

ЗАСТОСУВАННЯ МЕТОДУ ДЕКОМПОЗИЦІЇ ПРЕДИКАТІВ ДЛЯ РЕАЛІЗАЦІЇ ДОВІЛЬНИХ ВІДНОШЕНЬ РЕЛЯЦІЙНИМИ МЕРЕЖАМИ

Наразі широке поширення отримали розробки, спрямовані на вдосконалення процесу автоматизованого проектування цифрових пристроїв, які можуть виконувати функції природномовного інтелектуального інтерфейсу, проблема створення й впровадження якого поглинає всі сфери людської діяльності. Це пов'язано насамперед зі специфікою кожної предметної області, на впровадження в яку й розрахований інтелектуальний інтерфейс. Тому розробка саме універсальних методів та моделей є актуальною на сьогодні. Розвиток способів формального представлення [1; 2] довільних відношень та їх подальшої схемної реалізації приводить до значного підвищення швидкодії таких систем штучного інтелекту [3].

Одним з ефективних способів реалізації природномовних структур є подання інформації у вигляді реляційних мереж, направлених на широке розпаралелювання знань у процесі їх обробки [3; 4]. Так було побудовано реляційну мережу для закінчення повних неперисвійних прикметників [5], принципи побудови якої поширилися і на інші частини мови (було досліджено словозміну іменників [6] та дієслів [7]). Кожна гілка реляційної мережі є лінійним логічним перетворенням. Метод багат шарової декомпозиції предикатів можна також вважати методом для знаходження лінійних логічних перетворень. Він дає можливість моделювати різні інформаційні процеси. Представлення структури об'єкта як математичної моделі спрощує його проектування на апаратному рівні.

При побудові реляційних мереж дослідники іноді доводиться вирішувати завдання за

$$E_1(x'_1, x''_1) = \forall x_2 \in A_2 \quad \forall x_3 \in A_3 \quad (P(x'_1, x_2, x_3) \approx P(x''_1, x_2, x_3)).$$

Теорема про супроводжуючу еквівалентність. Нехай предикат $P(x_1, x_2, x_3, \dots, x_n)$ визначений на декартовому добутку множин $A_1 \times A_2 \times \dots \times A_n$. Тоді

$$E_i(x'_i, x''_i) = \forall x_1 \in A_1 \forall x_2 \in A_2 \dots \forall x_{i-1} \in A_{i-1} \forall x_{i+1} \in A_{i+1} \dots \forall x_n \in A_n \\ (P(x_1, x_2, \dots, x_{i-1}, x'_i, x_{i+1}, \dots, x_n) \approx$$

таких умов визначення предметної області, для формалізації і подальшого розв'язання яких доцільно вводити n -арні предикати, оскільки бінарних предикатів виявляється замало.

Таким чином, постає завдання узагальнити метод розшарування бінарних предикатів [8]. **Метою** даної статті є розробка методу тришарової декомпозиції саме n -арних предикатів на прикладі створення математичної моделі флексії повних неперисвійних прикметників.

1. Двошарова декомпозиція предикатів 1-го роду. Метод тришарової декомпозиції n -арних предикатів поєднує методи двошарової декомпозиції предикатів 1-го та 2-го роду. Для двошарової декомпозиції предикатів 1-го роду вихідний предикат необхідно представити через свої характеристичні функції (сюр'єкції) і деякий предикат L (визначений на множині меншої розмірності).

Далі дослідження будемо проводити на прикладі закінчень повних неперисвійних прикметників. Існує всього 13 пар закінчень повних неперисвійних прикметників:

-ий, -ого, -ому, -им, -ом, -ая, -ую, -ой, -ою,
-оє, -ье, -ых, -ыми
-ий, -его, -ему, -им, -ем, -яя, -юю, -ей, -ею,
-еє, -ие, -их, -ими

Необхідно розглянути тернарний предикат і відшукати для нього такий вигляд, за якого порівняння значень трьох відповідних функцій f_1 , f_2 і f_3 виконувалося б за допомогою найпростішого в деякому сенсі предикату.

