УДК 681.3.06

І.В. Редько, Н.М. Снігур

АЛГЕБРИЧНА ХАРАКТЕРИСТИКА КЛАСУ ГРАФОВИХ ПЕРЕТВОРЕНЬ НА МНОЖИНІ ЗВАЖЕНИХ ГРАФІВ

The paper under scrutiny considers a class of computable functions on a set of weighted graphs. Crucially, we find a generating set for the algebra of weighted graph functions and prove it is complete.

Вступ

Алгебричні методи дослідження застосовуються в інформатико-технологічній, в т.ч. програмістській діяльності, майже від самого виникнення комп'ютерної науки [1]. Основу цих методів становлять програмні алгебри, тобто алгебри, носіями яких є спеціальні класи функцій, а операціями — композиції, що являють собою абстракції засобів синтезу програм. Зокрема, в термінах цих алгебр точно ставляться та вирішуються проблеми повноти в різних класах обчислюваних функцій. Ці проблеми досі є актуальними в сучасному програмуванні.

Один із найважливіших напрямів досліджень в сучасному програмуванні — вивчення методів та створення ефективних засобів специфікації задач та їх розв'язків. Вагоме місце в цій проблематиці займають ієрархічні структури організації інформації та методи маніпулювання ними (див., наприклад, [3—11]). Тому вивчення відповідних класів маніпуляцій над ієрархічними структурами є актуальним.

Постановка задачі

Об'єктом дослідження є графові перетворювачі, орієнтовані на множину зважених графів. У статті подано точне визначення поняття графового перетворювача, проводиться прагматично обумовлена типізація сім'ї таких перетворювачів. Зокрема, в ній виділяється та досліджується прагматично важливий клас перетворювачів, які зберігають денотати. Предметом дослідження є алгебричні характеристики вибраних класів графових перетворювачів. Інструментом дослідження виступають так звані примітивні програмні алгебри (ППА) [2]. Доцільність саме такого вибору інструментарію обгрунтовано результатами, отриманими в [12—21].

Метою дослідження є пошук квазіскінченних породних множин і базисів для ППА над вибраними класами графових перетворювачів. Усі використані та невизначені в статті поняття та позначення наводяться за [15].

Базові поняття та визначення

Під (скінченним) графом g будемо розуміти пару $<V, E>$, де V — деяка скінченна непуста множина об'єктів (вершин графа), а $E \subset V \times V$ — множина його дуг. Не обмежуючи загальностей, покладемо $V \subset N$. При цьому на множині вершин вводиться цілком природна впорядкованість.

У деяких випадках ребрам графа можуть ставитись у відповідність різні дані — атрибути (мітки). Якщо як атрибути використовуються цілі чи дійсні числа, то такі графи називають *зваженими*. Фактично, зважений граф являє собою трійку $< V, E, c >$, де $c : E \to Z$ ($c : E \to$ $\to R$) — певна функція, визначена на множині дуг графа. Надалі обмежимось простим випадком, коли вага кожної дуги — натуральне число, тобто $c : E \to N$.

Вершини графа позначимо латинськими літерами u, v, w, а дуги — літерами e, p, s, можливо, з індексами $v_1, \ldots, e_1, \ldots$ За необхідності явно вказати вершини дуги та її напрямок, використовуватимемо запис $\langle v, u \rangle$, вважаючи, що дуга направлена від v до u. Вершину з найменшим номером називатимемо *першою* вершиною графа. Граф будемо позначати через $g^r_{|E|}$, де r — номер першої вершини графа g, а $|E|$ — кількість його дуг, або скорочено просто g. Щоб зазначити, що дуга e_i має вагу k_i, будемо використовувати запис (e_i, k_i).

Множину всіх графів позначимо G. Під функціями далі розуміємо часткові функції з аргументами та значеннями із G, а під предикатами — часткові предикати на G.

Обчислюваність на G вводиться як нумераційна обчислюваність (див. [1]) за допомогою арифметичної функції, що представляє цю функцію на G в деякій зафіксованій нумерації α_G множини G [19]. Існування такої нумерації випливає зі зліченості множини N.

