УДК 519.683.8

Т.Л. Захарченко[1], Д.І. Редько[2], І.В. Редько[1], П.О. Яганов[1]

[1]Національний технічний університет України “КПІ”, Київ, Україна
[2]Київський національний університет імені Тараса Шевченка, Київ, Україна

ПРИМІТИВНА ПРОГРАМНА АЛГЕБРА ОБЧИСЛЮВАНИХ ФУНКЦІЙ НАД ЗАПИСАМИ

Background. The research is conducted in the context of compositional approach to programming. Problematic of the research is development of scientific foundations of programmer’s problems solution genesis. Its basis is concept of composition.
Objective. The objective of the research is general method development for function classes’ algebraic characteristics obtaining and application of the method for description of pragmatically important class of partially recursive functions on records.
Methods. Creations made in the paper are based on software analysis algebraic methods and compositional programming methodic. Problems of computable functions’ characteristics obtaining, problems of generative sets and bases finding, which are one of the most important questions in programmer’s problematic, are strictly stated and solved in the context of so called “program algebras”.
Results. In the paper method of mentioned problems solution was proposed in context of primitive program algebras (PPA) on different classes of computable functions. Received results are stated as sequence of original statements, lemmas, and theorems. They can be used for different classes of computable functions algebraic characteristics exploration in problems of programming languages semantics formalization.
Conclusions. Received results are foundations of adaptive programming environments development. Next steps in this direction will be connected with exploration of general concept of composition and development of related reduction methods of function exploration as environments of pragmatic depended programmer’s problems decomposition.

Keywords: fullness of calculable functions; fullness problem in PPA; complete system; pairs of natural numbers; pr-functions; pr-predicates.

Вступ

Сьогодні стан справ у інформаційно-технологічній галузі, в т.ч. й у програмуванні, істотно визначається процесом все більш глибокого та разом із тим широкого проникнення її у всі сфери життя. Звісно, що з кожним кроком у цьому напрямі за необхідності посилюються вимоги, що ставляться як до якості програмного продукту, так і до ефективності його виробництва. І незважаючи на те що програмістська діяльність (ПД) сьогодні продукує вражаючі результати, які “говорять самі за себе”, стає все більш очевидним, що ці результати переважно мають явно виражений екстенсивний характер і підтримання відзначеної тенденції в цих реаліях стає все більш проблематичним, а в перспективі неможливим. Причина цього — характерна для сьогоднішнього розуміння ПД, у т.ч. і програмування, його надмірна спрощеність, яка не відповідає рівню складності відзначеної проблематики [1–7].

Щодо програмування, то спрощеність його розуміння, яка проходить по лінії “значиме–другорядне”, полягає насамперед у тому, що увага акцентується переважно на результатах програмування, виводячи за рамки розгляду процеси їх отримання. Яскравими представниками такої точки зору на програмування є, наприклад, функціональний, об’єктно-орієнтований підходи, назви яких вже говорять самі за себе [1, 3]. Інша “спрощуюча” точка зору хоча і акцентує увагу на процесі, але штучно обмежує розгляди вузькими їх класами. Це стосується, наприклад, підходу систем алгоритмічних алгебр [4] та меншою мірою навіть структурного підходу [5]. Природно, що згаданими підходами не обмежується перелік спрощених точок зору на програмування, однак названі є в цьому переліку репрезентативними [6–9]. Будь-яка з них занадто суб’єктивізує процес розв’язання програмістських задач, ставлячи на чільне місце тут інтуїтивну основу, та не дає змоги сьогодні серйозно говорити про найважливіші проблеми сучасного програмування, такі як керування якістю створюваних програм, ефективність їх виробництва та збереження інвестицій. Лавиноподібне наростання подібної фактографії стимулювало розмови про кризисні прояви в програмуванні, депресії в галузі комп’ютерної індустрії тощо [5, 8–10], у той час як мова мала б іти не про кризу в галузі, а про кризу шляхів її розвитку! Сказане ще раз показує, що сучасне програмування як інформаційно-технічна галузь у цілому вже не може ефективно розвиватись виключно на суб’єктивно-інтуїтивній основі, що є джерелом диференціації уявлень про ПД. Проблеми галузі вже давно вийшли за ту межу складності та значимості, коли інтуїтивні висновки необхідно адекватно суб'єктивізувати, доповнивши їх, по можливості, точними дослідженнями та розробками. Мова йде насамперед про головний носій інтуїції в ПД — програмування як процес побудови програм. Питання, що пов'язані з розкриттям семантики мов програмування, грають тут ключову роль. Тому дослідження цієї проблематики, розробка загального методу отримання алгебричних характеристик класів функцій та застосування його для опису прагматично важливого класу частково рекурсивних функцій над записами є метою цієї роботи. Першорядне значення тут має композиційна парадигма програмування [11] як методологічна основа розгляду всього розмаїття як загальних, так і окремих методів побудови програм. Саме вони, точніше їх експлікації у вигляді тих або інших класів композицій [12], є об'єктом дослідження. Предметом дослідження є проблема побудови характеристик класів обчислюваних функцій над різними носіями в примітивних програмних алгебрах (ППА) [13—15]. Основна увага приділена пошуку породжуючих сукупностей та базисів таких алгебр. Всі невизначені нижче загальноматематичні поняття та позначення трактуються в сенсі [16, 17], а поняття теорії нумерацій та теорії алгоритмів — у сенсі [18].

Постановка задачі

Мета роботи — дослідити питання розкриття семантики мов програмування з метою розробки загального методу отримання алгебричних характеристик прагматико-обумовлених класів обчислюваних функцій; застосувати розроблений метод для опису прагматично важливого класу частково-рекурсивних функцій над записами.

Загальні положення

Носій ППА становлять *n*-арні функції та *n*-арні предикати (далі — функції та предикати) ($n = 1, 2, \ldots$). Сигнатуру ППА (що позначається тут Ω) становлять операції суперпозиції, розгалуження та циклювання, які являють собою адекватні уточнення основних методів конструювання програм, що властиві більшості високорівневих мов програмування [1—5, 8, 9]. Нагадаємо формальні визначення цих операцій. Для зручності та компактності викладення при позначенні функцій та предикатів перевага буде віддаватись не операторній, а термальній формі запису [16].

Нехай задані *m* функцій $f_1, \ldots, f_m$ однакової арності (наприклад, k) вигляду $A^k \to B$, визначені на деякій попередньо зафіксованій множині A зі значеннями з множини B [12, 16] (не обов'язково, щоб $A \cap B = \varnothing$, більше того — допускається такий випадок, коли $A \cap B = \varnothing$) та нехай на множині B визначена *m*-арна функція f, що набуває значення з деякої множини C. Розглянемо нову *k*-арну функцію $g : A \to C$ таку, що її значення на аргументі $\langle a_1, \ldots, a_k \rangle$ задається таким чином: $g(\langle a_1, \ldots, a_k \rangle) \cong$ $\cong f(f_1(\langle a_1, \ldots, a_k \rangle), \ldots, f_m(\langle a_1, \ldots, a_k \rangle))$. Будемо казати, що функція g є результатом застосування $(m+1)$-арної *суперпозиції*, що позначається S^{m+1}, до кортежу функцій $\langle f, f_1, \ldots, f_m \rangle$, іншими словами $g \equiv S^{m+1}(\langle f, f_1, \ldots, f_m \rangle)$.

Нехай тепер, крім вже заданих функцій $f_1, \ldots, f_m$, додатково задані функція *h* того ж вигляду $A^k \to B$ та *m*-значна функція $\beta : B \to$ $\to \{1, 2, \ldots, m\}$. Будемо казати, що *k*-арна функція $g : A^k \to B$ виникає з функцій *h*, $f_1, \ldots, f_m$ $(m+1)$-арної параметричної операції розгалуження $\Diamond_\beta^{m+1}$, якщо для довільного аргументу $\langle a_1, \ldots, a_k \rangle \in A^k$ значення функції $g(\langle a_1, \ldots, a_k \rangle)$ задається таким чином: $g(\langle a_1, \ldots, a_k \rangle) \cong$ $\cong f_r(\langle a_1, \ldots, a_k \rangle)$, якщо $\beta(h(\langle a_1, \ldots, a_k \rangle)) = r$, ($1 \le r \le m$).