З цією метою доцільно ввести поняття супроводжуючих еквівалентностей предиката, яку виразимо такою формулою:

$$P(x_1, x_2, \dots, x_{i-1}, x''_i, x_{i+1}, \dots, x_n).$$

Таким чином,

$$P(x_1, x_2, \dots, x_n) = L(f_1(x_1), f_2(x_2), \dots, f_n(x_n)).$$

$$P(x_1, x_2, x_3) = L(f_1(x_1), f_2(x_2), f_3(x_3)).$$

Якщо замінити в останньому виразі $f_i(x_i)$ на v_i , знайдемо

$$L(v_1, v_2, v_3) = P(f_1^{-1}(v_1), f_2^{-1}(v_2), f_3^{-1}(v_3)).$$

Далі необхідно побудувати таблицю предиката $P(x_1, x_2, x_3)$, яка б характеризувала зв'язок між змінною x_1 і відношенням (x_2, x_3) , тобто між першою, другою, третьою літерами закінчення. Якщо закінчення (x_1, x_2, x_3) наявне в одному з можливих 26 закінчень повних неприсвійних прикметників, то на перетині стовпчика і рядка в таблицю заносять «1», якщо ні – «0».

Зв'язку між другою та першою, третьою літерами закінчення відповідає супроводжуюча еквівалентність, яку можна виразити таким чином:

$$E_2(x'_2, x''_2) = \forall x_1 \in A_1 \forall x_3 \in A_3 (P(x_1, x'_2, x_3) \approx P(x_1, x''_2, x_3)).$$

Третя супроводжуюча еквівалентність, яка описує відношення, що пов'язує третю літеру закінчень повних неприсвійних прикметників з першою та другою, можна представити у такому вигляді:

$$E_3(x'_3, x''_3) = \forall x_1 \in A_1 \forall x_2 \in A_2 (P(x_1, x_2, x'_3) \approx P(x_1, x_2, x''_3)).$$

Для подальших досліджень треба отримати класи розбиття. Для цього об'єднують однакові рядки (або стовпчики) таблиці. Шарам (класам) розбиття дають відповідні назви (як правило, це перша літера запису відповідного шару).

2. Двошарова декомпозиція предикатів 2-го роду. Двошарова декомпозиція предикатів 2-го роду дозволяє представити вихідний предикат через відображення й елементарний предикат, єдиний для всіх предикатів, подібний до предиката рівності. Таким чином, двошарова декомпозиція предикатів 2-го роду – це наступний крок до представлення відношення узагальненим і простим виразом.

Для цього необхідно записати характеристичні функції предиката P у вигляді функцій $f_i(x_i) = v_i$, еквівалентності E_i , ($i = 1, 3$) для трьох літер закінчення.

Далі необхідно записати образ предиката P , який описує закінчення повних неприсвійних прикметників російської мови. Для цього предикат P записують так:

$$P(x_1, x_2, x_3) = L(f_1(x_1), f_2(x_2), f_3(x_3)),$$

звідки знаходимо

$$L(v_1, v_2, v_3) = P(f_1^{-1}(v_1), f_2^{-1}(v_2), f_3^{-1}(v_3)).$$

Далі необхідно скласти таблиці предиката L за такою схемою: замінити літери іменами

шарів розбиття, до яких вони належать, x_i – на v_i , ($i = \overline{1,3}$), P – на L , а потім стовпчики й рядки, які повторюються, виключити з таблиці.

Для того, щоб отримати на наступному етапі системи рівнянь, доцільно скористатись теоремою про імплікативне розкладання предиката. Причому для подальшої апаратно-програмної реалізації вважаємо за доцільне оцінити обидві системи рівнянь (імплікативне розкладання за рядками і за стовпчиками), враховуючи кількість рівнянь, кількість змінних та кількість значень змінних.

Слід відзначити, що складність схем, які реалізують отримані системи рівнянь, можна оцінювати за будь-яким з критеріїв залежно від вихідних умов та задання предметної області.

Аналогічні обчислення необхідно провести і для інших відношень, які було досліджено.

3. Тришарова декомпозиція предикатів.

Форму запису предиката еквівалентності можна змінити за допомогою предикату рівності та характеристичних функцій. Предикат еквівалентності представимо в узагальненій формі через кон'юнкцію його характеристичних предикатів.

