КІБЕРНЕТИКА
та КОМП'ЮТЕРНІ
ТЕХНОЛОГІЇ

УДК 512.562:007.001.33 DOI:10.34229/2707-451X.21.1.4

І.І. ТКАЧОВ

НОВИЙ ПІДХІД ДО РОЗВ'ЯЗАННЯ ЗАДАЧІ ГЕНЕРУВАННЯ МНОЖИН СКЛАДНИХ СТРУКТУРНИХ ОБ'ЄКТІВ НА БАЗІ КВАЗІ-ЕКВІВАЛЕНТНОГО ПЕРЕТВОРЕННЯ СХЕМИ РОЗМІТКИ

Введено та досліджено перетворення схеми розмітки, яке отримало назву розфарбування розміток стовпчика схеми. Показано, що його виконання приводить до квазі-еквівалентної схеми розмітки, за розв'язком якої можна однозначно відтворити розв'язок початкової задачі. Запропонований метод застосування нововведеної операції для перетворення схеми розмітки в квазі-еквівалентну схему, в якій стає можливим регулярне виконання операції роз'єднання стовпчика. Ця властивість операції розфарбування розміток відкриває шлях до створення метода розв'язання задачі відтворення схеми розмітки, яка генерує задану множину узгоджених розміток.

Ключові слова: *реляційна схема, схема узгодженої розмітки, еквівалентні перетворення схеми розмітки, задача виконання обмежень.*

Робота присвячена теоретичному дослідженню з розробки методів побудови породжуючих конструкцій на базі схем розмітки для генерування та розпізнавання множин складних структурних об'єктів.

Цей напрямок досліджень входить до широкого розділу сучасної прикладної інформатики, який об'єднує методи розв'язання задачі виконання обмежень (Constraint Satisfaction Problem) і споріднені теми [1–9]. Між тим саме ця задача раніше не ставилася і регулярних методів її розв'язку досі не існує.

Аналіз вищевказаних методів ведеться на базі формалізму задачі узгодженої розмітки [6, 10, 11], яка з одного боку – це узагальнення багатьох постановок дискретних задач виконання обмежень, а з іншого боку – прозора теоретична конструкція з добре розробленим математичним підґрунтям.

Задачу побудови реляційної схеми (в даному випадку – схеми розмітки), що породжує задану множину відображень множини об'єктів у множину міток, за аналогією з лінгвістичними моделями можна називати задачею «відтворення граматики» [12–14].

В попередніх дослідженнях показано, що для розв'язання цієї задачі має сенс використати еквівалентні перетворення схем розмітки [11]. В даному випадку для розширення можливостей чотирьох відомих еквівалентних перетворень схеми розмітки – викреслення та додавання несуттєвої розмітки, а також з'єднання та роз'єднання стовпчиків – було додано нееквівалентне перетворення – «розфарбування розміток стовпчика».

Вводиться клас квазі-еквівалентних перетворень схем розмітки. Показано, що використання операції «розфарбування розміток», що відноситься до цього класу перетворень, дозволяє застосовувати операцію роз'єднання стовпчиків до будь-якого стовпчика схеми, що відкриває шлях до побудови алгоритму розв'язку задачі «відтворення граматики».

1. Розфарбування розміток стовпчика – квазі-еквівалентне перетворення схеми розмітки

Спочатку наведемо основні визначення, пов'язані з схемою узгодженої розмітки, які необхідні для викладення матеріалу даної роботи. Далі введемо операцію «розфарбування розміток стовпчика» та дослідимо її властивості. Ця операція – нееквівалентне перетворення схеми розмітки. На її основі буде визначений клас квазі-еквівалентних перетворень схеми розмітки.

Для подальшого розгляду нам знадобляться деякі визначення.

Визначення 1. Схемою розмітки називається четвірка $< U, L, W, R >$, де $U = \{u\}$ та $L = \{l\}$ – скінченні множини об'єктів та міток, відповідно, $W = \{w^{\sim}\}$ – множина неупорядкованих наборів об'єктів $w^{\sim}_i = (u_1, u_2, \ldots, u_{ni})$, R – множина наборів виду $(u_1, l_1, u_2, l_2, \ldots, u_n, l_n)$, де l_i – мітка об'єкта u_i, $i = 1, 2, \ldots, n$. Множину R називають ще множиною обмежень (пор. із «задачею виконання *обмежень*»).

Набір міток $l^{\sim} = (l_1, l_2, \ldots, l_n)$ з набору $(u_1, l_1, u_2, l_2, \ldots, u_n, l_n)$ будемо називати розміткою набору об'єктів $w^{\sim} = (u_1, u_2, \ldots, u_n)$ або, просто, розміткою, якщо буде зрозуміло, про який набір об'єктів йде мова. Схему розмітки представимо у вигляді таблиці.