Будь-яку частково-рекурсивну багатомісну функцію або будь-який частково-рекурсивний багатомісний предикат (чр-функція, чр-предикат) на G будемо називати *графовим перетворювачем,* або коротко — *перетворювачем,* коли це не призводить до протиріччя.

Через $A_G^{\text{чр}}$ позначимо ППА, носій якої становлять графові перетворювачі на G. *Породну множину* алгебри $A_G^{\text{чр}}$ назвемо її повною системою (ПС); ПС ППА — I_m^n*-базисом*, якщо будь-яка її підсистема, що отримується видаленням якого-небудь предиката або якоїнебудь функції, відмінної від селекторної, вже не буде повною.

Поняття функції, яка зберігає денотати, вводиться по аналогії з [1], проте, з певним узагальненням. Нехай $\beta : G \underset{\text{на}}{\rightarrow} 2^N$, де 2^N — множина всіх скінченних підмножин N.

Будемо казати, що функція f арності n β-зберігає денотати, якщо існує скінченна множина $V_f \subset N$ така, що для будь-якого $< x_1, \ldots, x_n > \in \mathrm{dom} f$ виконується

$$\beta(f(< x_1, \ldots, x_n >)) \subseteq \bigcup_{i=1}^{n} \beta(x_i) \cup V_f.$$

У випадку графів функція β є виділенням множини вершин графа, тобто для $g = \langle V, E \rangle$ будемо мати $\beta(g) = V$.

Через $A_{G,\beta}^{\text{чр}}$ позначимо ППА, носій якої становлять перетворювачі на G, які β-зберігають денотати. Визначення породної множини, повної системи та базису — такі самі, як і у випадку $A_G^{\text{чр}}$.

При знаходженні повних систем ППА є корисними певні необхідні умови повноти, які тут наведемо у вигляді тверджень.

***Твердження* 1.** Будь-яка повна система ППА для довільної множини L містить хоча б одну функцію, яка не зберігає L.

***Твердження* 2.** Будь-яка повна система ППА класу функцій, які не зберігають денотати, містить хоча б одну функцію, яка не зберігає денотати.

Нарешті, як сигнатуру ППА візьмемо сукупність $\Omega = \{\circ, \Diamond, *^{(n+1)}\}$, де $\circ$ — операція суперпозиції, $\Diamond$ — операція розгалуження, а $*^{(n+1)}$ — так зване $(n+1)$-арне циклування (визначення див. у [15]).

ППА чр-функцій і чр-предикатів на множині скінченних зважених графів

Розглянемо $g = \langle V = \{v_1, \ldots, v_n\},\ E = \{(\langle v_{i_1}, v_{i_2} \rangle, k_1), (\langle v_{i_3}, v_{i_4} \rangle, k_2), \ldots\}, 1 \le i_j \le n,\ j = 1, 2, \ldots$, де $k_1, k_2, \ldots$ — ваги дуг графа. Для подальшого викладу отриманих результатів буде доречним домовитись про певну впорядкованість дуг графа, тобто про впорядкованість пар натуральних чисел[1]. Зауважимо, що порядок на множині N^2 можна вводити по-різному[2], проте, в цьому випадку зручнішим здається такий:

$$\langle 0,0 \rangle < \langle 0,1 \rangle < \langle 0,2 \rangle < \ldots < \langle 0,n \rangle < \ldots <$$
$$< \langle 1,0 \rangle < \langle 1,1 \rangle < \ldots < \langle 2,0 \rangle < \langle 2,1 \rangle < \ldots \qquad (1)$$

Тоді дуги графа пронумеруємо відповідно до порядку (1), тобто першою дугою буде та, що відповідає найменшій парі в цьому порядку (найлівішій), а найбільший номер матиме дуга, що відповідає найбільшій парі (найправішій).

Часто в літературі з теорії графів розглядаються такі природні функції на множині графів як об'єднання та різниця графів, вилучення дуги графа, вилучення його вершини з усіма інцидентними їй дугами. У цій статті розглянемо дещо модифіковані графові перетворювачі.