Тришаровою декомпозицією предиката E називається його подання в такому вигляді:

$$E(x_1, x_2, x_3) = D(g_1^{-1}(f_1(x_1)), g_2^{-1}(f_2(x_2)), g_3^{-1}(f_3(x_3))),$$

де $f_1, f_2, f_3, g_1, g_2, g_3$ – деякі функції.

Узагальнення теореми про загальний вигляд 2-го роду предиката на n -арні предикати має такий вигляд:

$$E(x_1, x_2, \dots, x_n) = \exists v \in B (F_1(x_1, v) \wedge F_2(x_2, v) \wedge \dots \wedge F_n(x_n, v)),$$

де

$$F_i(x_i, v) = \exists x_1 \in A_1 \exists x_2 \in A_2 \dots \exists x_{i-1} \in A_{i-1} \exists x_{i+1} \in A_{i+1} \dots \exists x_n \in A_n$$

$$S(x_1, x_2, \dots, x_n, v),$$

S – функція, яка присвоює будь-які різні імена v усім наборам (x_1, x_2, \dots, x_n) , для яких $E(x_1, x_2, \dots, x_n) = 1$; B – множина всіх таких імен.

Запропонована форма запису загального вигляду предиката 2-го роду за допомогою предиката рівності та відображення більш зручна для практики.

Представлений спосіб знаходження характеристичних предикатів використовує деяку функцію S , яка присвоює деякі різні імена v усім парам предметів, для яких відповідний предикат є істинним.

Таким чином, отримаємо

$$E(x_1, x_2, \dots, x_n) = \exists v \in B (F_1(x_1, v) \wedge F_2(x_2, v) \wedge \dots \wedge F_n(x_n, v)),$$

де

$$F_1(x_1, v) = \exists x_2 \in A_2 \exists x_3 \in A_3 \quad S(x_1, x_2, x_3, v),$$

$$F_2(x_2, v) = \exists x_1 \in A_1 \exists x_3 \in A_3 \quad S(x_1, x_2, x_3, v),$$

$$F_3(x_3, v) = \exists x_1 \in A_1 \exists x_2 \in A_2 \quad S(x_1, x_2, x_3, v).$$

Предикат $E(x_1, x_2, x_3)$, за визначенням, є

$$p_1^1 \vee p_1^3 = v_2^e v_3^s; p_1^4 = v_2^a v_3^i; p_1^6 = v_2^j v_3^s; p_1^5 \vee p_1^7 = v_2^b v_3^s; p_1^2 = v_2^d v_3^s;$$

$$p_2^1 \vee p_2^2 = v_1^u; p_2^3 \vee p_2^4 \vee p_2^5 = v_1^i; p_2^6 = v_1^a; p_2^7 = v_1^o.$$

$$p_1^8 \vee p_1^{11} \vee p_1^{16} = v_1^u v_3^s; p_1^{10} = v_1^i v_3^i; p_1^{12} = v_1^i v_3^o; p_1^9 \vee p_1^{13} = v_1^i v_3^s; p_1^{14} = v_1^i v_3^s; p_1^{15} = v_1^o v_3^s;$$

$$p_2^8 \vee p_2^9 = v_2^e; p_2^{10} = v_2^a; p_2^{11} \vee p_2^{12} \vee p_2^{13} = v_2^i; p_2^{14} = v_2^j; p_2^{15} = v_2^b; p_2^{16} = v_2^d.$$

$$p_1^{17} = v_1^u v_2^e; p_1^{19} = v_1^i v_2^a; p_1^{20} = v_1^i v_2^i; p_1^{18} \vee p_1^{21} = v_1^i v_2^i;$$

$$p_2^{17} \vee p_2^{18} = v_3^e; p_2^{19} = v_3^i; p_2^{20} = v_3^o; p_2^{21} = v_3^e.$$

Метод тришарової декомпозиції предикатів об'єднує двошарову декомпозицію предикатів 1-ого та 2-ого роду та дозволяє представити предикат у загальному вигляді і в більш компактному вигляді через предикат рівності.