$u_1, u_2, \ldots, u_n$	. . .	$v_1, v_2, \ldots, v_m$
$l^1_1, l^1_2, \ldots, l^1_n$	....	$k^1_1, k^1_2, \ldots, k^1_m$
. . .	. . .	. . .
$l^s_1, l^s_2, \ldots, l^s_n$		$k^r_1, k^r_2, \ldots, k^r_m$

Тут символами u_i та v_i позначені об'єкти, а l^s_i та k^r_i – мітки. Кожен стовпчик відповідає набору об'єктів, записаному в верхньому рядку. Усі інші рядки стовпчика – розмітки цього набору, що задаються наборами з множини R. У всіх змістовних задачах схема розмітки допускає таке представлення.

Визначення 2. Нехай задана схема розмітки $< U, L, W, R >$. Узгодженою відносно W та R розміткою будемо називати відображення (функцію) f: $U \rightarrow L$, яке ставить у відповідність кожному об'єкту з U мітку з L таким чином, що для будь-якого набору об'єктів $(u_1, u_2, \ldots, u_n) \in W$ у множині R знайдеться набір $(u_1, f(u_1), \ldots, u_n, f(u_n)) \in R$.

В подальшому узгоджені відносно W і R розмітки будемо називати просто узгодженими. Множину всіх узгоджених розміток схеми $S = < U, L, W, R >$ позначимо $\boldsymbol{L}(W, R)$.

Визначення 3. Під задачею схеми розмітки будемо розуміти пошук всіх узгоджених розміток $\boldsymbol{L}(W, R)$ за заданою схемою $S = < U, L, W, R >$.

Зазначимо, що на відміну від нашого розгляду, в якому суттєво важливо мати на увазі всю множину $\boldsymbol{L}(W, R)$, в деяких випадках для розв'язання прикладних задач достатньо визначити, чи є множина узгоджених розміток непорожньою. Відомо, що навіть при цьому спрощенні ця задача відноситься до класу *NP*-повних.

Вищевизначена схема розмітки – це *реляційна система*, оскільки кожен стовпчик фактично є відношенням (англ. relation) на множині об'єктів із значеннями з множини міток. Введення поняття узгодженої розмітки надає додаткового змісту такій реляційні системі, зокрема, воно дозволяє ввести поняття класу еквівалентних схем та розглядати еквівалентні (а також – нееквівалентні) перетворення схем розмітки.

Визначення 4. Схеми розмітки називаються еквівалентними, якщо вони визначають одну й ту ж саму множину узгоджених розміток.

В роботі [11] описана повна система еквівалентних перетворень схеми розмітки. Її складають чотири перетворення:

- викреслення (видалення) несуттєвої розмітки;
- дописування несуттєвої розмітки;
- з'єднання узгоджених стовпчиків та
- роз'єднання стовпчика.

Зазначимо, що в роботі [11], де вводилися ці перетворення, останні два були названі *склеювання стовпчиків* та *розшивання стовпчика*, відповідно. На наш погляд назви, що використовуються в даній роботі, є більш виправданими.

Те, що ця система перетворень є *повною* означає, що для будь-яких двох еквівалентних схем $S_1 = < U, L, W_1, R_1 >$ і $S_2 = < U, L, W_2, R_2 >$ існує послідовність з вказаних перетворень, що переводить одну схему в іншу.

Визначення 5. Під задачею відтворення схеми розмітки [12] будемо розуміти наступне:

нехай U – скінчена множина об'єктів, $|U| = N$;

L – скінчена множина міток;

задана множина кортежів K у вигляді N-арного відношення.

Потрібно побудувати схему $S = < U, L, W, R >$ так, щоб $\mathbf{L}(W, R) = K$.

За допомогою табличного представлення цю задачу можна відобразити наступним чином:

$U =$	$u_1, u_2, \ldots, u_N$
	K

→

$W =$	$w_1^{\sim}$	$w_2^{\sim}$	…..	$w_m^{\sim}$
	R_1	R_2		R_m

Тут $w_i^{\sim}$ та R_i – набір об'єктів і набір розміток i-го стовпчика цільової схеми, відповідно. Кількість стовпчиків m визначається як результат розв'язку задачі та не є цільовим параметром.

Зауважимо, що такий перехід не є однозначним. Нас цікавить у першу чергу таке перетворення, яке значно скорочує обсяг таблиці. В ідеальному випадку побудована таблиця схеми має бути мінімальною за розміром серед всіх еквівалентних схем.

Користуючись аналогією з області формальних граматик, можемо назвати множину K «мовою», цільову схему – «граматикою», що її породжує, а саму задачу – задачею автоматичного відтворення граматики за заданою мовою [15].

