відношення, то квантори існування і загальності виражаються формулами:

$$\psi_{\exists R.C(x)}(y) = \exists x.R(y, x) \wedge \psi_{C(x)}, \quad \psi_{\exists R.C(y)}(y) = \forall x.R(y, x) \rightarrow \psi_{C(x)},$$

де y – нова змінна, а числові обмеження виражаються формулами

$$\psi_{\geq nR}(x) = \exists y_1, \dots, \exists y_{n+1}. R(x, y_1) \wedge \dots \wedge R(x, y_{n+1}) \wedge \bigwedge_{i < j} y_i \neq y_j,$$

$$\psi_{\leq nR}(x) = \forall y_1, \dots, \forall y_{n+1}. R(x, y_1) \wedge \dots \wedge R(x, y_{n+1}) \wedge \bigvee_{i < j} y_i = y_j.$$

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

1.4. Об'єкти в області інтерпретації $D(T)$. Оскільки реальні об'єкти онтосистеми належать області інтерпретації $D(T)$, то необхідно описати світ, в якому вони інтерпретуються. Цей опис представляється в термінах концептів і бінарних відношень, які беруться із області інтерпретації $D(T)$. Деякі атомарні концепти і відношення можуть визначатися як термінологічні імена і 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:

$$C(a) \text{ і } R(b, c),$$

де $C(a)$ означає властивість « a належить концепту C », а $R(b, c)$ означає властивість відношення, яке означає, що « c наповнення відношення R для b ». Наприклад, якщо атомарними концептами виступає B =ІМ'Я, C =ФАХ, атомарним бінарним відношенням є R =БАТЬКО, а іменами індивідуумів в $D(T)$ є ПЕТРО, ІВАН, СТЕПАН, то інтерпретацією концепту B^c є множина {ПЕТРО, ІВАН, СТЕПАН}, C^F = {ЛІНГВІСТ, ЛІКАР, ПРОГРАМІСТ}, а R^F = {(ПЕТРО, ІВАН), (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 $D(T)$ 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 $D(T)$, як база даних, є відкритою для даної ПО, тому що нормалізоване представлення знань виконується в ситуації, коли немає припущення про повноту цих знань. Крім того, $D(T)$ зв'язує семантичними відношеннями концепти C і відношення R із R , що немає аналогу в семантиці реляційних БД.

Приклад 3. Нехай на вході є текст:

«Поняття множини належить до аксіоматичних понять математики, і точне його означення дати неможливо. Часто приймається формулювання інтуїтивного поняття множини Г. Кантора, який є основоположником цієї теорії.

Означення. *Довільна сукупність певних предметів нашої інтуїції чи інтелекту, які можна відрізнити один від одного і які уявляються як єдине ціле, називається множиною. Предмети, які входять до складу множини, називаються її елементами*».

Суттєвим пунктом канторівського розуміння множини є те, що зібрання предметів розглядається як один предмет («уявляється як єдине ціле»). Основна увага тут переноситься з окремих предметів на зібрання предметів, які, в свою чергу, можна розглядати як предмети.

Що стосується «предметів нашої інтуїції чи інтелекту», то це формулювання дає значну свободу, насамперед, тим, що ніяк не обмежує природу предметів, з яких складається множина. Множина може складатися, наприклад, з людей, точок площини, простих чисел, планет Всесвіту. Зауважимо також, що канторівське формулювання множини дає змогу розглядати множини, елементи яких з певних причин точно визначити неможливо. У зв'язку з цим згадаємо, що елементи будь-якої нескінченної множини неможливо зібрати, навіть теоретично, в скінченну сукупність. Відомі також скінченні множини, які мають таку ж міру невизначеності, як нескінченна множина.

З'ясуємо, нарешті, зміст висловів: «які можна відрізнити один від одного» і «певні предмети». У першому випадку для будь-яких двох предметів, що розглядаються як елементи даної множини, повинна існувати можливість з'ясувати, чи різні ці предмети, чи однакові. У другому випадку, якщо задані деяка

множина і який-небудь предмет, то можна визначити, чи цей предмет є елементом даної множини чи ні. Звідси випливає, що всяка множина повністю визначається своїми елементами. Ця канторівська вимога формулюється у вигляді аксіоми.

Аксіома екстенсіональності. Дві множини рівні тоді і тільки тоді, коли вони складаються з одних і тих же елементів. Рівність двох Множин A і B позначають $A = B$.

Отже, дві множини рівні, якщо кожний елемент однієї з них є елементом другої, і навпаки.

Множина називається скінченною, якщо вона складається із скінченного числа елементів. Запис $a \in A$ ($a \notin A$) означає, що a є (не є) елементом множини A . Однозначно визначена множина, елементами якої є a_1, a_2, \dots, a_n , позначається $\{a_1, a_2, \dots, a_n\}$.

