УДК 519.683.8

ГОРЄЛОВ А.В., РЕДЬКО І.В., ЯГАНОВ П.О.
Національний технічний університет України "КПІ"

КОМПОЗИЦІЙНІ ЗАСАДИ ПРОГРАМІСТСЬКОЇ ДІЯЛЬНОСТІ

Мета*. Розробка технологічних засад генезису рішень програмістських задач.*

Методика*. Проведені в роботі дослідження базуються на алгебраїчних методах дослідження програм та методах композиційного програмування. Основу останніх складають програмні алгебри, носіями яких є спеціальні класи функцій, а операціями – композиції, що представляють собою абстракції від засобів синтезу програм.*

Результати*. У рамках програмних алгебр строго ставляться та вирішуються проблеми повноти в класах обчислюваних функцій, що займають одне з чільних місць в програмістській проблематиці. Одним із шляхів вирішення цієї проблеми є представлений у роботі метод, що викладений у вигляді ряду оригінальних тверджень, лем та теорем. За допомогою запропонованого в роботі метода була отримана повна система алгебри для множини пар натуральних чисел.*

Наукова новизна*. В роботі запропоновано універсальній метод вирішення проблем повноти в примітивних програмних алгебрах (ППА) над різними класами обчислюваних функцій.*

Практична значимість*. Результати можуть бути використанні при дослідженні алгебраїчних характеристик різних класів обчислюваних функцій в задачах формалізації семантик мов програмування.*

Ключові слова: *повнота обчислювальних функцій, проблема повноти в ППА, повна система, пари натуральних чисел, чр-функції, чр-предикати.*

Вступ. Створення програмного продукту є таким же звичним видом людської діяльності, як і будь-яка інша, де створюється додана вартість. Ця галузь є відносно молодою, оскільки пов’язана безпосередньо з комп’ютерними технологіями. Хоча опанування технологією передбачає отримання кінцевого продукту, якість якого не залежить від випадковості, до створення програмного продукту це має відношення лише з певними застереженнями. Якісний програмний продукт схожий на витвір мистецтва. Секрети програмування відомі тільки виконавцю, якість оцінюють за кінцевим результатом, а програма, спрямована на вирішення конкретної задачі, має всі ознаки унікальності. Модифікація програмного продукту дуже часто неможлива без безпосередньої участі її творця.

Таким чином, наявне певне протиріччя у технології програмування. З одного боку, програма, як технологічний продукт, повинна відтворювати сукупність способів переробки інформації та процесів, що супроводжують ці види робіт, з іншого боку її реалізація занадто особистісна і оцінюється лише за кінцевим результатом.

Метою дослідження є розробка технологічних засад програмістської діяльності, що націлена на підтримку процесів генезису рішень програмістських задач.

Об’єктом дослідження є програмістські генетичні структури – загальнозначущі способи, методи організації процесів генезису програм, що експлікуються у вигляді тих або інших класів композицій як засобів створення сучасних відкрито-замкнутих систем

(OC-System) [1, 2]. Особливістю технології композиційного програмування є наявність набагато ширшого у порівнянні з традиційними підходами арсеналу вирішення програмістських задач. Основу розширення складає фундаментальне поняття композиції. Воно дозволяє об'єктивізувати розуміння програмування як породження та застосування композицій [3]. Це, у свою чергу, збагачує можливості розгляду програм як рішень програмістських задач вже не тільки як самостійних (окремих) об'єктів, але і в якості результатів відповідних процесів програмування. Змістовно кажучи, дана технологія програмування реалізує точку зору на рішення задачі як інтеграцію рішень її підзадач. Тому в рамках заявленої мети виокремлення та дослідження репрезентативних класів композицій є актуальною задачею.

Предметом даної роботи є розробка універсального методу вирішення проблем повноти в примітивних програмних алгебрах (ППА).

Точні визначення ППА наведені в [4, 5], всі не визначенні нижче загально математичні поняття та позначення трактуються у сенсі [6], а поняття теорії нумерацій та теорії алгоритмів – у сенсі [7, 8] відповідно.

Загальні визначення. Носієм ППА є n-арні функції і предикати для $n = 1, 2, \ldots$ [5, 8]. При їх позначенні перевага буде віддаватися не операторній, а термальній формі запису, зважаючи на її компактність [8, 9].