До цього часу розглядались еквівалентні перетворення схеми розмітки, які змінюють множини W та R, або тільки R, але не зачіпають множини об'єктів U та множини міток L, тобто не виводять перетворену схему за межі Σ_{UL} – множини всіх можливих еквівалентних схем розмітки на базі множин U та L. Водночас можна розглянути такі перетворення схем розмітки, які змінюють множини U та L, але при цьому залишаються еквівалентними в деякому узагальненому сенсі. При такому перетворенні отримана схема визначає множину узгоджених розміток, яка відрізняється від множини узгоджених розміток початкової схеми. Разом з тим за цією множиною за допомогою простої стандартної процедури може бути відновлена початкова множина узгоджених розміток.

Тепер розглянемо операцію «розфарбування розміток стовпчика схеми». Вона є змістом перетворення схеми розмітки, яке можна було би назвати «екстраполяцією схеми розмітки».

Нехай $S = < U, L, W, R >$ – початкова схема розмітки.

Виберемо один стовпчик в таблиці цієї схеми:

$u_1, u_2, \ldots, u_n$
$l^1{}_1,\ l^1{}_2, \ldots,\ l^1{}_n$ . . . . . . $l^p{}_1,\ l^p{}_2, \ldots,\ l^p{}_n$

Додаємо до набору об'єктів цього стовпчика ще один об'єкт v, який не належить до множини U. В результаті будуть отримані множини U_1 та W_1:

$$U_1 = U \cup \{v\},$$
$$W_1 = \{W \setminus (u_1, u_2, \ldots, u_n) \cup (u_1, u_2, \ldots, u_n, v)\}.$$

В цьому стовпчику до кожної розмітки дописуємо по одній деякій мітці, що відповідає нововведеному об'єкту. Це можуть бути, як мітки з множини L, так і якісь нові. У загальному випадку частина дописаних міток не належить L, і, таким чином, множина міток L розширюється до L_1. У відповідності з описаною операцією множина обмежень схеми R переходить у множину R_1. Стовпчик, розмітки якого були «розфарбовані» описаним вище чином, приймає наступний вигляд:

$u_1, u_2, \ldots, u_n, v$
$l^1{}_1,\ l^1{}_2, \ldots,\ l^1{}_n, k^1$ . . . . . . $l^p{}_1,\ l^p{}_2, \ldots,\ l^p{}_n, k^p$

а множина міток перетворюється таким чином:

$$L_1 = L \cup \{k_1, k_2, \ldots, k_p\}.$$

Позначимо множину обмежень, що визначається вибраним стовпчиком, R^*, а множину обмежень стовпчика, що отримується у результаті, – R^{**}. Тоді можемо записати множину обмежень отриманої схеми розмітки наступною формулою:

$$R_1 = R \setminus R^* \cup R^{**}.$$

Решта стовпчиків схеми S не змінилася. При наступних «розфарбуваннях» розміток у стовпчиках об'єкти, що додаються, кожен раз мають бути новими. Перед «розфарбуванням» розміток обраного стовпчика деякі розмітки в ньому можна продублювати (вочевидь це еквівалентне перетворення), а при розфарбуванні до кожної копії одної розмітки треба буде дописувати різні мітки. В такому випадку кількість не дубльованих розміток у стовпчику зростає.

Теорема 1. Нехай в результаті операції розфарбування розміток стовпчика схеми розмітки $S = < U, L, W, R >$ отримана схема $S_1 = < U_1, L_1, W_1, R_1 >$. Тоді:

1) для кожної узгодженої відносно W_1 та R_1 розмітки f_1: $U_1 \rightarrow L_1$, $f_1 \in \boldsymbol{L}(W_1, R_1)$ знайдеться узгоджена відносно W та R розмітка f: $U \rightarrow L$ така, що для всіх $u \in U$, буде виконано $f(u) = f_1(u)$;

2) для кожної узгодженої відносно W та R розмітки, f: $U \rightarrow L$, $f \in \boldsymbol{L}(W, R)$ знайдеться узгоджена відносно W_1 та R_1 розмітка f_1: $U_1 \rightarrow L_1$, $f_1 \in \boldsymbol{L}(W_1, R_1)$ така, що для всіх $u \in U$, буде виконано $f(u) = f_1(u)$.

Доведення 1. Нехай $f_1 \in \boldsymbol{L}(W_1, R_1)$. Необхідно показати, що для будь-якого набору $(u_1, u_2, \ldots, u_n) \in W$ схеми S у відповідному стовпчику знайдеться розмітка $(f_1(u_1), f_1(u_2), \ldots, f_1(u_n))$. Це положення, безумовно, має місце для всіх стовпчиків, крім того, до якого застосовувалась операція розфарбування, оскільки ці стовпчики в схемах S та S_1 співпадають.