Семантико-синтаксичний аналіз розбиває концепти і смислові відношення, які знаходяться в цьому тексті, на такі класи (таблиця).

Таблиця

Концепти	Відношення	Уточнення (ознаки)
Аксіома	Однакові	Скінченна(ий)
Визначення	Належить	Нескінченна(ий)
Множина	Включає	Певна(ий)
Елемент	Рівні	Задана(ий)
Сукупність	Приймається	Дана(ий)
Предмети	Позначається	Будь-яка(ий)
Число	Вважається	Довільна(ий)
Висловлювання	Складається	Канторівський
Теорія	Еквівалентні	Інтуїтивне(а,ий)
Інтуїція	Визначається	
Інтелект	Обмежує	
Екстенсіональність	Має	
Однозначність		
Основоположник		
Ознака		
Побудова		
Формулювання		
Пункт		
Природа		

Тепер виконаємо поповнення перерахованих в таблиці концептів, використовуючи як абстрактні атомарні концепти «сукупність, елемент, A, B, C, \dots » і атомарні бінарні відношення «має, належить, складається-із'» для побудови онтології ПО «Елементи теорії множин»:

Множина	<input type="checkbox"/> сукупність <input type="checkbox"/> має.елемент
множина-скінченна	$\equiv \{a_1, a_2, \dots, a_n\}$
елем-множини	<input type="checkbox"/> множина <input type="checkbox"/> \exists належить.елемент
множина-пуста	\equiv множина <input type="checkbox"/> $\neg \exists$ має.елемент
алфавіт-число	$\equiv \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
слово-в-алфавіті	<input type="checkbox"/> $\geq n$ складається-із.алфавіт-число
натуральне-число	<input type="checkbox"/> \forall складається-із.слово-в-алфавіті
натуральні-числа	\equiv множина <input type="checkbox"/> \forall складається-із.натуральне-число
<input type="checkbox"/> підмножина. <input type="checkbox"/> A <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> підмножина. $B \subseteq A \subseteq B$ – термінологічна аксіома	
<input type="checkbox"/> підмножина. <input type="checkbox"/> A <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> підмножина. $B \subseteq A \equiv B$ – термінологічна аксіома	
об'єднання-множин	$\equiv A \cup B$
перетин-множин	$\equiv A \cap B$

Ця побудова, в разі необхідності введення нових концептів і відношень, може продовжуватися далі. Наприклад, до цих означень додаються закони комутативності для операцій об'єднання та перетину і властивість подвійного доповнення:

$$\begin{aligned} A \sqcup B &\equiv B \sqcup A \\ A \sqcap B &\equiv B \sqcap A \\ \neg(\neg A) &\equiv A. \end{aligned}$$

Подібним чином вводяться й інші закони алгебри множин.

Після виконання дефініцій і додаткових уточнень настає етап трансляції введених концептів і відношень у формули логічної мови. Логічна мова може вибиратися в залежності від поставлених вимог до онтосистеми. Але, як було сказано вище, такою мовою, як правило, виступає мова предикатів першого порядку. А це означає, що алфавітом такої мови є множина концептів X , елементарними предикатами виступають атомарні відношення, (можливо, параметризовані), тобто предикати, які відповідають елементам множини R . Ця множина відношень може поповнюватися новими відношеннями, введеними за допомогою дефініцій.

Повернемося до вищенаведеного приклада.

Приклад 4. Дано. Алфавіт $X = \{\text{множини} = \{A, D, \dots, A_1, B_1, \dots\}, \text{елементи} = \{a, b, \dots, s, a_n, \dots\}, \text{відношення}$
 $R = \{\text{належить } (a, A), \text{ складається-із}(a_i)\}$.

Запитання: чи справедливий закон дистрибутивності для операцій об'єднання і перетину, тобто чи вірно, що

$$A \sqcup (B \sqcap C) = (A \sqcup B) \sqcap (A \sqcup C).$$

Трансляція в мову предикатів перетворює даний вираз до вигляду (який зрозумілий системі логічного виведення).

Мета – довести (методом включення в обидві сторони): «підмножина $(A \sqcup (B \sqcap C), (A \sqcup B) \sqcap (A \sqcup C))$ \wedge підмножина $((A \sqcup B) \sqcap (A \sqcup C), A \sqcup (B \sqcap C))$ ». Для доведення виконується трансляція формули, яка йде знизу-вверх за вищенаведеними дефініціями. Конструктори \sqcup і \sqcap інтерпретуються відповідно теоретико-множинними операціями \cup і \cap . Отже, потрібно довести включення

$$\langle\langle (A \cup (B \cap C)) \cap \forall \text{ належить. } a \rangle\rangle \subseteq \langle\langle (A \cup B) \cap (A \cup C) \rangle\rangle \cap \forall \text{ належить. } a \rangle\rangle.$$

Доведення: Припущення: нехай $\langle\langle (A \cup (B \cap C)) \cap \text{належить. } a \rangle\rangle$, тоді $\langle\langle (A \cap \text{належить. } a) \cup ((B \cap C) \cap \text{належить. } a) \rangle\rangle \rightarrow \langle\langle (A \cap \text{належить. } a) \cup ((B \cap \text{належить. } a) \cap (C \cap \text{належить. } a)) \rangle\rangle \rightarrow \langle\langle (A \cup B) \cap \text{належить. } a \rangle\rangle \cup \langle\langle (A \cup C) \cap \text{належить. } a \rangle\rangle \rightarrow \langle\langle (A \cap B) \cup (A \cap C) \cap \text{належить. } a \rangle\rangle$.