Об'єднання графів $\mathbin{\overset{\frown}{\cup}}$ визначимо так. Нехай $g_1 = \langle V_1, E_1, c_1 \rangle$, $g_2 = \langle V_2, E_2, c_2 \rangle$, $g_1 \mathbin{\overset{\frown}{\cup}} g_2 = \langle V_1 \cup V_2,\ E_1 \cup E_2,\ c \rangle$, де $c(e) = c_1(e)$, якщо $e \in E_1, e \notin E_2$, $c(e) = c_2(e)$, якщо $e \notin E_1$, $e \in E_2$ та $c(e) = c_1(e) + c_2(e)$, якщо $e \in E_1, e \in E_2$. Різницею графів $\setminus$ вважатимемо наведену далі функцію. Для $g_1 = \langle V_1, E_1, c_1 \rangle$ та $g_2 = \langle V_2, E_2, c_2 \rangle$, позначимо $g_1 \setminus g_2 = \langle V, E, c \rangle$, де $E = \{e$ такі, що $e \in E_1$, $e \notin E_2$, $c(e) = c_1(e)$ або $e \in E_1$, $e \in E_2$, $c(e) = c_1(e) - c_2(e) > 0\}$.

---

[1] Такий крок є цілком природним, оскільки множина N^2, як відомо, є ефективно зліченою.

[2] Наприклад, як у канторівській нумерації (див. [1, с. 60]).

Замість функцій вилучення визначимо функції *виділення "першої" дуги* E_e так, що для $g = \langle\{v_1,\ldots\},\{(\langle v_1, v_{i_1}\rangle, k_1),\ldots\}\rangle$, $E_e(g) = (\langle v_1, v_{i_1}\rangle, k_1)$, та *виділення "першої" вершини* E_v: для $g = \langle V = \{v_1,\ldots\}, E\rangle$, $E_v(g) = \langle\{v_1\}, \varnothing\rangle$. (Функції вилучення отримуємо за допомогою щойно визначених функцій виділення та різниці графів).

Також розглянемо такі графові перетворювачі: D_v — *видалення першої вершини*, якщо вона ізольована; константна функція $C_0^{\mathsf{G}}(g) = g_0^1$; *видалення всіх дуг графа* $\overset{\circ}{D}e(\pi) = (E_e(\pi) \neq \Delta) \underset{\pi}{*} \langle D_e(\pi)\rangle$; функція *генерації порожнього графа* $C_{\Delta_{\mathsf{G}}}(g) = \Delta_{\mathsf{G}}$; S_{G} — *збільшення на одиницю номера першої вершини*, а саме для $g = \langle V = \{v_1, v_2,\ldots\}, E\rangle$, $S_{\mathsf{G}}(g) = < V, E_1 >$, де $E_1 = E \setminus \{< v_1, v_j > | < v_1, v_j > \in E\} \cup E_2$ та $E_2 \equiv \{< v_1 + 1, v_j > | < v_1, v_j > \in E\}$; D_e — *зменшення на одиницю ваги "першої" дуги* графа $g = \langle V, E\rangle$, причому якщо вага дорівнює одиниці, то дуга видаляється повністю. Формально, нехай $g = \langle\{v_1,\ldots\}, \{(\langle v_1, v_{i_1}\rangle, k_1),\ldots\}\rangle$, тоді $D_e(g) = \langle\{v_1,\ldots\},\{(\langle v_1, v_{i_1}\rangle, k_1 - 1),\ldots\}\rangle$, якщо $k_1 > 1$ та $D_e(g) = \langle\{v_1, \ldots\}, \{(e_2, k_2),\ldots\}\rangle$, якщо $k_1 = 1$.

Нарешті знадобиться ще часткова функція L, яка діє на множині із двох нульових графів $\{m\}$ та $\{n\}$ таким чином, що $L(\{m\},\{n\}) = \langle\{m, n\}, \{\langle m, n\rangle\}\rangle$ — *створення дуги*.

Можна безпосередньо перевірити, що всі ці функції є частково-рекурсивними.