Далі отримаємо опис предиката D_B , тобто третього шару схеми (див. рис. 1):

$$E(x_1, x_2, x_3) = D_B(g_1^{-1}(f_1(x_1)), g_2^{-1}(f_2(x_2)), g_3^{-1}(f_3(x_3))),$$

$$E(x_1, x_2, x_3) = D_B(p_1, p_2, p_3) = \bigvee_{\sigma} p_1^{\sigma} p_2^{\sigma} p_3^{\sigma} =$$

$$= p_1^1 p_2^1 p_3^1 \vee \dots \vee p_1^{21} p_2^{21} p_3^{21} = t.$$

$$r^i = p_1^i p_2^i p_3^i \quad (i = \overline{1, 21});$$

$$t = \bigvee_{i=1}^{21} r_i = r_1 \vee r_2 \vee \dots \vee r_{21}.$$

Побудована схема реалізує широке розпаралелювання обробки інформації. Схема працює в декількох режимах: обчислює значення предиката за заданим закінченням, визначає невідомі літери закінчення за відомими тощо.

Висновки. Розробку методу тришарової декомпозиції було здійснено шляхом узагальнення: взяли предикат еквівалентності, виключили властивість однозначності – отримали толерантність, виключили ще одну властивість рефлексивності, замінивши її квазірефлексивністю (рефлексивність не на всій області

композицією предикатів H_1, H_2, H_3 та D_C :

$$E(x_1, x_2, x_3) = \exists p_1, p_2, p_3 \in C (H_1(x_1, p_1) \wedge \wedge H_2(x_2, p_2) \wedge H_3(x_3, p_3) \wedge D_C(p_1, p_2, p_3)).$$

Далі за останніми отриманими при декомпозиції предикатів 2-го роду таблицями [9] необхідно отримати функції $g_1: R \rightarrow B_1$; $g_2: R \rightarrow B_2$; $g_3: R \rightarrow B_3$; ($R = \{1, 2, \dots, 21\}$). Наведемо отримані рівняння:

визначення, а на її підмножині) – отримали квазітолерантність. Потім виключили останню властивість – симетричність – і отримали довільний n -арний предикат.

Кожну вершину отриманої схеми можна схарактеризувати своєю змінною, яка визначена на відповідній множині. Насправді, кожна така вершина є не просто значенням змінної x , а унарним предикатом $P(x)$. Таким чином, кожна вершина містить деяку множину, і під час паралельної передачі інформації в інші вершини на виході знаходиться також множина значень. У результаті кожна гілка схеми є лінійним логічним перетворенням, що забезпечує її ефективну роботу.

Таким чином, метод декомпозиції предикатів дає узагальнений підхід до представлення довільних відношень. Побудовані на сьогодні реляційні мережі [5] орієнтовані на конкретну предметну область за рахунок уведення проміжних змінних. Незважаючи на те, що введення проміжних змінних забезпечує компактність та економність при паралельній обробці даних, реляційні мережі потребують окремої побудови для кожної предметної області. Метод декомпозиції предикатів є своєрідним узагальненням і дозволяє будувати схеми (хоча, як правило, надлишкові) для довільних відношень.