Відзначимо, що таким чином введена параметрична операція розгалуження являє собою адекватне уточнення відомого методу конструювання програм типу $\mathrm{case\_of}$. Корисним частковим випадком цієї операції, очевидно, є тернарна операція *розгалуження* $\Diamond$, яка ставить у відповідність двом функціям $f_i : A^k \to$ $\to B$, $i = 1, 2$, та одному предикату $p : A^k \to$ $\to \{T, F\}$ *k*-арну функцію $g \equiv \Diamond(\langle p, f_1, f_2 \rangle)$, значення якої на довільному аргументі $\langle a_1, \ldots, a_k \rangle \in A^k$ задається так:

$$g(\langle a_1, \ldots, a_k \rangle) \cong \begin{cases} f_1(\langle a_1, \ldots, a_k \rangle), p(\langle a_1, \ldots, a_k \rangle) = T, \\ f_2(\langle a_1, \ldots, a_k \rangle), p(\langle a_1, \ldots, a_k \rangle) = F. \end{cases}$$

Тепер поповнимо список заданого k-арним предикатом $p: A^k \to \{T, F\}$. Розглянемо k-арну функцію $g: A^k \to B$, значення якої $g(\langle a_1, \dots, a_k \rangle)$ на довільному аргументі $\langle a_1, \dots, a_k \rangle \in A^k$ вважається рівним першій компоненті першого кортежу послідовності кортежів $[\langle a_1^i, \dots, a_k^i \rangle]_{i=0,1,2,\dots}$, де $a_j^0 = a_j$, $j = 1, 2, \dots, k$, та $a_j^{i+1} = f_j(\langle a_1^i, \dots, a_k^i \rangle)$, $j = 1, 2, \dots, k$, для якого (позначимо його, наприклад, $\langle a_1^s, \dots, a_k^s \rangle$) $p(\langle a_1^s, \dots, a_k^s \rangle) = F$ за умови, що для всіх $r = 1, 2, \dots, s-1$ значення $p(\langle a_1^r, \dots, a_k^r \rangle) = T$. Функція g виникає застосуванням $(m+1)$-операції циклування до функцій кортежу $\langle p, f_1, \dots, f_m \rangle$. Домовимось позначати її $g \equiv$ $\equiv *^{m+1}(\langle p, f_1, \dots, f_m \rangle)$. Таким чином, відповідно до сказаного, $g(\langle a_1, \dots, a_k \rangle) = a_1^s$. Відзначимо, до сих пір для позначення введених операцій нами використовувалась виключно операторна форма запису. Використовуючи ж термальну форму запису операцій сигнатури ППА, будемо явно вказувати тільки ті змінні, значення яких мають вагу в цьому випадку. Наприклад, для операції циклування будемо використовувати запис вигляду $p(x_1, \dots, x_m) \underset{y_1 \dots y_k}{*} \langle f_1(z_1^1, \dots, z_{m_1}^1), \dots,$ $f_m(z_1^k, \dots, z_{m_k}^k) \rangle$, вказуючи явно тільки ті змінні, від яких функції та предикат значно залежать. При цьому функція $f_j(z_1^j, \dots, z_{m_j}^j)$ "керує" змінною y_j, $j = 1, 2, \dots, k$, а змінна y_1 вважається "вихідною". Спосіб відновлення операторної форми тут очевидний.

Зафіксуємо тепер деяку зліченну множину D та для деякого $k \in N$ розглянемо класи Φ^k часткових k-арних функцій та предикатів вигляду $D^k \to D$ і $D^k \to \{N, F\}$ відповідно та $\Phi \equiv \bigcup_k \Phi^k$ багатомісних часткових функцій і предикатів на D. Далі під функціями (предикатами) на D, D-функціями (D-предикатами), будемо розуміти функції (предикати) з Φ. Обчислюваність на D вводиться як нумераційна обчислюваність [18]. Домовимось через $A_D^{\text{чр}}$ позначати ППА, носій якої становлять частково рекурсивні функції (обчислювані функції, чр-функції) на D. Породжуючу множину алгебри $A_D^{\text{чр}}$ назвемо її *повною системою* (ПС), а повну систему ППА – її I_m^n*-базисом*, якщо будь-яка її підсистема, що отримується видаленням з неї будь-якої функції, відмінної від селекторної чи предиката, вже не буде повною.

При дослідженні повних систем ППА корисними будуть деякі наведені нижче позначення, терміни, а також властивості та супутні їм результати.

***Властивість* 1.** n-арна функція f зберігає множину $L \subset D, L \neq \varnothing$, якщо $f(\underbrace{L \times \dots \times L}_{n}) \subseteq$ $\subseteq L$ [13].

Нехай тепер D – множина об'єктів, відносно яких коректно говорити про їх складові. Уявімо, що універсально множина таких складових є зліченною. Позначимо її B. Зафіксуємо відображення $\beta: D \to 2^B$, де 2^B – множина всіх скінченних підмножин множини B. Змістовно кажучи, для будь-якого $d \in D$ $\beta(d) \in 2^B$ – множина елементів з B, які є складовими d – його денотатами.

***Властивість* 2.** n-арна функція f β-зберігає денотати, якщо існує скінченна множина $B_f \subset$ $\subset B$ така, що для будь-якого $d \equiv \langle d_1, \dots, d_n \rangle \in$ $\in \mathrm{dom}\, f$ справедливо $\beta(f(\langle d_1, \dots, d_n \rangle)) \subseteq$ $\subseteq \bigcup_{i=1}^{n} \beta(d_i) \cup B_f$.

Цікаво те, що наведені властивості D-функцій зберігаються в сигнатурі Ω. Це дає можливість сформулювати низку простих і корисних необхідних умов повноти для ПС $A_D^{\text{чр}}$.

Твердження 1. Будь-яка повна система алгебри $A_D^{\text{чр}}$ для будь-якої непустої множини L $(L \subset D,\ L \neq \varnothing)$ містить D-функцію, яка не зберігає множину L.

Твердження 2. Будь-яка повна система алгебри $A_D^{\text{чр}}$ містить хоча б одну D-функцію, яка не β-зберігає денотати.

Корисним наслідком останнього твердження, очевидно, є таке твердження.

Твердження 3. Будь-яка повна система алгебри $A_D^{\text{чр}}$ містить хоча б одну нескінечнно-значну D-функцію.

Відзначимо, що в ППА легко моделюються логічні зв'язки довільної складності через завжди істинний та завжди хибний предикати — p_T та p_F відповідно, за допомогою часткового випадку операції $\Diamond_\beta^{m+1} - \Diamond$. Так, наприклад,

$$p(x_1,\dots,x_n) \vee q(y_1,\dots,y_m) = \Diamond(p(x_1,\dots,x_n), p_{\mathrm{T}}(x_1), \Diamond(q(y_1,\dots,y_m), p_T(x_1), p_{\mathrm{F}})).$$

$$p(x_1,\dots,x_n)\,\&\,q(y_1,\dots,y_m) = \Diamond(p(x_1,\dots,x_n), \Diamond(q(y_1,\dots,y_m), p_T(x_1), p_{\mathrm{F}}), p_{\mathrm{F}}(x_1)).$$

$$\neg p(x_1,\dots,x_n) = \Diamond(p(x_1,\dots,x_n), p_{\mathrm{F}}, p_T(x_1)).$$

Повнота в класах чр-функцій і чр-предикатів. Критерій повноти

Розглянемо загальний метод знаходження повних систем ППА *D*-функцій і *D*-предикатів. Він буде представлений рядом взаємопов'язаних результатів, що подані у вигляді доведених у процесі викладення лем і теорем. Для початку визначимо деякі поняття та домовимось про низку корисних позначень і домовленостей.