Сигнатуру ППА (позначатимемо її Ω) складають операції суперпозиції, розгалуження та $(n+1)$-арного циклування, що являють собою адекватні уточнення стандартних методів конструювання рішень програмістських задач, що предствлені в більшості мов програмування. Задля зручності викладення та сприйняття матеріалу нагадаємо формальне визначення найбільш складної з них – операцію циклування (детальніше див. [9]).

Циклування представляє собою $(n+1)$-арну параметричну операцію $*^{(n+1)} : \langle p, f_1, \ldots, f_n \rangle \to g$, $n = 1, 2, \ldots$, де p — предикат, а f_i, g — функції, причому арність p, f_i, g дорівнює n. Значення $g(x_1, \ldots, x_n)$ покладається рівним першій компоненті першого кортежу послідовності кортежів $\left\{\langle y_i^1, \ldots, y_i^n \rangle\right\}_{i=0,1,\ldots}$, де $y_0^j \equiv x_j$, $y_{i+1}^j \equiv f_j\left(y_i^1, \ldots, y_i^n\right)$, $j = \overline{1, n}$, для якого (позначимо його, наприклад, $\langle y_k^1, \ldots, y_k^n \rangle$) $p\left(\langle y_k^1, \ldots, y_k^n \rangle\right) = false$. Тобто, $g\left(\langle x_1, \ldots, x_n \rangle\right) = y_k^1$. Якщо такого кортежу в послідовності не існує, то $g\left(x_1, \ldots, x_n\right)$ вважається невизначеним.

Використовуючи термальний запис операцій з Ω будемо вказувати явно лише ті змінні, значення яких суттєво використовуються. Відновлення операторного запису при цьому очевидно.

Зафіксуємо деяку злічену множину D і надалі під *функціями* (*предикатами*), а при необхідності явним чином вказати множину D – D-*функціями* (D-*предикатами*) будемо розуміти часткові багатомісні функції з аргументами і значеннями із D (часткові багатомісні предикати з аргументами із D та значеннями із {T, F}, де T (F) означає істиносні значення «істина» («хиба») відповідно).

Обчислюваність на D вводиться як нумераційна обчислюваність [7]. Через $A_D^{чр}$ позначимо ППА, носій якої складають функції та предикати на D. *Породжуючу множину* алгебри $A_D^{чр}$ назвемо *повною системою* (ПС), а повну систему ППА $A_D^{чр}$ її I_m^n*– базисом*, якщо будь-яка її підсистема, що утворюється видаленням будь-якої функції (предиката), відмінної від селекторної фукнкції, вже не буде повною.

При дослідженні повних систем ППА корисними будуть наступні властивості та пов'язані з ними результати.

Визначення 1.Функція f арності n *зберігає множину* $L \subset G, L \neq \varnothing$, якщо $f(\underbrace{L \times \ldots \times L}_{n}) \subseteq L$ [7, 8].

Наступна властивість стосується випадку, коли елементи з D мають структуру. Зафіксуємо відображення $\beta : D \underset{на}{\rightarrow} 2^B$, де B – деяка злічена множина, а 2^B – множина всіх скінчених підмножин B. Змістовно, вважається, що елементи з D будуються з елементів множини B, а $\beta(d), d \in D$ – скінчена множина елементів з B, з яких побудовано d.

Визначення 2. Будемо вважати, що *функція* f арності n β*-зберігає денотати*, якщо існує скінчена множина $B_f \subset B$ така, що для будь-якого $\langle x_1, \ldots, x_n \rangle \in \mathrm{dom}\, f$ виконується $\beta\left(f\left(\langle x_1, \ldots, x_n \rangle \right) \right) \subseteq \bigcup\limits_{i=1}^{n} \beta(x_i) \cup B_f$.

Легко переконатись, що визначені властивості функцій зберігаються в сигнатурі Ω. Таким чином, очевидною є справедливість наступних простих та корисних необхідних умов повноти для породжуючих сукупностей $A_D^{чр}$.