Покладемо, що

$$\sigma_{\mathsf{G}} \coloneqq \{C_0^{\mathsf{G}}, S_G, \overset{\frown}{\cup}, \setminus, E_e, E_v, D_v, L, =_{\mathsf{G}}, I_m^n\}_{m=\overline{1,n}}^{n=1,2,\ldots}$$

та

$$\sigma_{\mathsf{G},\beta} \coloneqq \{C_0^{\mathsf{G}}, \overset{\frown}{\cup}, \setminus, E_e, E_v, D_v, L, =_{\mathsf{G}}, I_m^n\}_{m=\overline{1,n}}^{n=1,2,\ldots}.$$

Увесь подальший виклад присвячено доведенню того факту, що сукупність σ_{G} ($\sigma_{\mathsf{G},\beta}$) є базисом ППА $A_G^{\text{чр}}$ ($A_{G,\beta}^{\text{чр}}$).

Із означення поняття графа, наведеного вище, випливає таке його (ефективне) подання у вигляді вектора. Покладемо вектор $A_g = \Big(\underbrace{v_{i_1} v_{i_2} v_{i_1} v_{i_2} \ldots v_{i_1} v_{i_2}}_{k_1} \; \underbrace{v_{i_3} v_{i_4} \ldots v_{i_3} v_{i_4}}_{k_2} \ldots 0 v_{l_1} 0 v_{l_2} \ldots\Big)$, де v_{l_j}, $j = 1, 2, \ldots$ — ізольовані вершини (якщо вони є). Тобто кожну дугу $e_i = \langle v_{i_j}, v_{i_m}\rangle$ перерахуємо в цьому векторі таку кількість раз, якою є її вага k_i.

Очевидно, в такому поданні порожньому графу відповідає порожній вектор Λ, а повністю незв'язному графу, тобто графу $g \equiv < V, E >$, де $E = \varnothing$, відповідає вектор $(0v_1 0v_2 0 \ldots 0v_n)$. Задану таким чином відповідність можна навести у вигляді відображення $\varphi : \mathsf{G} \to \mathsf{N}^*$, де $\mathsf{N}^* = \bigcup_{i=0}^{\infty} \mathsf{N}^i$, $\mathsf{N}^0 = \{\Lambda\}$, яке вочевидь є ін'єкцією (проте, не є сюр'єкцією). Також зрозуміло, що множина $\mathsf{V} := \varphi(\mathsf{G})$ є рекурсивною в нумерації α_{G}.

***Означення* 1.** V-функцію $F(x_1,\ldots,x_n)$ назвемо *векторним образом* графового перетворювача F, якщо $F(\varphi(g_1), \ldots, \varphi(g_n)) \cong \varphi\mathsf{F}(g_1, \ldots, g_n)$ для усіх $g_1,\ldots,g_n \subset \mathrm{dom}\,\mathsf{F}$. Аналогічно V-предикат $P(x_1,\ldots,x_n)$ назвемо векторним образом граф-предиката $\mathsf{P}(\xi_1,\ldots,\xi_n)$, якщо $P(\varphi(g_1),\ldots, \varphi(g_n)) \cong \varphi\mathsf{P}(g_1,\ldots,g_n)$ для усіх $g_1,\ldots,g_n \subset \mathrm{dom}\,\mathsf{F}$.

Майже очевидною є лема.

Лема 1. Векторний образ чр-графового перетворювача (чр-графового предиката) є чр-V-функція (чр-V-предикат).

Будь-яка чр-V-функція є чр-N^*-функцією (тобто просто векторною чр-функцією). Для чр-V-предикатів аналогічно.

Векторний образ чр-графового перетворювача (чр-граф-предиката) є векторною чр-функцією (векторним чр-предикатом).

З метою моделювання векторних функцій графовими перетворювачами задамо також таке відображення $\Phi : \mathsf{N}^* \to \mathsf{G}$ так: будь-якому вектору $A = (v_1 v_2 \ldots v_n)$ поставимо у відповідність граф $g = \langle V = \{v_1, v_2, \ldots, v_n\}, E = \{\langle v_1, v_2\rangle, \langle v_2, v_3\rangle, \ldots, \langle v_{n-1}, v_n\rangle\}\rangle$, кожна дуга якого має вагу 1.