Зафіксуємо дві зліченні множини D_1 та D_2 за заданими для них ефективними нумераціями* $\alpha_1 : N \to D_1$, $\alpha_2 : N \to D_2$ та розглянемо ППА $A_{D_1}^{\text{чр}}$ і $A_{D_2}^{\text{чр}}$. Домовимось позначати елементи множин D_1 та D_2 маленькими літерами $a^1, b^1,\dots$ і $a^2, b^2,\dots$, можливо з нижніми індексами, відповідно. Нехай для алгебри $A_{D_1}^{\text{чр}}$ вирішена проблема повноти, іншими словами, задана її повна система σ_{D_1} та конструктивно задані ін'єктивні відображення $\varphi : D_2 \to D_1$ і $\Phi : D_1 \to D_2$. Причому множини $\varphi(D_2)$ і $\Phi(D_1)$ — рекурсивні [6, 9]. Розглянемо підхід до вирішення проблеми повноти для алгебри $A_{D_2}^{\text{чр}}$.

Для позначення D_1- і D_2-функцій використовуємо маленькі та прописні літери $f, g,\dots$ і $F, G,\dots$ відповідно, а для D_1- і D_2-предикатів — літери $p, r,\dots$ і $P, R,\dots$ відповідно. При використанні термальної форми запису змінні для D_1-функцій і D_1-предикатів будемо позначати маленькими літерами латинського алфавіту $x, y, z,\dots$, а для D_2-функцій і D_2-предикатів — маленькими літерами грецького алфавіту $\tau, \xi, \pi,\dots$ У всіх випадках можливе використання нижніх індексів.

***Означення* 1.** $\varphi(D_2)$-функцію $f(x_1,\dots,x_n)$ назвемо D_1-образом D_2-функції $F(\tau_1,\dots,\tau_n)$, якщо $f(\varphi(a_1^2),\dots,\varphi(a_n^2)) \cong \varphi(F(a_1^2,\dots,a_n^2))$ для будь-яких $a_1^2,\dots,a_n^2 \in \varphi(D_2), \varphi(D_2) \subseteq D_1$.

***Означення* 2.** $\varphi(D_2)$-предикат $p(x_1,\dots,x_n)$ назвемо D_1-образом D_2-предиката $P(\tau_1,\dots,\tau_n)$, якщо $p(\varphi(a_1^2),\dots,\varphi(a_n^2)) \cong P(a_1^2,\dots,a_n^2)$ для будь-яких $a_1^2,\dots,a_n^2 \in \varphi(D_2), \varphi(D_2) \subseteq D_1$.

Покажемо, що задекларовані наведеними визначеннями відношення "бути образом функції" та "бути образом предиката" зберігають властивість часткової рекурсивності. Строго кажучи, справедлива така теорема.

Теорема 1. D_1-образ D_2-чр-функції (D_2-чр-предиката) є D_1-чр-функцією (D_1-чр-предикатом).

Дійсно, зважаючи на ефективність нумерацій α_1 та α_2, а також на конструктивність відображення φ, легко перевірити, що φ як відображення нумерованої множини $\langle D_2, \alpha_2\rangle$ на нумеровану множину $\langle \varphi(D_2), \alpha_1(\alpha_2^{-1}(\varphi(D_2)))\rangle$ є чр-еквівалентністю [16, 18]. Застосувавши теорему 2.1.5 [18], отримаємо, що справедлива така лема.

Лема 1. D_1-образом D_2-чр-функції (D_2-чр-предиката) є $\varphi(D_2)$-чр-функція ($\varphi(D_2)$-чр-предикат).

Звідси, а також з рекурсивності множини $\varphi(D_2)$, безпосередньо слідує така лема.

Лема 2. Будь-яка $\varphi(D_2)$-чр-функція є D_1-чр-функцією. Аналогічно для предикатів.

Звідси безпосередньо слідує справедливість теореми 1.

***Означення* 3.** D_2-чр-функцію $F(\tau_1,\dots,\tau_n)$ назвемо D_2-моделлю D_1-функції $f(x_1,\dots,x_n)$, якщо $F(\Phi(a_1^1),\dots,\Phi(a_n^1)) \cong \Phi(f(a_1^1,\dots,a_n^1))$ для будь-яких $a_1^1,\dots,a_n^1 \in D_1$. D_2-модель D_1-предиката вводиться аналогічним чином.

* Ефективність нумерації $\alpha : N \to D$ означає, що конструктивно задана процедура відновлення (*а*) по будь-якому номеру $n \in N$ — відповідного нумерованого елемента $\alpha(n) \in D$ та (*б*) по будь-якому елементу нумерованої множини $d \in D$ — його номера $n \in N$ такого, що $\alpha(n) = d$.

Нехай, $\psi = \varphi \cdot \Phi^*$. Очевидно що $\psi : D_2 \to \to \Phi(\varphi(D_2))$ – бієкція. Тому коректно казати про обернене до ψ відображення. Через χ позначимо деяке розширення відображення $\psi^{-1} : D_2 \to \to D_2$. Змістовно кажучи, D_2-функції ψ та χ грають ролі кодуючої та розкодуючої функцій відповідно. Позначимо σ_{D_2} таку сукупність D_2-функцій і D_2-предикатів, що, по-перше, D_2-модель D_1-функції (D_1-предиката) з ПС σ_{D_1} може бути побудована з D_2-функцій та D_2-предикатів сукупності σ_{D_2} скінченним застосуванням операцій сигнатури Ω і, по-друге, D_2-функції ψ та χ аналогічним способом можуть бути побудовані з D_2-функцій та D_2-предикатів сукупності σ_{D_2}.

***Означення* 4.** Впорядковану шістку $\Sigma \equiv \equiv \langle D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi \rangle$ назвемо допустимою системою (ДС), а пару $\langle D_1, \sigma_{D_1} \rangle$ – її основою.

Очевидно, що в контексті введених кодуючої та розкодуючої функцій справедлива така лема.

Лема 3. Нехай $F(\tau_1,\ldots,\tau_n)$ – D_2-чр-функція, а $H(\pi_1,\ldots,\pi_n)$ – D_2-модель D_1-образу функції $F(\tau_1,\ldots,\tau_n)$. Тоді $F(a_1^2,\ldots,a_n^2) \cong \chi(H(\psi(a_1^2), \ldots, \psi(a_n^2)))$ для будь-яких $a_1^2,\ldots,a_n^2 \in D_2$.

Лема 4. Нехай $P(\tau_1,\ldots,\tau_n)$ – D_2-чр-предикат, а $R(\pi_1,\ldots,\pi_n)$ – D_2-модель D_1-образу предиката $P(\tau_1,\ldots,\tau_n)$. Тоді $P(a_1^2,\ldots,a_n^2) \cong R(\psi(a_1^2),\ldots, \psi(a_n^2))$ для будь-яких $a_1^2,\ldots,a_n^2 \in D_2$.

Звідси безпосередньо слідує, що справедлива така теорема.

Теорема 2. σ_{D_2} – ПС алгебри $A_{D_2}^{\text{чр}}$.

Зважаючи на те що елементи множин D_1 і D_2 – абстрактні об'єкти, в структуру ми не вникаємо, а до самих множин ставляться лише найбільш загальні вимоги, проведені побудови мають максимально загальний характер. Це дає змогу сформулювати просту, але ефективну умову повноти системи функцій ППА.