Твердження 1. Будь-яка повна система алгебри $A_G^{чр}$ для будь-якої множини L ($L \subset \mathrm{G}, L \neq \varnothing$) містить хоча б одну функцію, яка не зберігає дану множину. для довільної множини L ($L \subset \mathrm{G}, L \neq \varnothing$);

Твердження 2. Будь-яка повна система алгебри $A_D^{\div\partial}$ містить хоча б одну функцію, що не є β-зберігаючою денотати.

Повнота в класах чр-функцій та чр-предикатів. Критерій повноти. Розглянемо загальну методологію побудови повних систем ППА чр-функцій та чр-предикатів над фіксованими зліченими множинами. Для цього зафіксуємо основні передумови розгляду, уведемо деякі поняття і встановимо систему позначень. Сам же метод буде представлений послідовністю взаємопов'язаних результатів, сформульованих у вигляді доведених у процесі викладення лем та теорем.

1. Зафіксуємо дві злічені множини D_1 та D_2, для яких існують ефективні (в інтуїтивному сенсі) нумерації $\alpha_1 : N \rightarrow D_1$ і $\alpha_2 : N \rightarrow D_2$ і розглянемо ППА $A_{D_1}^{чр}$ та $A_{D_2}^{чр}$. Тут ефективність нумерації розуміється в тому сенсі, що а) для будь-якого $n \in N$ ефективно будуються об'єкти $\alpha_i(n) \in N, i = 1,2$ та б) для будь-якого $d^i \in D_i, i = 1,2$ ефективно будуються їх відповідні номери.

2. Припустимо, що для алгебри $A_{D_1}^{чр}$ відомою є її ПС σ_{D_1}.

3. Нехай також конструктивно задані ін'єктивні відображення $\varphi : D_2 \to D_1$ і $\Phi : D_1 \to D_2$, а множини $\varphi(D_2)$ й $\Phi(D_1)$ є рекурсивними [7].

4. Будемо позначати елементи множин D_i, $i = 1{,}2$, буквами a^i, b^i, …, можливо з нижніми індексами.

5. У якості функціональних (предикатних) символів для D_1 (D_2)– функцій (D_1 (D_2) – предикатів) будемо використовувати рядкові (прописні) букви латинського алфавіту *f*, *g*, …(*F*, *G*,) (*p*, *r*, …(*P*, *R*, …)), можливо з індексами. Змінні для D_1 (D_2)-функцій (D_1 (D_2)-предикатів) будемо позначати рядковими буквами латинського та грецького алфавітів *x*, *y*, … (ζ, π, … відповідно), можливо з індексами.

6. Частково-рекурсивні D_1 (D_2)– функції (D_1 (D_2) – предикати будемо також називати D_1 (D_2)– чр-функції (D_1 (D_2) – чр-предикати. Часткова рекурсивність розуміється згідно з нумераційним підходом [7].

Визначення 3. $\varphi(D_2)$-функцію $f(x_1, \ldots, x_n)$ назвемо D_1 – образом D_2- функції $F(\zeta_1, \ldots, \zeta_n)$ якщо $f\left(\varphi\left(a_1^2\right), \ldots, \varphi\left(a_n^2\right)\right) \cong \varphi\left(F\left(a_1^2, \ldots, a_n^2\right)\right)$, для всіх $a_1^2, \ldots, a_n^2 \in \varphi(D_2), \varphi(D_2) \subseteq D_1$. Аналогічно $\varphi(D_2)$ предикат $p(x_1, \ldots, x_m)$ назвемо D_1 – образом D_2- предиката $P(\zeta_1, \ldots, \zeta_n)$ якщо $p\left(\varphi\left(a_1^2\right), \ldots, \varphi\left(a_m^2\right)\right) \cong P\left(a_1^2, \ldots, a_m^2\right)$, для всіх $a_1^2, \ldots, a_m^2 \in \varphi(D_2), \varphi(D_2) \subseteq D_1$.

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

Доказ. Використовуючи ефективність нумерацій α_1, α_2 і конструктивність φ, можна перевірити, що φ – відображення нумерованої множини $\langle D_2, \alpha_2 \rangle$ на нумеровану множину $\left\langle \varphi(D_2), \alpha_1\left(\alpha_2^{-1}(\varphi(D_2))\right)\right\rangle$ є чр-еквівалентністю [7]. Залишається застосувати теорему 2.1.5 [7].