***Означення* 2.** Графовий перетворювач $\mathsf{F}(\xi_1,\ldots,\xi_n)$ називається *граф-моделлю* векторного перетворювача $F(x_1,\ldots,x_n)$, якщо $\mathsf{F}(\Phi(v_1),\ldots, \Phi(v_n)) \cong \Phi(F(v_1, \ldots, v_n))$ для всіх $v_1, \ldots, v_n \subset$

$\subset \mathrm{dom}\, F$. Граф-модель предиката вводиться аналогічно.

Лема 2. Для будь-яких векторних чр-функцій та чр-предикатів існують їх граф-моделі, які належать замиканню $[\sigma_{\mathrm{G}}]_{\Omega}$ $([\sigma_{\mathrm{G},\beta}]_{\Omega})$.

Нехай $\psi := \varphi\Phi$. Очевидно, що $\psi : \mathrm{G} \to \to \Phi(\mathrm{V})$ – бієкція. Через χ позначимо яке-небудь розширення відображення ψ^{-1}. Графові перетворювачі ψ та χ можна розглядати як *функцію-кодер* та *функцію-декодер* відповідно.

Безпосередньо із наведених вище означень випливає лема.

Лема 3. Нехай $\mathrm{F}(\xi_1,\ldots,\xi_n)$ – чр-граф-функція, а $\mathrm{H}(\pi_1,\ldots,\pi_n)$ – граф-модель векторного образу функції $\mathrm{F}(\xi_1,\ldots,\xi_n)$. Тоді матимемо

$$\mathrm{F}(A_1,\ldots,A_n) = \chi(\mathrm{H}(\psi(A_1),\ldots,\psi(A_n)))$$

для усіх A_i, $i = \overline{1,n}$.

Аналогічно, нехай $\mathrm{P}(\xi_1,\ldots,\xi_m)$ – чр-граф-предикат, а $\mathrm{K}(\pi_1,\ldots,\pi_m)$ – граф-модель векторного образу цього предиката. Тоді отримаємо

$$\mathrm{P}(A_1,\ldots,A_n) = \mathrm{K}(\psi(A_1),\ldots,\psi(A_n))$$

для усіх A_i, $i = \overline{1,n}$.

Лема 4. Має місце $\psi,\chi \in [\sigma_{\mathrm{G}}]$ $(\psi,\chi \in [\sigma_{\mathrm{G},\beta}])$.

Доведення.

1. ψ. Покладемо, що

$$\mathrm{G}_1(\pi) = (E_e(\xi) \neq \Delta) \underset{\pi}{*} \langle D_e(\pi)\rangle$$

та

$$\mathrm{G}_2(\pi,\xi) = (E_e(\xi) \neq \Delta) \underset{\pi,\xi}{*} \langle \pi \cup E_e(\xi) \cup$$
$$\cup L(D_v(G_1(E_e(\xi))), E_v(E_e(G_1(\xi)))), G_1(\xi)\rangle.$$

Тоді $\mathrm{G}_2(\Delta, g)$ являє собою граф-код за винятком ізольованих вершин. А отже, повний граф-код

$$\hat{g} = \psi(g) = G_2(\Delta, g) \cup \mathring{D}e(g).$$

2. χ. Розглянемо функцію

$$\mathrm{G}_3(\pi,\xi) = (E_e(\xi) \neq \Delta) \underset{\pi,\xi}{*} \langle \pi \cup E_e(\xi), G_1(G_1(\xi))\rangle.$$

Тоді

$$g = \chi(\hat{g}) = G_3(\Delta, \hat{g}) \cup \mathring{D}e(\hat{g}).$$

Що і треба було довести.

Теорема 1. σ_{G} $(\sigma_{\mathrm{G},\beta})$ є базисом ППА $A_G^{\text{чр}}$ $(A_{G,\beta}^{\text{чр}})$.