[*] $f \cdot g$ є стандартним добутком функцій, іншими словами така функція, що $\mathrm{dom}(f \cdot g) \equiv \mathrm{dom}(f)$ та $\mathrm{ran}(f \cdot g) \subseteq \subseteq \mathrm{ran}(g)$ і яка будь-якому $d \in \mathrm{dom}(f \cdot g)$ зіставляє значення $f \cdot g(d) \equiv g(f(d))$.

Отже, нехай $D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi$ – об'єкти, введені раніше. Тоді справедлива така теорема.

Теорема 3. Якщо $\langle D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi \rangle$ є ДС, то σ_{D_2} – ПС алгебри $A_{D_2}^{\text{чр}}$.

Значимість цього результату полягає ще й у тому, що в абсолютно мотивованому сьогодні припущенні справедливості тези Черча в його тлумаченні в контексті властивості нумераційної обчислюваності теорема перетворюється на критерій повноти.

Теорема (критерій повноти). Щоб система σ_{D_2} була ПС алгебри $A_{D_2}^{\text{чр}}$, необхідно та достатньо, щоб система $\langle D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi \rangle$ була допустимою системою.

Отримані результати дають у своїй сукупності досить цілісне уявлення про метод побудови повних систем ППА частково рекурсивних функцій та предикатів над зліченними множинами. Нижче цей метод буде застосований для вирішення проблеми повноти ППА у класі чр-функцій та чр-предикатів над прагматично значним у програмування типом даних – множиною записів.

ППА чр-функцій і чр-предикатів над множиною записів

Різноманітні інтуїтивні трактування запису дуже широко представлені в інформаційних технологіях і програмуванні. Незважаючи на те що окремі його розуміння досить сильно різняться, всім їм властиве намагання адекватно тою чи іншою мірою використовувати для опису складно агрегованих сутностей іменні структури. Часто ці намагання обтяжені незначними частинними деталями, що розмивають у таких описах системотвірну значимість власне механізмів іменування. Хоча, як показує досвід, саме іменні структури тут становлять той "спільний знаменник", через який варто розглядати всі інші аспекти розв'язуваних задач. Саме це намагання лежить в основі наступних побудов.

Зафіксуємо V та W – непусті зліченні множини елементів, що трактуються відповідно як множини імен та значень (денотатів). У загальному випадку допускається, що деякі імена можуть виступати у ролі значень та навпаки; іншими словами, можливо, що $V \cap W \neq \varnothing$.

Для подальшого викладення необхідно домовитись про деякі позначення, ввести основні та допоміжні визначення. Деякі з них дамо зараз же, інші будемо вводити в міру необхідності. Усі невизначені поняття та позначення будуть трактуватись у сенсі [11].

Одним із основних понять цього розділу є запис. Множину всіх записів над множиною імен V і значень W позначимо $Z^{(V,W)}$. Тепер дамо визначення самого запису.

***Означення* 5.** Під записом над множиною імен V і значень W (просто записом, якщо з контексту зрозуміло, про які V та W йде мова) будемо розуміти скінченне функціональне бінарне відношення між множинами імен V та значень W.

Для позначення записів домовимось користуватись прописними літерами $I, J, K, \ldots$ Маленькими літерами $u, v, w, \ldots$ будемо позначати імена елементів записів, літерами $a, b, c, d, \ldots$ – їх значення, а літерами $\lambda, \mu, \eta, \ldots$ – власне елементи записів. У всіх випадках за необхідності можливе використання індексів. За необхідності явно вказати в позначенні елемента запису його імені та (або) значення будемо використовувати відповідно верхній та (або) нижній ліві індекси. Наприклад, нехай $\lambda = (v, a)$. Тоді допустимі такі варіанти позначення цього елемента: ${}^{v}\lambda$, ${}_{a}\lambda$ і ${}^{v}_{a}\lambda$.

У подальшому викладенні доречним може бути використання поряд із записами їх так званих схем, які є іменними шаблонами відповідних записів.

***Означення* 6.** Під схемою запису K будемо розуміти скінченну множину імен $\{v_1, \ldots, v_n\}$, яка являє собою проекцію цього запису по першій компоненті, іншими словами $\{v_1, \ldots, v_n\} = = pr_1(K)$, де pr_i – функція проекції по i-й компоненті m-арного відношення ($1 \le i \le m$) [11].

Домовимось схему запису I позначати $sh(I) = \{v_1, \ldots, v_n\}$, а сам запис називати для компактності $sh(I)$-записом, або записом типу $sh(I)$. Записи, що мають однакові схеми, домовимось називати односхемними записами. У випадку необхідності явно вказати тип запису I будемо використовувати позначення $I^{sh(I)}$. Множину усіх записів типу $\{v_1, \ldots, v_n\}$ позначимо $Z[\{v_1, \ldots, v_n\}]$. Тоді матимуть місце такі частинні випадки: $I^{\varnothing} = \varnothing$ та $Z[\varnothing] = \{\varnothing\}$. Зважаючи на сказане, очевидно, що $Z^{(V,W)} \equiv \bigcup_{V' \in 2^V} Z[V']$. У розгорнутій формі будемо подавати запис як $I^{\{v_1, \ldots, v_n\}} \equiv \{(v_1, d_1), \ldots, (v_n, d_n)\}$.

Для коректного використання нумераційної обчислюваності на множині записів необхідно довести існування ефективної нумерації множини $Z^{(V,W)}$. Враховуючи зліченність множин V, W, а також те, що у цьому випадку важлива не стільки форма подання імен та значень, скільки їх принципово різні ролі, можна, не обмежуючи загальність наступних побудов, вважати, що $V = W \equiv N$. Таким чином, усі подальші формальні побудови будемо проводити над множиною записів $Z^{(V,W)} \equiv Z^{(N,N)}$.

Здійснимо побудову нумерації в декілька кроків. По-перше, необхідно врахувати, що для будь-якого непустого запису $I = \{(v_1, d_1), \ldots,$ $(v_m, d_m)\}$ її номер збігається з номером скінченної множини $M_I = \{n_1, \ldots, n_m\}$, де n_i є номером іменованого елемента запису $(v_i, d_i)^*$. Для самої ж множини M_I її номер задається, наприклад, так:

$$\alpha'(M) \equiv 2^{n_{j_1}+1} \cdot 3^{n_{j_2}+1} \cdot \ldots \cdot p_{m-1}^{n_{j_m}+1},$$

де $n_{j_1} < n_{j_2} < \ldots < n_{j_m}$, $n_{j_s} \in M$, $s = 1, \ldots, m$, а p_i, $i = 0, 1, 2, \ldots$ – i-те просте число ($p_0 = 2$, $p_1 = 3$, $p_2 = 5$, ...). Тоді шукана нумерація $\alpha_{Z^{(V,W)}}$ множини $Z^{(V,W)}$ задається через таку кускову схему:

$$\alpha'_{Z^{(V,W)}}(K) = \begin{cases} 1, \text{ якщо } I = \varnothing, \\ \alpha'(M_K) \text{ інакше}, \end{cases}$$

де K – деякий запис. Через те що $\alpha'_{Z^{(V,W)}}$, очевидно, є бієкцією, можна вважати, що $\alpha_{Z^{(V,W)}} \equiv (\alpha'_{Z^{(V,W)}})^{-1}$.

Перейдемо безпосередньо до питання відшукання повної системи ППА чр-функцій та чр-предикатів над записами $A^{\text{чр}}_{Z^{(V,W)}}$. Зважаючи на отримані вище результати, вирішення проблеми повноти ППА $A^{\text{чр}}_{Z^{(V,W)}}$ зводиться до побудови відповідної допустимої системи. Для цього звернемось до поняття мультимножини, розглянутого, наприклад, у [14, 15]. Нехай U – деяка скінченна, можливо пуста, множина.

* Ефективна нумерація множини N^2 побудована, наприклад, у [19].