Розглянемо стовпчик з набором об'єктів ($u_1, u_2, ... , u_n$), який був змінений у результаті розфарбування розміток. Оскільки f_1: $U_1 \to L_1$ – узгоджена відносно W_1, та R_1 розмітка, тоді в стовпчику схеми S_1 з набором об'єктів ($u_1, u_2, ... , u_n, v$) (це той стовпчик, що був отриманий у результаті операції розфарбування) знайдеться розмітка ($f_1(u_1), f_1(u_2), \dots , f_1(u_n), f_1(v)$) = ($l_1, l_2, ... , l_n, k$). За визначенням операції розфарбування розмітка ($l_1, l_2,..., l_n$) знаходилась у стовпчику ($u_1, u_2, ... , u_n$) до застосування операції розфарбування. Таким чином, встановлено, що й у цьому стовпчику знайдеться розмітка ($f_1(u_1), f_1(u_2), \dots , f_1(u_n)$). Отже, $f \in \boldsymbol{L}(W, R)$, де $f(u) = f_1(u)$ для всіх $u \in U$.

Перейдемо до п. 2) теореми. Нехай f: $U \to L$, $f \in \boldsymbol{L}(W, R)$. Виберемо в стовпчику схеми S, до якого була застосована операція розфарбування, розмітку, яка входить до узгодженої розмітки f, – ($l_1, l_2,..., l_n$) = ($f(u_1), f(u_2), \dots , f(u_n)$). При виконанні операції розфарбування до цієї розмітки була дописана деяка мітка k з множини міток, що додаються, – K, яка відповідає новому введеному об'єкту v. Це означає, що в схемі S_1 у стовпчику, що розглядається, є розмітка ($f(u_1), f(u_2),\dots, f(u_n), k$). З погляду того, що решта стовпчиків схем S та S_1 співпадають, можна заключити, що розмітка $f_1 : \{U \cup v\} \to \{L \cup K\}$, така що $f_1(u) = f(u)$ для всіх $u \in U$ та $f_1(v) = k$, є узгодженою відносно W_1 та R_1 розміткою – $f_1 \in \boldsymbol{L}(W_1, R_1)$. Теорема доведена.

Розглянемо більш складний випадок m-кратного застосування операції розфарбування.

Теорема 2. Нехай у результаті m-кратного застосування операції розфарбування до стовпчиків схеми $S = \, < U, L, W, R >$ отримана схема $S_1 = \, < U_1, L_1, W_1, R_1 >$. При цьому $U_1 = U \cup \{v_1, v_2, \dots , v_m\}$ та $L_1 = L \cup K_1 \cup K_2 \cup ... \cup K_m$, де $\{v_1, \dots , v_m\}$ та K_i , $i = 1,\dots, m$ – введені в схему об'єкти та мітки, відповідно. Тоді:

1) будь-яка узгоджена розмітка схеми S_1, якщо її обмежити на множину об'єктів U, визначить узгоджену розмітку схеми S:

$$f_1 \in \boldsymbol{L}(W_1, R_1) => f_1|_U \in \boldsymbol{L}(W, R);$$

2) будь-яка узгоджена розмітка схеми S, якщо її довизначити на множину доданих об'єктів $\{v_i\}^m_{i=1}$, за принципом:

$f_1|_U = f$ та $f_1(v_i) = k_i^*$, де $k_i^* \in K_i$ та в i-му стовпчику, в який доданий об'єкт v_i ця мітка приписана до розмітки ($f(u_1), f(u_2), \dots, f(u_n)$), тоді таке відображення $f_1 : U_1 \to L_1$ визначить узгоджену розмітку схеми S_1:

$$f \in \boldsymbol{L}(W, R) = > f_1 \in \boldsymbol{L}(W_1, R_1).$$

Доведення теореми 2 аналогічне вищенаведеному доведенню теореми 1.