Результат: включення вірне: $\langle\langle (A \cup (B \cap C)) \rangle\rangle$ є підмножиною $(A \cap B) \cup (A \cap C)$. Обернене включення доводиться аналогічно. В підсистемі логічного виведення такі доведення можна виконати методом резолюцій або методом семантичного табло.

Висновки

Підводячи підсумок з вищесказаного, отримуємо таку послідовність етапів побудови і аналізу текстів в онтологоподібній системі:

- 0) зафіксувати предметну область, до якої відноситься онтологія (корінь онтології);
- 1) виконати семантико-синтаксичний аналіз вхідного тексту T ;
- 2) побудувати розбиття синтаксичних і семантичних одиниць тексту T на класи (концепти, відношення, обмеження тощо), тобто побудувати множини X і R та область інтерпретації $D(T)$;
- 3) виконати трансляцію елементів множини концептів і відношень у формули логіки предикатів першого порядку;
- 4) виконати поповнення множини концептів і відношень новими концептами і відношеннями (в разі необхідності);
- 5) використовуючи систему логічного виведення, виконати перевірку генерації логічних наслідків на наборі тестових запитів до онтосистеми;

6) якщо в процесі логічного виведення необхідно виконати поповнення концептів і відношень, то перейти на крок 5) (таке поповнення виконується в діалозі системи з користувачем і його необхідність виникає в результаті неповноти фактів, потрібних для завершення процесу виведення).

Література

1. Палагін О.В., Петренко М.Г. Тлумачний онтографічний словник з інженерії знань. Київ: Інтерсервіс. 2017. 477 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. and other. The Description Logic Handbook. Cambridge: University Press. 2007. 601 p.
4. Кривий С. Л., Дарчук Н.П., Ясенова И.С., Головина А.Л., Соляр А.С. Методы и средства систем представления знаний. Publisher: ITHEA. Inter. Journ. «Information Content and Processing». 2017. Vol. 4. № 1. С. 62–99.

References

1. Palagin A.V., Petrenko N.G.. Ontografical explanatory dictionary for engineering knowledge. Kiev: Interservice. 2017. 477 p.
2. Darchuk N.P. Computer annotation of texts in Ukrainian language: results and perspectives. Kiev: Ukraine Education. 2013. 543 p.
3. Baader F., Calvanese D., McGuinness D.L. and other. The Description Logic Handbook. Cambridge: University Press. 2007. 601 p.
4. Krivyi S.L., Darchuk N.P., Jasenova I.S., Golovina A.L., Soliar A.S. Methods and tools of systems for knowledge representation. Publisher: ITHEA. – Inter. Journ. «Information Content and Processing». 2017. Vol. 4. N 1. P. 62–99.

Про авторів:

Кривий Сергій Лук'янович,

доктор-фізико-математичних наук, професор,
професор Київського національного університету імені Тараса Шевченка.
Кількість наукових публікацій в українських виданнях – 155.
Кількість наукових публікацій в зарубіжних виданнях – 60.
Індекс Хірша – 4.
<http://orcid.org/0000-0003-4231-0691>,

Дарчук Наталія Петрівна,

доктор філологічних наук,
професор Київського національного університету імені Тараса Шевченка.
Кількість наукових публікацій в українських виданнях – 155.
Кількість наукових публікацій в зарубіжних виданнях – 16.
<http://orcid.org/0000-0003-4564-0872>,

Проватар Олександр Іванович,

доктор-фізико-математичних наук, професор,
професор Київського національного університету імені Тараса Шевченка.
Кількість наукових публікацій в українських виданнях – 100.
Кількість наукових публікацій в зарубіжних виданнях – 30.
Індекс Хірша – 4.
<http://orcid.org/0000-0066-0278-5897>.

Місце роботи авторів:

Київський національний університет імені Тараса Шевченка,
01601, Київ, вул. Володимирська, 60.

Тел.: +38 (097) 334 6056,
+38 (050) 443 5200,
+38 (050) 444 1705.

E-mail: krivoi@i.com.ua
sl.krivoi@gmail.com,
nataliadarchuk@gmail.com,
aprovata@unicyb.kiev.ua