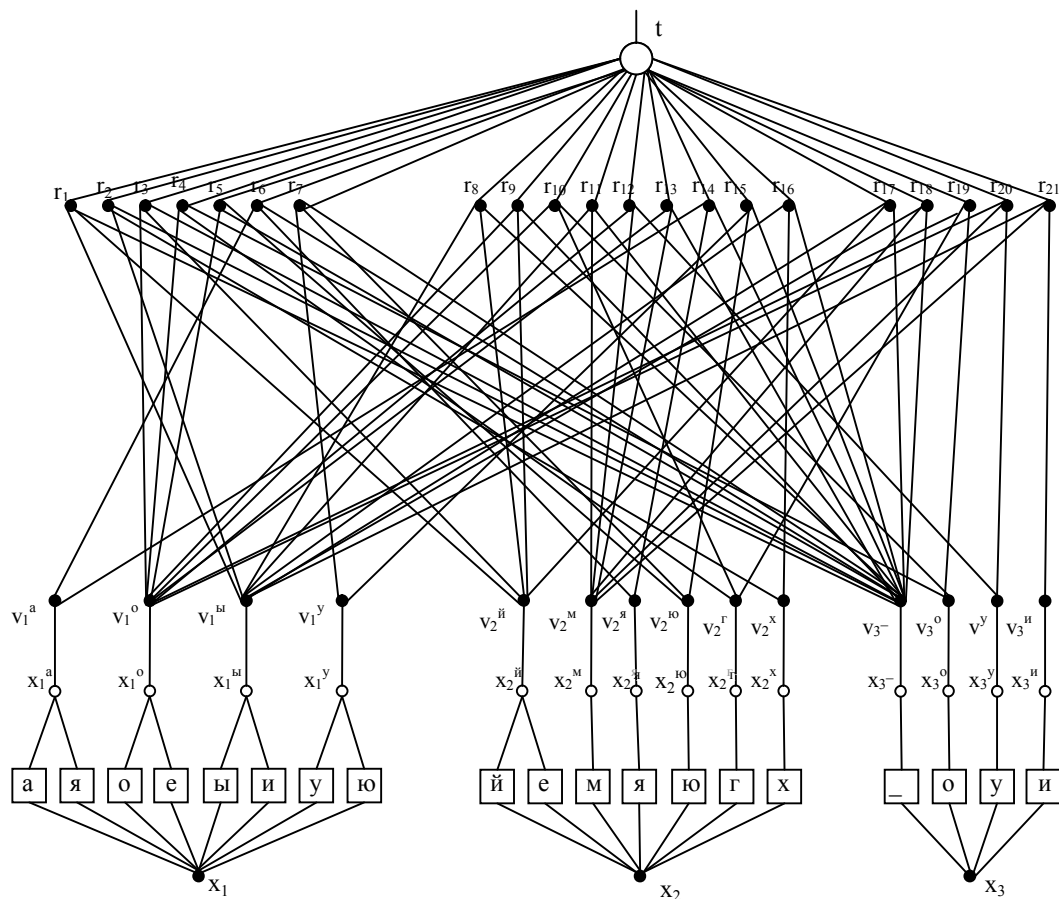


Рис. 1. Схема тришаровой декомпозиции предиката

Тому для подальших дослідженнях становить інтерес детальний розгляд взаємозв'язку між методом декомпозиції предикатів реляційними схемами предикатів для знаходження загального методу побудови реляційних мереж на основі методу тришарової декомпозиції предикатів. По суті, для цього необхідно

більш детально дослідити перехід від декомпозиції 1-го роду до декомпозиції 2-го роду предикатів, а саме, ґрунтуючись на методи побудови реляційних схем, сформувані нове правило побудови вузлів v_i .

Література

1. Шабанов-Кушнарєнко Ю. П. Приложения теории интеллекта к синтезу комбинационных схем / Ю. П. Шабанов-Кушнарєнко, М. Ф. Бондаренко, Г. Г. Четвериков, З. Ю. Шабанова-Кушнарєнко // АСУ и приборы автоматики. – 1980. – Вып. 53. – С. 10–18.
2. Широков В. А. Очерк основных принципов квантовой лингвистики / В. А. Широков // Бионика интеллекта. – 2007. – № 1 (66). – С. 25–32.
3. Бондаренко М. Ф. Основы теории багатозначных структур і кодування в системах штучного інтелекту / М. Ф. Бондаренко, З. Д. Коноплянко, Г. Г. Четвериков. – Х. : Фактор-друк, 2003. – 336 с.
4. Бондаренко М. Ф. О мозгоподобных ЭВМ / М. Ф. Бондаренко, З. В. Дударь, И. А. Ефимова, В. А. Лещинский, С. Ю. Шабанов-Кушнарєнко // Радиоэлектроника и информатика. – 2004. – № 4. – С. 83–99.
5. Бондаренко М. Ф. Модели языка / М. Ф. Бондаренко, В. А. Чикина, Ю. П. Шабанов-Кушнарєнко // Бионика интеллекта. – 2004. – № 61/1. – С. 27–37.
6. Ефимова И. А. Моделирование механизмов естественного языка с помощью бинарных логических сетей / И. А. Ефимова, В. А. Лещинский // Вестник НТУ «ХПИ»: сб. науч. тр. Тематич. вып.: Новые решения в современных технологиях. – 2005. – № 57. – С. 3–10.
7. Дударь З. В. Логическая сеть для модели глагольной флексии русского языка / З. В. Дударь, А. А. Иванюков, В. В. Климушев, В. И. Обризан // Восточно-европейский журнал передовых технологий. – 2006. – № 4/2. – С. 80–89.
8. Дударь, З. В. Предикаты эквивалентности в задачах компараторной идентификации / З. В. Дударь, С. А. Пославский, А. В. Пронюк, С. Ю. Шабанов-Кушнарєнко // Проблемы бионики. – 1999. – № 51. – С. 19–26.
9. Вечирская И. Д. Расслоение предикатов на примере словоизменения прилагательных русского языка / И. Д. Вечирская, Г. Г. Четвериков, Т. Н. Федорова // Искусственный интеллект. – 2009. – № 3. – С. 170–177.