В результаті виконання операцій розфарбування узгоджені розмітки початкової схеми $\{f : U \to L\} = \boldsymbol{L}(W, R)$ немов би екстраполюються за межі множини об'єктів U за рахунок їх довизначення на об'єктах, що вводяться, – $\{v_1, v_2, \dots , v_m\}$. При цьому якщо до виконання операції розфарбування деякі розмітки в стовпчику дублювалися (нагадаємо, що це еквівалентне перетворення), а до них дописувалися різні мітки, то після розфарбування буде отримана нова схема S_1, в якій кількість узгоджених відносно W_1 та R_1 розміток – $|\boldsymbol{L}(W_1, R_1)|$, може перевищувати кількість узгоджених відносно W та R розміток – $|\boldsymbol{L}(W, R)|$. Важливо, що і в цьому випадку обмеження будь-якої узгодженої розмітки $f_1 \in \boldsymbol{L}(W_1, R_1)$ на множину U не виводить за межі множини $\boldsymbol{L}(W, R)$: $f_1|_U \in \boldsymbol{L}(W, R)$. Ця властивість операції розфарбування розміток дозволяє нам по-новому поглянути на поняття еквівалентності у відношенні перетворень схеми розмітки. Дане перетворення не є еквівалентним, оскільки при його виконанні змінюються множини U та L. Разом з тим воно зберігає множину узгоджених розміток початкової схеми у вигляді множини *часткових* узгоджених розміток (за аналогією з *частковими* функціями), обмежених на початкову множину об'єктів U. Така властивість перетворення, що розглядається, дозволяє назвати його *квазі-еквівалентним*.

2. Розфарбування розміток стовпчика для розв’язання задачі «відновлення граматики»

Розглядатимемо тепер одне важливе застосування операції розфарбування розміток.

Раніше була введена операція роз’єднання (розшивки) стовпчика. Ця операція – еквівалентне перетворення схеми розмітки, парна до операції з’єднання стовпчиків. Її істотна відмінність від операції з’єднання стовпчиків полягає в тому, що якщо для з’єднання досить вибрати два довільних стовпчики схеми, то для виконання операції роз'єднання необхідно виконання певних вимог до структури обраного стовпчика. Саме, після розбиття набору об’єктів обраного стовпчика на три частини – A_1, C та A_2 – треба переконатися, що розмітки цього стовпчика утворюють блоки з певними властивостями (такі блоки розміток назвемо *комплектними*) [11]. Якщо не всі блоки, на які розбилися розмітки, є комплектними, то роз’єднати стовпчик з набором об’єктів A_1CA_2 на стовпчики з наборами A_1C та CA_2 неможливо. Разом з тим регулярне застосування операції роз’єднання стовпчиків представляється дуже корисним для створення методів розв’язання *задачі відтворення породжуючої схеми розмітки* [12–14]. Як згадувалось раніше, ця задача аналогічна *проблемі відтворення граматики* [15] із області теорії формальних граматик, коли треба розробити алгоритм, який за заданою мовою будує граматику, що її породжує.

Якщо при постановці задачі не будемо обмежуватися множиною еквівалентних схем розмітки Σ_{UL}, тобто не будемо вимагати побудови *еквівалентної* породжуючої схеми, а допустимо отримання як результата квазі-еквівалентної схеми, то для розв’язання задачі можемо застосувати квазі-еквівалентні перетворення, зокрема, операцію розфарбування розміток, поряд з уже відомими еквівалентними перетвореннями [11]. До породжуючої квазі-еквівалентної схеми, яка буде отримана за допомогою еквівалентних та квазі-еквівалентних перетворень, слід додати «фільтр», який буде відбирати тільки ті частини узгоджених розміток, які відповідають первинній множині об’єктів – U. Така конструкція буде точно відповідати постановці задачі, що розглядається. Далі покажемо, яким чином операція розфарбування розміток відкриває можливість регулярного застосування операції роз’єднання стовпчиків.

Виберемо у початковій схемі S_1 стовпчик для роз’єднання. Далі, з деяких зовнішніх щодо самої операції роз’єднання стовпчика міркувань виділяємо три частини набору об’єктів обраного стовпчика – A_1, C та A_2, де піднабір C відіграє роль сполучної частини, а піднабори A_1C та CA_2, утворять набори об’єктів двох стовпчиків, на які розіб’ється початковий стовпчик. Потім виконуємо поділ розміток стовпчика на блоки. З цього починається виконання операції роз’єднання стовпчика, яка детально описана в [11]. Нагадаємо, що *блок* утворюють всі розмітки стовпчика, які містять одну й ту саму підрозмітку, що відповідає піднабору C. Далі нам знадобиться

Визначення 6. Блок розміток стовпчика з набором об’єктів A_1CA_2 і підрозміткою $l^{\sim}$ піднабору C однаковою для всіх розміток блоку будемо називати *комплектним*, якщо він містить всі розмітки виду $k^{\sim}l^{\sim}m^{\sim}$, де $k^{\sim}$ – підрозмітка піднабору A_1 з деякої розмітки блоку і $m^{\sim}$ – підрозмітка піднабору A_2 з деякої іншої розмітки блоку.