Доведення. Той факт, що σ_{G} $(\sigma_{\mathrm{G},\beta})$ є породною множиною $A_G^{\text{чр}}$ $(A_{G,\beta}^{\text{чр}})$, випливає безпосередньо із отриманих вище результатів.

Для остаточного доведення теореми досить показати, що видалити жодну з функцій σ_{G} $(\sigma_{\mathrm{G},\beta})$ неможливо. Для цього використаємо твердження 1 та 2:

1) $=_{\mathrm{G}}$ є єдиним предикатом множини;

2) лише функція C_0^G не зберігає множину графів $G \backslash \{g = \langle V, E\rangle \mid V = \{v_1,\ldots\}, v_1 = 1\}$ (множина графів, у яких номер першої вершини більший 1);

3) лише функція $\hat{\cup}$ не зберігає множину $\{g = \langle V, E\rangle \mid |V| \le 2\}$ (графи з кількістю вершин, меншою або рівною 2);

4) лише функція $\backslash$ не зберігає множину $G \backslash \{g = \langle V, \varnothing\rangle \mid |V| \ge 2\}$ (усі графи, крім тих, які повністю незв'язні та мають більше однієї вершини);

5) лише E_e не зберігає $G \backslash (\{g = \langle\{1,2\},\{\langle 1,2\rangle\}\rangle\} \cup \cup \{g = \langle\{v_1,\ldots\}, E\rangle \mid v_1 = 2\} \cup \{g = \langle\{1,2\},\varnothing\rangle\})$ (усі графи, крім таких: графа, що складається із дуги $\langle 1,2\rangle$; незв'язного графа, що складається із двох вершин з номерами 1 та 2; множини графів, для яких номер першої вершини дорівнює 2);

6) лише E_v не зберігає $G \backslash (\{g = \langle\{2\},\varnothing\rangle\} \cup \cup \{g = \langle\{1,2\},\varnothing\rangle\} \cup \{g = \langle V, \{e_1,\ldots\}\rangle \mid e_1 = \langle 1,2\rangle\})$ (усі графи, крім таких: графа з однієї вершини з номером 2; графа із двох вершин з номерами 1 та 2; множини графів, у яких першим ребром є $\langle 1,2\rangle$);

7) лише D_v не зберігає $\{g = \langle\{v_1,\ldots\}, E\rangle \mid v_1 = = 1\} \cup \Delta_G$ (множина графів, для яких номер першої вершини дорівнює 1, та порожній граф);

8) нарешті, лише L не зберігає множину $\{g = \langle V, \varnothing\rangle\} \cup \Delta_G$ (множина усіх повністю незв'язних графів і порожній граф).

Також у випадку $A_G^{\text{чр}}$ лише функція $S_{\mathrm{G}} \in \sigma_{\mathrm{G}}$ не зберігає денотати.

Висновок

Таким чином, доведено, що сукупність σ_G ($\sigma_{G,\beta}$) є не тільки породною множиною, а й базисом ППА $A_G^{\text{чр}}$ ($A_{G,\beta}^{\text{чр}}$). Лише функція збільшення на одиницю номера першої вершини $S_G \in \sigma_G$ не зберігає денотати у випадку $A_G^{\text{чр}}$.

Проведене дослідження репрезентативних класів графових перетворювачів, отримання відповідних алгебричних характеристик та визначення базисів у рамках ППА є важливим кроком у напрямі вивчення методів та створення ефективних засобів специфікації задач та їх розв'язків. Зокрема, отримані результати дають можливість як поглянути з єдиних позицій на наявні нині методи та засоби специфікації задач (UML, BPWin, ERWin і т.д.), так і, що найважливіше, створюють адекватне підгрунтя для пошуку нових рішень у напрямі дослідження складних структур організації інформації та методів маніпулювання ними. Окрему увагу було приділено пошуку природних I_m^n-базисів для зазначених вище предметних областей, що, безперечно, робить отримані результати прозорішими з точки зору їх застосування для реальних задач. Наступні роботи будуть присвячені практичному застосуванню отриманих результатів.