Надійшла до редколегії 25.01.2010

Анотації

Досліджено метод тришарової декомпозиції предикатів, який поєднує декомпозицію на рівні предикатів та на рівні змінних. Наведено застосування методу на прикладі словозміни повних прикметників російської мови (для трьох змінних). Проведено аналіз методу розшарування предиката та методу побудови реляційних мереж як засобів формального подання довільних відношень.

Исследован метод трехслойной декомпозиции n -арных предикатов, объединяющий декомпозицию отношений на уровне предикатов и на уровне переменных. Представлено использование метода на примере словоизменения прилагательных русского языка (для трех переменных). Проведен анализ метода расчленения предиката и метода построения реляционных сетей как средств формального представления произвольных отношений.

The method of pair predicate three-layer decomposition which unites decomposition of relations on the level of predicates and variables is researched. The usage of this method on example of Russian adjectives' changes (for three variables) is presented. The method of predicate's exfoliation and the method of relational systems' building as means for formal presentation of random relations are analyzed.

УДК 004.4'413

Г. Ф. ДЮБКО,

кандидат технічних наук,

*професор кафедри програмного забезпечення електронних обчислювальних машин,
Харківського національного університету радіоелектроніки,*

В. В. ГОЛЯН,

кандидат технічних наук,

*доцент кафедри програмного забезпечення електронних обчислювальних машин
Харківського національного університету радіоелектроніки,*

Р. О. ГЛУЩЕНКО,

магістр

Харківського національного університету радіоелектроніки

СИНТАКСИЧНИЙ АНАЛІЗ ІЗ ВИКОРИСТАННЯМ АНОТОВАНИХ ГРАМАТИК

Синтаксичний аналіз, або розбір, як його ще називають, – це процес, у якому досліджується ланцюжок лексем і встановлюється, чи задовольняє він структурні умови, сформульовані у визначенні синтаксису мови. Розбір – це одна з найбільш важливих фаз компіляції. За сукупністю синтаксичних правил для деякої формальної мови можна автоматично побудувати синтаксичний аналізатор, який буде перевіряти, чи має вихідна програма синтаксичну структуру, обумовлену цими правилами.

В основі будь-якого синтаксичного аналізатора лежить алгоритм аналізу. Найбільш прості алгоритми використовують перебиральні стратегії [1], що призводить до набагато більших витрат апаратних ресурсів, зокрема часу. Проблема побудови ефективних синтаксичних аналізаторів актуальна донині, оскільки досить часто створюються нові мови програмування, мови розмітки (RDF, OWL) тощо, і для них необхідно створювати синтаксичні аналізатори з оптимальними тимчасовими характеристиками.

За допомогою певних теоретичних знань можна вручну створити такий аналізатор, але це завдання досить трудомістке й потребує багатьох обчислень, перевірок і інших операцій. Однак процес створення синтаксичного аналізатора можна автоматизувати, тим самим значно скоротивши витрати тимчасових ресурсів. **Мета** даної статті – проаналізувати існуючі методи й підходи до завдання автоматизації побудови синтаксичних аналізаторів і запропонувати альтернативне й ефективне рішення.

Огляд існуючих методів вирішення проблеми. Сьогодні розроблені генератори синтаксичних аналізаторів, що підтримують найпоширеніші методи граматичного розбору.

Одним із найбільш ранніх інструментів подібного типу стало поєднання YACC та LEX. Ці утиліти з'явилися на початку 1970-х рр. для системи UNIX. YACC розшифровується як Yet another compiler-compiler (ще один компілятор компіляторів). Він був розроблений С. Джонсоном і застосовувався для створення багатьох компіляторів.