Якщо всі блоки комплектні, то виконуємо роз’єднання стовпчика на два з наборами об’єктів A_1C та CA_2 за правилами операції роз’єднання. В іншому випадку в стовпчику є некомплектні блоки розміток. Намагаємось доповнити кожен некомплектний блок новими розмітками до комплектності. Знову маємо два випадки:

1) додані розмітки є *несуттєвими* (тобто протирічать всім розміткам деякого іншого стовпчика). Оскільки додавання несуттєвої розмітки є еквівалентним перетворенням, то отриманий стовпчик, знову-таки можна роз’єднати за відомими правилами;

2) серед доданих розміток є *суттєві*. (Розмітку вважаємо суттєвою, якщо в кожному стовпчику схеми знайдеться хоча б одна узгоджена з нею розмітка). Додавання таких розміток може і не вплинути на множину узгоджених розміток схеми, тобто привести до еквівалентної схеми, однак, не розв’язавши задачу схеми розмітки повністю, ми не зможемо знати цього наперед. Тому ми змушені допустити, що таке перетворення може виявитися нееквівалентним та призводити до появи нових узгоджених розміток. В такому випадку операція розфарбування дає один з шляхів виходу з цієї ситуації.

Далі опишемо метод застосування операції розфарбування для виконання роз'єднання стовпчика. У сполучну частину набору об'єктів обраного стовпчика C вводимо новий об'єкт – v. В некомплектному блоці розміток виділяємо підмножину розміток, які за відсутності решти розміток могли б скласти комплектний блок. Таку множину розміток можемо назвати *умовно комплектним блоком*. У всі ці розмітки додаємо мітку k_1 об'єкта v. Так сформується комплектний блок стовпчика, який утворюється. З решти розміток початкового стовпчика знову виділяємо умовно комплектний блок. Це завжди можна зробити, оскільки будь-яка поодинока розмітка є умовно комплектним блоком. В розмітки виділеного блока додаємо мітку k_2 об'єкта v, $k_2 \neq k_1$. Ці мітки можуть як належати до множини міток L схеми S_1, так і бути новими мітками, які розширюють множину міток. Важливо лише, щоб кожна мітка, що додається в ході операції, відрізнялася від тих, що були введені раніше. Виділяємо наступний комплектний блок міток (або – умовно комплектний блок), позначаємо його новою міткою та повторюємо цей процес, поки всі розмітки стовпчика не увійдуть у комплектні блоки.

В результаті проведення операції розфарбування стовпчика за вищевказаним методом отримуємо квазі-еквівалентну схему S_2, яка відрізняється від початкової схеми тільки одним перетвореним стовпчиком, в якому виконані всі умови для роз'єднання на два стовпчики з наборами об'єктів A_1Cv та CvA_2. В квазі-еквівалентній схемі S_2, виконуємо роз'єднання стовпчика, отримуючи еквівалентну їй схему S_3.

В теорії двомірних граматик можна знайти часткову аналогію такого роз'єднання з розфарбуванням. Це операція «введення проміжних вершин» [16], застосування якої до двомірних граматик в деяких випадках дозволяє значно зменшити обсяг пам'яті комп'ютера, необхідної для реалізації алгоритмів викреслення та «двомірного програмування».

Розглянемо простий приклад операції «роз'єднання стовпчика з розфарбуванням».

Нехай початковий стовпчик має такий вигляд:

u_1	u_2	u_3
1	3	5
1	3	4
2	3	5
2	4	6

Покладемо $A_1 = \{u_1\}$, $C = \{u_2\}$, $A_2 = \{u_3\}$. В даних умовах роз'єднати стовпчик на два з наборами об'єктів $\{u_1, u_2\}$ та $\{u_2, u_3\}$ неможливо, оскільки перший блок розміток, в якому об'єкт u_2 помічений міткою «3», не є комплектним.

Виконуємо операцію розфарбування: додаємо об'єкт v і, виділяючи комплектні підблоки, вводимо в них нові мітки:

u_1	u_2	v	u_3	
1	3	0	5	(1-й блок)
1	3	0	4	
2	3	9	5	(2-й блок)
2	4	8	6	(3-й блок)

Тепер до отриманого стовпчика можна застосувати еквівалентне перетворення – операцію роз'єднання стовпчика:

u_1	u_2	v
1	3	0
2	3	9
2	4	8

u_2	v	u_3
3	0	5
3	0	4
3	9	5
4	8	6

Використовуючи об’єкт v як сполучну частину в обох стовпчиках, маємо можливість роз’єднати їх ще раз.

Роз’єднуємо перший стовпчик:

u_1	v
1	0
2	9
2	8

v	u_2
0	3
9	3
8	4

Роз’єднуємо другий стовпчик:

u_2	v
3	0
3	9
4	8

v	u_3
0	5
0	4
9	5
8	6

Серед отриманих чотирьох стовпчиків один дублюється, тому залишаємо один екземпляр і як результат маємо таку схему:

u_l v	v u_2	v u_3
1 0	0 3	0 5
2 9	9 3	0 4
2 8	8 4	9 5
		8 6

Перетворення початкового стовпчика не зменшило обсяг даних, натомість зменшилась арність обмежень. Тепер кожен стовпчик – бінарне відношення.