***Означення* 7.** Мультимножиною α з основою U будемо називати скінченновизначену функцію вигляду $\alpha : U \to N^{+}$, де N^{+} – множина натуральних чисел без нуля, тобто $N^{+} \equiv N \backslash \{0\} = \{1, 2, 3, \ldots\}$. У випадку необхідності вказати основу α будемо використовувати для цієї мультимножини також позначення α^{U}.

Без обмеження загальності можна вважати, що $U \subseteq N$. Позначимо сукупність усіх мультимножин з основою U через M_U. Тоді, очевидно, що $M \equiv \bigcup_{U \in 2^N} M_U$ – множина усіх мультимножин (над N).

Домовимось елементи множини M позначати маленькими літерами грецького алфавіту $\alpha, \beta, \delta, \ldots$, можливо з індексами. Елементи мультимножини будемо позначати парами вигляду $\langle a, d \rangle$, кожна з компонент може бути з індексом. Першу компоненту пари – a – будемо називати аргументом, другу – d – значенням (денотатом, кратністю). З мультимножинами зручно пов'язати поняття характеристики χ_{α} та повного образу $f[\alpha]$. Перша являє собою параметричну функцію $\chi_{\alpha} : D \to N$, значення якої задається кусковою схемою $\chi_{\alpha}(a) = \begin{cases} \alpha(a), \text{ якщо } a \in \mathrm{dom}\,\alpha, \\ 0 \text{ інакше} \end{cases}$ для всіх $a \in N$. Другий – по мультимножині α^{U} та відносно функції $f : D \to D$ будує мультимножину $f[\alpha]^{f(U)}$, де $f(U)$ – повний образ множини U відносно функції f, а характеристика довільного аргументу a цієї множини задається таким чином: $\chi_{f[\alpha]^{f(U)}}(a) = \sum_{a' \in f^{-1}(a)} \chi_{\alpha}(a')$, де $f^{-1}(a)$ означає повний прообраз елемента a відносно функції f. При цьому у випадку пустої множини доданків сума вважається рівною нулю.

Перейдемо тепер до розгляду ППА $A_M^{\text{чр}}$ M-чр-функцій та M-чр-предикатів. Цікавою є така сукупність M-чр-функцій та M-чр-предикатів σ_M, яка включає *предикат рівності* $\alpha = \beta$, що визначається, наприклад, так: $\alpha = \beta \Leftrightarrow \forall a (a \in N \Rightarrow \chi_{\alpha}(a) = \chi_{\beta}(a))$; *функцію об'єднання* $\bigcup_{\mathrm{All}}$ таку, що мультимножинам α та β зіставляють таку множину $\alpha \bigcup_{\mathrm{All}} \beta$, що для будь-якого аргументу a його характеристика рівна $\max(\chi_{\alpha}(a), \chi_{\beta}(a))$, тобто $\chi_{\alpha \bigcup_{\mathrm{All}} \beta}(a) \equiv \equiv \max(\chi_{\alpha}(a), \chi_{\beta}(a))$; функцію *прямого з'єднання* $\otimes$, яка довільним мультимножинам $\alpha^{U_{\alpha}}$ і $\beta^{U_{\beta}}$ зіставляє мультимножину $(\alpha^{U_{\alpha}} \otimes \beta^{U_{\beta}})^{U_{\alpha} \times U_{\beta}}$, характеристика аргументів $< a_1, a_2 >$ якої задається так: $\chi(< a_1, a_2 >) = \chi_{\alpha^{U_{\alpha}}}(a_1) \cdot \chi_{\beta^{U_{\beta}}}(a_2)$, $\forall a_1, a_2 \in \in N$; функції *додавання* $\oplus$ та *віднімання* $\div$, що визначаються відповідно виразами $\alpha \oplus \beta = = +[\alpha \otimes \beta]$ і $\alpha \div \beta = \dot{-}[\alpha \otimes \beta]$; константні функції $\{1^1\}(\alpha)$ та $\varnothing_m(\alpha)$, що фіксують відповідно мультимножини $\{< 1, 1 >\}$ та $\varnothing_m \equiv \alpha^{\varnothing}$; функцію *кратності* φ, що зіставляє двом мультимножинам вигляду $\{\langle n, 1 \rangle\}$ та $\{\langle r, 1 \rangle\}$ мультимножину $\{\langle n, r \rangle\}$; а також селекторні функції I_m^n. Значимість описаної вище сукупності σ_M полягає в тому, що справедлива така теорема.

Теорема (про повноту мультимножинної ППА). Сукупність $\sigma_M \equiv \{=, \bigcup_{\mathrm{All}}, \oplus, \div, \{1^1\}, \varnothing_m, \varphi, I_m^n\}_{m=1,\ldots,n}^{n=1,2,3,\ldots}$ є повною системою ППА $A_M^{\text{чр}}$ [14, 15].

Через подібність множини записів та мультимножин нескладно побудувати ін'єктивне відображення з множини записів у мультимножини $\varphi : Z^{(V,W)} \to M$, що визначається як $\varphi(K) = \{\langle v_1, d_1 + 1 \rangle, \langle v_2, d_2 + 1 \rangle, \ldots, \langle v_m, d_m + 1 \rangle\}$, де $m \in N$ – кількість елементів запису K. Обернене до нього відображення $\Phi : M \to Z^{(V,W)}$ будується аналогічним чином: $\Phi(\delta) = \{(a_1, d_1 - 1), (a_2, d_2 - 1), \ldots, (a_m, d_m - 1)\}$, де δ – деяка мультимножина, а $m \in N$ – кількість елементів у ній. Крім цього, очевидно, що множини $\varphi(Z^{(V,W)})$ і $\Phi(M)$ рекурсивні.

З урахуванням сказаного вище має місце така лема.

Лема 5. M-образ $Z^{(V,W)}$-чр-функції ($Z^{(V,W)}$-чр-предиката) є $\varphi(Z^{(V,W)})$-чр-функцією ($\varphi(Z^{(V,W)})$-чр-предикатом).

З очевидної рекурсивності множини $\varphi(Z^{(V,W)})$ можна зробити висновок, що будь-яка $\varphi(Z^{(V,W)})$-чр-функція є M-чр-функцією. Аналогічно для предикатів. Таким чином, справедливий такий наслідок.

Наслідок. *M*-образ $Z^{(V,W)}$-чр-функції ($Z^{(V,W)}$-чр-предиката) є *M*-функцією (*M*-предикатом).

Розглянемо наступні функції та предикати на множині записів, навівши для окремих із них прості, але репрезентативні приклади застосування. Попередньо введемо корисну для подальших побудов допоміжну параметричну функцію проекції запису $pr_{\{v_{i_1},\ldots,v_{i_k}\}}$, яка будь-якому запису, наприклад, $I=\{(v_1,d_1),\ldots,(v_n,d_n)\}$, ставить у відповідність новий запис $pr_{\{v_{i_1},\ldots,v_{i_k}\}}(I)\equiv$ $\equiv\{(v_{j_1},d_{j_1}),\ldots,(v_{j_p},d_{j_p})\}$, де $\{v_{j_1},\ldots,v_{j_p}\}\equiv$ $\equiv\{v_{i_1},\ldots,v_{i_k}\}\cap\{v_1,\ldots,v_n\}$ та $pr_{\{v_{i_1},\ldots,v_{i_k}\}}(I)\subseteq I$. Отже, *предикат рівності* $=_Z$ вводиться аналогічно предикату рівності мультимножин; *видалення по зразку* $\div^Z$: $I\div^Z J\equiv pr_{pr_1(I)\setminus pr_1(J)}(I)$. Для часткового випадку $pr_1(I)\cap pr_1(J)=\varnothing$, $I\div^Z J=$ $=I$. Наприклад, $\{(1,3),(2,10),(5,7)\}\div^Z\{(1,1),$ $(2,5),(3,7)\}=\{(5,7)\}$, $\{(5,7)\}\div^Z\{(6,5),(3,7)\}=$ $=\{(5,7)\}$, $\{(6,5),(3,7)\}\div^Z\varnothing=\{(6,5),(3,7)\}$ та $I\div^Z I=\varnothing$, $\forall I\in Z$; накладення записів ∇: для будь-яких $I,J\in Z^{(V,W)}$ $I\nabla J\equiv J\cup pr_{pr_1(I)\setminus pr_1(J)}(I)$. У випадку, коли $I\nabla\varnothing=I,\varnothing\nabla J=\varnothing$, а у випадку $pr_1(I)\cap pr_1(J)=\varnothing$ $I\nabla J=J\nabla I=J\cup I$. Наприклад, $\{(1,3),(5,7)\}\nabla\{(1,1),(2,5),(3,7)\}=$ $=\{(1,1),(2,5),(3,7),(5,7)\}$; *включення в запис* $\overset{+}{U}$:

$$\overset{+}{U}(I)\equiv\begin{cases}I\cup\{(\max(pr_1(I))+1,0)\},\text{якщо } I\neq\varnothing,\\ \{(0,0)\},\text{якщо } I=\varnothing,\end{cases}$$