1. *Басараб И.А., Редько В.Н.* Базы данных с логико-функциональной точки зрения // Программирование. – 1984. – № 2. – С. 53–67.
2. *Мальцев А.И.* Алгоритмы и рекурсивные функции. – М.: Наука, 1965. – 392 с.
3. *Колмогоров А.Н., Успенский В.А.* К определению алгоритма // Усп. мат. наук. – 1958. – **13**, № 4. – С. 3–28.
4. *Asser G.* Berechenbare Graphenabbildungen // Kompliziertheit von Lern-und Erkennungsprozessen. – Jena: Friedrich-Schiller-Universitat, 1975. – P. 7–17.
5. *Заславский И.Д.* Граф-схемы с памятью // Тр. мат. ин-та АН СССР. – 1964. – **72**. – С. 99–192.
6. *Babai L., Grigoryev D., Mount D.* Isomorphism of Graphs with Bounded Eigenvalue Multiplicity // Proc. 14th ACM Symp. on Theory of Comput., STOC. – 1982. – P. 310–324.
7. *Пролубников А.В.* Прямой алгоритм проверки изоморфизма графов // Компьютерная оптика: Сб. научн. тр. / Под ред. акад. РАН Ю.И. Журавлева. – Изд-во Самар. гос. ун-та, 2007. – Вып. 27. – С. 123–128.
8. *Foggia P.A., Sansone C., Vento M.* A Database of Graphs for Isomorphism and Sub-Graph Isomorphism Benchmarking // Proc. of the 3rd IAPR TC-15 Intern. WorkShop on Graph-Based Representations, Italy, 2001. – P. 157–168.
9. *Spence E.* The Stromgly Regular (40, 12, 2, 4) Graphs // The Electronical Journal Of Combinatorics. – **7(1)**, 2000. – P. 1–4.
10. *Ершов А.П.* Вычислимость в произвольных областях и базисах // Семиотика и информатика. – Вып. 19. – М.: Изд. ВИНИТИ, 1979. – С. 3–58.
11. *Агафонов В.Н.* Спецификация программ: понятийные средства и их организация. – Новосибирск: Наука, 1987. – 240 с.
12. *Буй Д.Б., Редько В.Н.* Примитивные программные алгебры целочисленных и словарных функций // Докл. АН УССР. Сер. А. – 1984. – № 10. – С. 69–71.
13. *Буй Д.Б., Мавлянов А.В.* К теории программных алгебр // Укр. мат. журн. – 1984. – **36**, № 6. – С. 761–764.
14. *Буй Д.Б.* Примитивные программные алгебры: Автореф. дис. ... канд. физ.-мат. наук. – К., 1985. – 22 с.
15. *Буй Д.Б., Редько В.Н.* Примитивные программные алгебры. I, II // Кибернетика. – 1984. – № 5. – С. 1–7; – 1985. – № 1. – С. 28–33.
16. *Снигур Н.Н.* Алгебраическая характеристика класса частично-рекурсивных графовых функций // Реєстрація, зберігання і обробка даних. – К., 2009. – С. 25–32.
17. *Буй Д.Б., Поляков С.А.* Композиційна семантика SQL-подібних мов: табличні структури даних, композиції, приклади // Вісник Київ. ун-ту. Сер. Фіз.-мат. науки. – 1999. – Вип. 1. – С. 130–140.
18. *Ершов Ю.Л.* Теория нумераций. – М.: Наука, 1977. – 416 с.
19. *Голунков Ю.В.* О полноте операций в системах алгоритмических алгебр // Алгоритмы и автоматы. – Казань: Изд-во Казан. ун-та, 1978. – С. 11–53.
20. *Новиков Ф.А.* Дискретная математика для программистов. – СПб.: Питер, 2000. – 304 с.
21. *Мальцев А.И.* Конструктивные алгебры. I // Усп. мат. наук. – 1961. – **16**, № 3. – С. 3–60.

Рекомендована Радою
факультету електроніки НТУУ "КПІ"

Надійшла до редакції
18 березня 2011 року