Приклад закінчено.

Цінність викладеного методу полягає у тому, що відкривається можливість побудови алгоритму розв'язання задачі «відновлення граматики» [12–14] на базі еквівалентних перетворень схем розмітки. При цьому треба враховувати, що допускаючи застосування операції розфарбування, ми будемо отримувати квазі-еквівалентні схеми. В цьому випадку об'єкти, що вводяться, слід розглядати як допоміжні або, в певному сенсі, керуючі. У породжуваних узгоджених розмітках нас будуть цікавити тільки ті частини відображень, які відповідають початковій множині об'єктів. Перехід від принципу еквівалентності, коли конструкція, що породжується, безпосередньо генерує задану множину узгоджених розміток, до принципу квазі-еквівалентності, коли задані відображення породжуються у вигляді частин узгоджених розміток (іншими словами, часткових відображень), дозволяє подолати «генеративну обмеженість» одновимірних конструкцій типу персептрона або «одновимірних граматик» [17, 18].

В подальших дослідженнях планується розробити та детально розглянути алгоритми розв'язання задачі «відтворення граматики» в рамках викладеного підходу.

Висновки. Введено та досліджено перетворення схеми розмітки, яке отримало назву *розфарбування розміток стовпчика схеми*.

Показано, що його виконання приводить до квазі-еквівалентної схеми розмітки, яка в класичному сенсі не є еквівалентною початковій схемі, але за розв'язком задачі в цій новій схемі однозначно відтворюється розв'язок початкової задачі за допомогою тривіальної процедури.

Показано, що операцію розфарбування розміток стовпчика схеми можна узгодити із структурою розміток будь-якого стовпчика таким чином, що в результаті виконання цієї операції отриманий стовпчик буде відповідати умовам, що необхідні для виконання еквівалентного перетворення схеми розмітки – роз'єднання стовпчика. Це означає, що операцію роз'єднання стовпчиків можна регулярно виконувати, як квазі-еквівалентне перетворення у парі з розфарбуванням розміток. Нові можливості цієї операції відкривають шлях до створення метода розв'язання задачі «відновлення граматики», тобто побудови схеми розмітки, яка задає дану множину відображень, на базі еквівалентних перетворень схеми розмітки.

Список літератури

1. Dechter R. Constraint processing. San Francisco: Morgan Kaufmann, 2003. 481 p.
2. Freuder E.C., Mackworth A.K. Constraint satisfaction: An emerging paradigm. *Foundations of Artificial Intelligence*. 2006. V. 2. P. 13–27. https://doi.org/10.1016/S1574-6526(06)80006-4
3. Tsang E. Foundations of Constraint Satisfaction. New York: Academic Press, 1993. 421 p.
4. Handbook of Constraints Programming. Elsevier B.V. 2006. 955 p.
5. Ткачов І.І. Спеціальні структури в схемах розмітки. II Міжнар. конф. *«Роль інновацій в трансформації образу сучасної науки»*. Київ: Інститут інноваційної освіти. 2018. С. 213–222. https://novaosvita.com/wp-content/uploads/2019/01/InnTrImModSc-Kyiv-Dec2018_v1.1.pdf
6. Ткачев И.И. Реляционные методы распознавания образов и задача гомоморфизма алгебраических моделей. *Доклады АН Украинской ССР Сер. А*. 1988. № 1. С. 76–78.
7. Ткачов І.І., Реляційні моделі для опису структур складних об´єктів у розпізнаванні образів. Міжнар. конф. «Наука, освіта, суспільство: актуальні питання і перспективи розвитку». Одеса: ГО «ІОМП». 2016. С. 164–169. https://en.calameo.com/read/0031683720c8ede790c06
8. Ткачов І.І. Узгоджені розмітки та гомоморфізми відношень в задачах аналізу структур складних об'єктів. II Міжнар. конф. *«Наука, освіта, суспільство: актуальні питання і перспективи розвитку»*. Київ: Інститут інноваційної освіти. 2016. С. 187–192. https://ru.calameo.com/read/0031683726793adec3caa
9. Ткачев И.И. Реляционные методы распознавания: две фундаментальные постановки. *"Математические методы распознавания изображений"* тезисы докладов. Киев: Ин-т кибернетики АН Украины. 1991. С. 32–34.