для всіх $I\in Z^{(V,W)}$. Наприклад, для $I=\{(1,3),$ $(2,10)\}$ та $J=\varnothing$ отримаємо відповідно $\overset{+}{U}(I)=$ $=\{(1,3),(2,10),(3,0)\}$ та $\overset{+}{U}(J)=\{(0,0)\}$; *вибір по максимальному імені* max: $\max(I)\equiv$ $\equiv pr_{\max(pr_1(I))}(I)$; обнулення значень $\{0\}$: $\{0\}(I)\equiv J$, де $pr_1(J)=pr_1(I)\,\&\,pr_2(J)=\{0\}$. Наприклад, $\{0\}(\{(1,3),(2,10)\})=\{(1,0),(2,0)\}$; *інкремент* $\uparrow$: ставить у відповідність будь-якому "непустому" запису $I\in Z^{(V,W)}$ запис $\uparrow(I)$ такий, що $\uparrow(I)=\{(v,a+1)\mid\forall(v,a)\in I\}$; *декремент* $\downarrow$: ставить у відповідність будь-якому "непустому" запису $I\in Z^{(V,W)}$ запис $\downarrow(I)$ такий, що $\downarrow(I)=\left\{(v,b)\,\middle|\,\forall(v,a)\in I\,\&\,b=\begin{cases}a-1,a>0\\0,a=0\end{cases}\right\}$. У випадку, якщо $I=\varnothing$, то і $\uparrow(I)=\downarrow(I)=\varnothing$.

Позначимо

$$\sigma_{Z^{(V,W)}}=\left\{=_Z,\nabla,\div^Z,\overset{+}{U},\max,\{0\},\uparrow,\downarrow,I_m^n\right\},\ {}_{m=1,\ldots,n}^{n=1,2,\ldots}\ -$$

множина $Z^{(V,W)}$-чр-функцій та $Z^{(V,W)}$-чр-предикатів.

Аналогічно до попереднього розділу розглянемо $Z^{(V,W)}$-функції ψ та χ — кодуючу та розкодуючу функції відповідно, такі що $\psi\equiv\varphi\cdot\Phi$, а χ — деяке розширене відображення ψ^{-1}.

Мають місце такі результати.

Лема 6. $Z^{(V,W)}$-модель *M*-функції (*M*-предиката), яка належить множині σ_M, може бути побудована з функцій множини $\sigma_{Z^{(V,W)}}$ за допомогою операцій ППА.

Для *M*-предиката рівності та *M*-функцій $\cup_{\text{All}},\div\cdots,\{1^1\},\varnothing_m,\varphi$ побудова їх $Z^{(V,W)}$-моделей не є складною. Тому не будемо зупинятись на них. Проведемо побудову для функції додавання мультимножин $\oplus$. Для цього введемо декілька допоміжних $Z^{(V,W)}$-функцій та $Z^{(V,W)}$-предикатів, а саме: *завжди хибний* та *завжди істинний* предикати: $\text{Fal}=S(=,I_1,S(\overset{+}{U},I_1))$ та $\text{Tru}=S(=,I_1,I_1)$; *предикат нерівності* $\text{Neq}=$ $=\Diamond(S(=,I_1,I_2),\text{Fal},\text{Tru})$; *константу* $\varnothing^Z=S(\div^Z,$ $I_1,I_1)$; *вибір по зразку* Sel: $\text{Sel}(I_1,I_2)=S^3(\div^Z,$ $I_1,S^3(\div^Z,I_1,I_2))$. Наприклад, $\text{Sel}(\{(1,1),(2,2)\},$ $\{(2,3),(4,5)\})=\{(2,2)\}$ — функція "вибирає" із запису I_1 ті компоненти, імена яких містяться як імена компонентів у записі I_2. Отже, I_2 є таким собі зразком для вибірки з I_1; максимальне додавання односхемних записів: $+^{\max}=$ $=*^3S^3(\text{Neq}(I_2^2,S^2(\{0\},I_2^2)),S^2(\uparrow,I_1^2),S^2(\downarrow,I_2^2))$. Так, наприклад, для записів $I_1=\{(1,1),(2,2)\}$ та $I_2=\{(1,3),(2,5)\}$ отримаємо $+^{\max}(I_1,I_2)=\{(1,1+5),$ $(2,2+5)\}=\{(1,6),(2,7)\}$. Слід зауважити, що операція максимального додавання в загальному випадку не комутативна $+^{\max}(I_1,I_2)\neq+^{\max}(I_2,I_1)$. Комутативність зберігається тільки для записів спеціального вигляду, наприклад для односхемних одноелементних записів. Наприклад, якщо $I_1 = \{(2,2)\}$ та $I_2 = \{(2,5)\}$, то $+^{\max}(I_1, I_2) = = +^{\max}(I_2, I_1) = \{(2,7)\}$.

Тепер можемо перейти безпосередньо до побудови $Z^{(V,W)}$-функції $\oplus^Z$-моделі M-функції $\oplus$. Нехай дано записи $I_1 = \{(v_1^1, d_1^1), \ldots, (v_{n_1}^1, d_{n_1}^1)\}$ та $I_2 = \{(v_1^2, d_1^2), \ldots, (v_{n_2}^2, d_{n_2}^2)\}$ такі, що $sh(I_1) \cap sh(I_2) = \{v_{r_1}, \ldots, v_{r_s}\}$. Тоді схематично операція $\oplus^Z$ може бути продемонстрована таблицею.

Очевидно, що ця операція "розбиває" записи I_1 та I_2 на "ділянки", позначені I_{1_1}, I_{1_2}, I_{2_1} та I_{2_2} зі схемами $sh(I_{1_1}) = sh(I_1) \setminus \{v_{r_1}, \ldots, v_{r_s}\}$, $sh(I_{1_2}) = \{v_{r_1}, \ldots, v_{r_s}\}$ та $sh(I_{2_2}) = \{v_{r_1}, \ldots, v_{r_s}\}$. Таким чином, результуючий запис I_1 може бути представлений як $I_3 = I_{1_1} \cup I_{2_1} \cup I_{1_2} \oplus^Z I_{2_2}$. Щодо перших двох "доданків" I_{1_1} та I_{2_1}, то вони легко отримуються за допомогою раніше введеної функції $\div^Z$, а саме: $I_{1_1} = I_1 \div I_2$ та $I_{2_1} = I_2 \div I_1$. Що ж до $I_{1_2} \oplus^Z I_{2_2}$, то самі I_{1_2} та I_{2_2} легко задаються за допомогою функції Sel: $I_{1_2} = \mathrm{Sel}(I_1, I_2)$ та $I_{2_2} = \mathrm{Sel}(I_2, I_1)$. Залишається промоделювати $Z^{(V,W)}$-функцію $\oplus^Z$ для односхемних записів. Неважко впевнитись, що $\oplus^Z$ може бути представлена, наприклад, так:

$$\oplus^Z = *^4(\underbrace{\mathrm{Neq}(I_1^2, I_2^2)}_{p},$$

$$\underbrace{I_1^2 \nabla (\max(I_2^2) +^{\max} (\mathrm{Sel}(I_1^2, \max(I_2^2)))}_{f_1},$$

$$\underbrace{I_2^2 \div^Z \max(I_2^2))}_{f_2}.$$

Очевидно, що у випадку, коли $sh(I_1) \cap \cap sh(I_2) = \varnothing$, запис $I_{1_2} \oplus^Z I_{2_2}$ теж є порожнім, отже, $I_{1_2} \oplus^Z I_{2_2} = \varnothing$ та, як наслідок, результат $I_3 \equiv I_1 \oplus^Z I_2 = I_{1_1} \cup I_{2_1}$. З урахуванням того що $sh(I_{1_1}) \cap sh(I_{2_1}) = \varnothing$ по побудові, очевидно, що $I_3 \equiv I_1 \oplus^Z I_2 = I_1 \nabla I_2 = I_{1_1} \nabla I_{2_1}$.

Лема 7. Функції ψ та χ можуть бути побудовані з функцій сукупності $\sigma_{Z^{(V,W)}}$ скінченним застосуванням операцій ППА.

Справедливість цього результату очевидна внаслідок згаданої подібності записів і мультимножин, простоти кодуючого та розкодуючого відображень φ і Φ, а також проведених вище побудов. Тому справедлива така лема.

Лема 8. $\langle M, Z^{(V,W)}, \sigma_M, \sigma_{Z^{(V,W)}}, \psi, \chi \rangle$ – допустима система.

Звідси та з урахуванням сказаного вище безпосередньо випливає справедливість такої теореми.

Теорема 4. $\sigma_{Z^{(V,W)}} = \left\{ =_Z, \nabla, \div^Z, \overset{+}{U}, \max, \{0\}, \uparrow, \downarrow, I_m^n \right\}_{\substack{n=1,2,\ldots \\ m=1,\ldots,n}}$ – породжуюча система ППА $A_{Z^{(V,W)}}^{\text{чр}}$.

Таблиця. Схема операції $\oplus^Z$

		v_1^1	v_2^1	v_3^1	...	v_{r_1}	...	v_{r_s}	...	$v_{n_1-2}^1$	$v_{n_1-1}^1$	$v_{n_1}^1$			
		d_1^1	d_2^1	d_3^1	...	$d_{r_1}^1$	...	$d_{r_s}^1$	...	$d_{n_1-2}^1$	$d_{n_1-1}^1$	$d_{n_1}^1$			
		I_{1_1}				I_{1_2}				I_{1_1}					
v_1^2	v_2^2	...				v_{r_1}	...	v_{r_s}	...					$v_{n_2-1}^2$	$v_{n_2}^2$
d_1^2	d_2^2	...				$d_{r_1}^2$	...	$d_{r_s}^2$	...					$d_{n_2-1}^2$	$d_{n_2}^2$
I_{2_1}						I_{2_2}								I_{2_1}	
v_1^2	v_2^2	v_1^1	v_2^1	v_3^1	...	v_{r_1}	...	v_{r_s}	...	$v_{n_1-2}^1$	$v_{n_1-1}^1$	$v_{n_1}^1$		$v_{n_2-1}^2$	$v_{n_2}^2$
d_1^2	d_2^2	d_1^1	d_2^1	d_3^1	...	$d_{r_1}^1 + d_{r_1}^2$	...	$d_{r_s}^1 + d_{r_s}^2$	...	$d_{n_1-2}^1$	$d_{n_1-1}^1$	$d_{n_1}^1$		$d_{n_2-1}^2$	$d_{n_2}^2$

Висновки

Сучасна інформатико-технологічна проблематика така, що необхідний безпосередній розгляд навіть не стільки результатів розв'язків програмістських задач, скільки процесів їх розв'язання. Тому дослідження загальнозначимих структур організації процесів набувають сьогодні пріоритетного значення. Особливе місце у цих дослідження займає проблематика, пов'язана з побудовою алгебричних характеристик прагматико-обумовлених класів функцій, у т.ч. з рішенням проблем повноти у відповідних алгебрах. У статті ці питання розглядаються на основі примітивних програмних алгебр [13–15]. Тут представлений метод знаходження породжуючих сукупностей у ППА, який потім застосований для дослідження важливого в теоретичному та прикладному програмуванні класу частково рекурсивних функцій над записами. На основі введених понять повної та допустимої системи, а також отриманих результатів, у т.ч. критерію повноти, обумовлена універсальність запропонованого методу в класах обчислюваних функцій над різноманітними носіями.

Отримані результати є фундаментом для розвитку напряму адаптивних середовищ програмування. Наступні кроки пов'язані з дослідженням загального поняття композиції та розробленням пов'язаних із ним редукційних методів дослідження функцій як середовищ прагматико-обумовленої декомпозиції програмістських задач.

Список літератури

1. *Hehner E.* A Practical Theory of Programming. – New York: Springer, 2012. – 247 p.
2. *Панченко Б.Є., Гайдабрус В.М.* Реляційний каркас та модель CASE-оболонки нового // Кибернетика и системный анализ. – 2013. – № 3. – С. 172–186.
3. *Armstrong D.J.* The quarks of object-oriented development // Communications of the ACM. – 2006. – **49**, № 2. – P. 123–128.
4. *Грубій А.М.* Автоматні реалізації процесу породження послідовності Коллаца // Кибернетика и системный анализ. – 2012. – № 1. – С. 129–138.
5. *Дейкстра Э.* Дисциплина программирования. – М.: Мир, 1978. – 275 с.
6. *Страуструп Б.* Программирование: принципы и практика использования С++. – 2-е изд., испр. – М.: Вильямс, 2011. – С. 1248.
7. *What* makes software design effective? / A. Tang, A. Aleti, J. Burge, H. Vliet // Design Studies. – 2010. – **31**, № 6. – P. 614–640.
8. *Batory D., Gonsalves R., Marker B.* Dark Knowledge and Graph Grammars in Automated Software Design // Software Language Eng.: Proc. 6th Int. Conf., October 26–28, 2013, Indianapolis, USA. – Springer Intl. Publishing, 2013. – **8225**. – P. 1–18.
9. *Коваленко М.С, Павлов П.О., Овсєєць М.І.* Асинхронні розподілені обчислення з обмеженною кількістю копій структурованого програмного ресурсу // Кибернетика и системный анализ. – 2012. – № 1. – С. 105–117.
10. *Брукс Ф.П.* Проектирование процесса проектирования: записки компьютерного эксперта. – М.: Вильямс, 2012. – 464 с.
11. *Басараб И.А., Никитченко Н.С., Редько В.Н.* Композиционные базы даннях. – К.: Либідь, 1992. – 192 с.
12. *Редько В.Н., Редько И.В.* Экзистенциальные основания композиционной парадигмы // Кибернетика и системный анализ. – 2008. – № 2.– С. 3–12.
13. *Буй Д.Б., Редько И.В.* Примитивные программные алгебры функций, сохраняющих денотаты // Докл. АН УССР. – 1988. – № 9. – С. 66–68.
14. *Богатирьова Ю.О.* Обчислюваність на скінченних множинах та мультимножинах // Вісник КНУ ім. Т. Шевченка. Сер. Фіз.-мат. науки. – 2010. – № 4. – С. 88–96.
15. *Богатирьова Ю.О.* Поняття мультимножини. Структура сімейства мультимножин // Матер. XIII Міжнар. наукової конф. ім. акад. М. Кравчука, 13–15 травня 2010 р., Київ. – К.: НТУ "КПІ", 2009. – С. 60.
16. *Мальцев А.И.* Алгоритмы и рекурсивные функции. – М.: Наука, 1965. – 391 с.
17. *Мальцев А.И.* Алгоритмические системы. – М.: Наука, 1970. – 392 с.
18. *Мальцев А.И.* Конструктивные алгебры. 1 // Успехи мат. наук. – 1961. – **16**, № 3. – С. 3–60.
19. *Горєлов А.В., Редько І.В., Яганов П.О.* Композиційні засади програмістської діяльності // Вісник КНУТД. – № 4. – 2014. – С. 25–34.