10. Haralick R.M. Scene Matching Problems. *Digital image processing and analysis: NATO advanced study institute*. 1978. P. 184–202.
11. Ткачев И.И. Эквивалентные преобразования в одном классе распознающих систем. *Кибернетика.* 1981. № 1. С. 27–32. http://www.nucpi.nas.gov.ua/images/staff/Tkachev/Kibernetika1981.pdf
12. Ткачев И.И. Преобразования реляционных схем и задача «восстановления грамматики». Міжнар. конф. *«Сучасна інформатика: проблеми, досягнення та перспективи розвитку».* Київ: Інститут кібернетики імені В.М. Глушкова НАН України. 2017. С. 154–157. https://ru.calameo.com/read/0031683724703e37cfd96
13. Ткачов І.І. Генеративні можливості реляційних схем. V Міжнар. конф. *«Обчислювальний інтелект (Результати, проблеми, перспективи)».* Ужгород. 2019. С. 131–132. https://en.calameo.com/read/003168372f07dda801899
14. Ткачев И.И. О модели конструктивного порождения множеств сложных объектов на основе реляционных схем. V Міжнародна конференція *«Відкриті еволюціонуючі системи»*. К.: ФООП Маслаков. 2020. С. 203–205. https://ru.calameo.com/read/0031683726ecb6303d80a
15. Фу К. Структурные методы в распознавании образов. М.: Мир, 1977. 319 с.
16. Шлезингер М.И. Синтаксический анализ двумерных зрительных сигналов в условиях помех. *Кибернетика*. 1976. № 4. С. 113–130.
17. Шлезингер М.И. Математические средства обработки изображений. Киев: Наукова думка, 1989. 200 с.
18. Минский М., Пэйперт С. Персептроны. М.: 1971. 262 с.

Одержано 18.01.2021

Ткачов Ігор Іванович,
кандидат фізико-математичних наук, доцент,
учений секретар Науково-учбового центру прикладної інформатики НАН України.
https://orcid.org/0000-0002-1154-1131
tk_i@ukr.net

UDC 512.562:007.001.33

I. Tkachov

A New Approach to Solving the Problem of Generating Sets of Complex Structural Objects Based on a Quasi-Equivalent Transformation of a Labeling Scheme

Scientific and Training center for Applied Informatics of the NAS of Ukraine, Kyiv
Correspondence: tk_i@ukr.net

The paper presents the results of a theoretical study related to the development of methods for constructing generating structures based on labeling schemes for generating sets of complex structural objects. In a theoretical aspect, generated objects are mappings of sets of objects into a set of labels, and in practical terms, they can be, in particular, visual images. The scientific and practical interest in generative constructions is that they can be used to determine whether objects belong to a certain class, that is, to solve the problem of pattern recognition.

The problem of constructing generating labeling scheme belongs to a wide section of modern applied informatics that embraces Constraint Satisfaction Problem and related themes [1–4]. But this problem has not been posed before and there are still no regular methods for solving it.

The analysis of the above methods is based on the formalism of the consistent labeling problem [6, 10, 11], which is, on the one hand, a generalization of many statements of discrete problems of Constraint Satisfaction, and, on the other hand, a transparent theoretical construction with a well-developed mathematical foundation.

The problem of constructing a relational scheme (in this case, labeling scheme) that generates a given set of mappings, by analogy with linguistic models, may be named “the problem of grammar restoration” [12–14].

In previous studies it was shown that to solve this problem it makes sense to use equivalent transformations of the labeling scheme [11]. This is because the source table listing all the complex objects that should be generated by the target scheme is itself a trivial variant of the scheme with a given set of consistent labelings. This means that the source scheme and target scheme are equivalent. However, one of the equivalent operations – disunion of a column – cannot be used regularly, since it requires certain conditions to be met regarding the internal structure of the column.

In this case, to expand the capabilities of four known equivalent transformations of the labeling scheme – deleting and appending nonexistent labeling, as well as joining of columns and column disunion – a non-equivalent transformation was added – "coloring the column labelings".

The purpose of the paper is to introduce and investigate operation of "coloring the column labelings" that leads to a non-equivalent transformation of a labeling scheme. Show the advisability of using the known equivalent and the introduced quasi-equivalent transformations of the labeling scheme to solve the problem of constructing generating structures based on labeling schemes.

Results. The transformation of the labeling scheme, called "coloring the labelings of the scheme column", has been introduced. It is shown that its implementation leads to a quasi-equivalent labeling scheme, by solving which it is possible to uniquely restore the solution of the original problem. A method is proposed for using the newly introduced operation to transform the labeling scheme into a quasi-equivalent labeling scheme, in which it becomes possible to regularly perform the column decoupling operation. This ability of the operation of "coloring the column labelings" opens the way to the creation of a method for solving the problem of restoring a labeling scheme that generates a given set of consistent labelings.

Keywords: relational scheme, consistent labeling scheme, equivalent labeling scheme transformations, constraint satisfaction problem.