References

1. E. Hehner, *A Practical Theory of Programming.* New York: Springer, 2012, 247 p.
2. B.E. Panchenko and V.N. Gajdabrus, “A relational framework and case shells of a new type”, *Cybernetica i Sistemnyj Analiz*, vol. 49, no. 3, pp. 475−486, 2013 (in Ukrainian).
3. D.J. Armstrong, “The quarks of object-oriented development”, *Communications of the ACM*, vol. 49, no. 2, pp. 123−128, 2006.
4. A.M. Grubiy , “Automaton implementations of the process of generating a Collatz sequence”, *Cybernetica i Sistemnyj Analiz*, vol. 48, no. 1 , pp. 108−116, 2012, (in Ukrainian).
5. E. Dijkstra, *A Discipline of Programming.* Moscow, Russia: Mir, 1978, 275 p. (in Russian).
6. B. Stroustrup, *Programming: Principles and Practice Using C++*, 2nd ed. Moscow, Russia: Williams, 2011, p. 1248 (in Russian).
7. A. Tang *et al.*, “What makes software design effective?”, *Design Studies*, vol. 31, no. 6, pp. 614−640, 2010.
8. D. Batory *et al.*, “Dark knowledge and graph grammars in automated software design”, in *Proc. 6th Int. Conf. Software Language Eng., Indianapolis*, Oct. 26−28, 2013, vol. 8225, pp. 1−18.
9. N.S. Kovalenko *et al.*, “Asynchronous distributed computations with a limited number of copies of a structured program resource”, *Cybernetica i Sistemnyj Analiz*, vol. 48, no. 1 , pp. 86−98, 2012 (in Ukrainian).
10. F.P. Brooks, *The Design of Design: Essays from a Computer Scientist*. Moscow, Russia: Williams, 2012, 464 p. (in Russian).
11. I.A. Basarab *et al.*, *Compositional Databases.* Kyiv, Ukraine: Lybid, 1992, 192 p. (in Russian).
12. V.N. Redko and I.V. Redko, “Existential foundations of the composition paradigm”, *Cybernetica i Sistemnyj Analiz*, vol. 44, no. 2, pp. 153−160, 2008 (in Russian).
13. D.B. Bui and I.V. Redko, “Primitive program algebras of functions, which preserve denotates”, *Doklad AN USSR*, no. 9, pp. 66−68, 1988 (in Russian).
14. Y.O. Bogatyryova, “Computability on finite sets and multi-sets”, *Visnyk KNU im. Tarasa Shevchenka*, no. 4, pp. 88−96, 2010 (in Ukrainian).
15. Y.O. Bogatyryova, “Concept of multi-set. Structure of multi-sets family”, in *Proc 13th Academician M. Kravchuk Int. Sci. Conf,* Kyiv, Ukraine, May 13−15, 2010, p. 60 (in Ukrainian).
16. A.I. Maltsev, *Algorithms and Recursive Functions.* Moscow, Russia: Nauka, 1965, 391 p. (in Russian).
17. A.I. Maltsev, *Algorythmic Systems.* Moscow, Russia: Nauka, 1970, 392 p. (in Russian).
18. A.I. Maltsev, “Constructive algebras. 1”, *Uspekhi Mat. Nauk*, no. 3, vol. 16, pp. 3−60, 1961 (in Russian).
19. A.V. Gorielov *et al.*, “Compositional foundations of programming activity”, *Visnyk KNUTD*, no. 4, pp. 25−34, 2014 (in Russian).

Т.Л. Захарченко, Д.І. Редько, І.В. Редько, П.О. Яганов

ПРИМІТИВНА ПРОГРАМНА АЛГЕБРА ОБЧИСЛЮВАНИХ ФУНКЦІЙ НАД ЗАПИСАМИ

Проблематика. Дослідження проводиться у рамках композиційного підходу до програмування. Проблематикою дослідження є розробка наукових засад генезису розв’язків програмістських задач. Його основу становить поняття композиції.

Мета дослідження. Метою дослідження є розробка загального методу отримання алгебричних характеристик класів функцій та застосування його для опису прагматично важливого класу частково рекурсивних функцій над записами.

Методи реалізації. Проведені в роботі побудови базуються на алгебричних методах дослідження програм та методах композиційного програмування. У рамках так званих програмних алгебр строго ставляться та вирішуються проблеми отримання характеристик репрезентативних класів обчислюваних функцій, проблеми знаходження породжуючих сукупностей та базисів, що посідають одне з чільних місць у програмістській проблематиці.

Результати дослідження. В роботі запропоновано загальний метод вирішення зазначених проблем у примітивних програмних алгебрах (ППА) над різними класами обчислюваних функцій. Отримані результати викладені у вигляді ряду оригінальних тверджень, лем та теорем. Вони можуть бути використанні при дослідженні алгебричних характеристик різних класів обчислюваних функцій у задачах формалізації семантик мов програмування.

Висновки. Отримані результати є фундаментом для розвитку напряму адаптивних середовищ програмування. Наступні кроки будуть пов’язані з дослідженням загального поняття композиції та розробкою пов’язаних із ним редукційних методів дослідження функцій як середовищ прагматико-обумовленої декомпозиції програмістських задач.

Ключові слова: повнота обчислюваних функцій; проблема повноти в ППА; повна система; пари натуральних чисел; чр-функції; чр-предикати.

Т.Л. Захарченко, Д.И. Редько, И.В. Редько, П.О. Яганов

ПРИМИТИВНАЯ ПРОГРАММНАЯ АЛГЕБРА ВЫЧИСЛИМЫХ ФУНКЦИЙ НАД ЗАПИСЯМИ

Проблематика. Исследование проводится в рамках композиционного подхода к программированию. Проблематика исследования – разработка научных засад генезиса решений программистских задач. Его основу составляет понятие композиции.

Цель исследования. Целью исследования является разработка общего метода получения алгебраических характеристик классов функций и применение его для описания прагматически важного класса частично рекурсивных функций над записями.

Методы реализации. Проведенные в работе построения базируются на алгебраических методах исследования программ и методиках композиционного программирования. В рамках так называемых программных алгебр строго ставятся и решаются проблемы получения характеристик репрезентативных классов вычислимых функций, проблемы нахождения порождающих совокупностей и базисов, что занимают одно из главных мест в программистской проблематике.

Результаты исследования. В работе предложен общий метод решения упомянутых проблем в примитивных программных алгебрах (ППА) над разными классами вычислимых функций. Полученные результаты изложены в виде ряда оригинальных утверждений, лемм и теорем. Они могут быть использованы при исследовании алгебраических характеристик разных классов вычислимых функций в задачах формализации семантик языков программирования.

Выводы. Полученные результаты являются фундаментом для развития направления адаптивных сред программирования. Следующие шаги будут связаны с исследованием общего понятия композиции и разработкой связанных с ним редукционных методов исследования функций как сред прагматико-обусловленной декомпозиции программистских задач.

Ключевые слова: полнота вычислимых функций; проблема полноты в ППА; полная система; пары натуральных чисел; чр-функции; чр-предикаты.

Рекомендована Радою
факультету електроніки
НТУУ “КПІ”

Надійшла до редакції
27 січня 2015 року