З рекурсивності множини $\varphi(D_2)$ випливає наступна лема.

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

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

Визначення 2. D_2-функцію $F(\zeta_1, \ldots, \zeta_n)$ називатимемо D_2-моделлю D_1-функції $f(x_1, \ldots, x_n)$, якщо $F\left(\Phi\left(a_1^1\right), \ldots, \Phi\left(a_n^1\right)\right) \cong \Phi\left(f\left(a_1^1, \ldots, a_n^1\right)\right)$ для всіх $a_1^1, \ldots, a_n^1 \in D_1$. D_2-модель D_1-предиката уводится аналогічним чином.

Нехай $\psi = \varphi \cdot \Phi$. Очевидно, що $\psi : D_2 \to \Phi(\varphi(D_2))$ – бієкція. Через χ позначимо деяке розширення відображення ψ^{-1}. Таким чином, D_2-функції ψ і χ відіграють ролі, що кодуючої та розкодуючої функцій відповідно.

Позначимо через σ_{D_2} сукупність D_2-функцій та D_2-предикатів, що задовольняють наступним умовам:

1. D_2-модель D_1-функції (D_1-предиката) з сукупності σ_{D_1} може бути побудована з функцій та предикатів сукупності σ_{D_2} скінченим застосуванням операцій ППА;

2. Функції ψ і χ можуть бути побудовані з функцій та предикатів сукупності σ_{D_2} скінченим застосуванням операцій ППА.

Визначення 3. Упорядковану шістку $\left\langle D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi \right\rangle$ назвемо припустимою системою (П-системою).

Очевидно, що має місце

Лема 3. Нехай $F(\zeta_1, \ldots, \zeta_n)$ – D_2-чр-функція, а $H(\pi_1, \ldots, \pi_n)$ – D_2 -модель D_1 -образа функції $F(\zeta_1, \ldots, \zeta_n)$. Тоді $F\left(a_1^2,\ldots,a_n^2\right) \cong \chi\left(H\left(\psi\left(a_1^2\right),\ldots,\psi\left(a_n^2\right)\right)\right)$ для всіх $a_1^2,\ldots,a_n^2 \in D_2$. Аналогічно, нехай $P(\zeta_1, \ldots, \zeta_n)$ – D_2-чр-предикат, а $R(\pi_1, \ldots, \pi_n)$ – D_2 -модель D_1 -образа цього предиката. Тоді $P\left(a_1^2,\ldots,a_k^2\right) \cong R\left(\psi\left(a_1^2\right),\ldots,\psi\left(a_n^2\right)\right)$ для всіх $a_1^2,\ldots,a_n^2 \in D_2$.

Таким чином, очевидно, що має місце

Теорема 1. Множина σ_{D_2} – повна система алгебри $A_{D_2}^{чр}$.

Елементи множин D_1 і D_2 – абстрактні об'єкти, структуру яких ми не беремо до уваги. До самих же множин були пред'явлені лише вкрай загальні вимоги. Усе це дозволяє сформулювати просту, але досить загальну умову повноти системи функцій.

Отже, нехай $D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi$ – об'єкти, уведені раніше. Тоді має місце наступна

Теорема 2. Якщо $\langle D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi \rangle$ – П-система, то σ_{D_2} – ПС алгебри $A_{D_2}^{чр}$.

Особливої значущості це твердження набуває у зв'язку з тим, що воно перетворюється у критерій повноти в припущенні вірності тези Чорча в його розширенному тлумаченні, тобто при тлумаченні обчислюваності функцій і предикатів як нумераційної обчислюваності. Сформулюємо його таким чином.

Теорема 3 (критерій повноти). Система σ_{D_2} повна тоді й тільки тоді, коли $\langle D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi \rangle$ – П-система.

Сформульовані результати дають доволі цілісне уявлення про методику знаходження повних систем ППА чр-функцій та чр-предикатів над зліченими множинами. Але все ж таки видається корисним для отримання певної закінченості викладеного матеріалу розглянути застосування викладеного методу до простого, але репрезентативного прикладу.

ППА чр-функцій та чр-предикатів над множиною пар натуральних чисел. Розглянемо обчислюваність на множині пар натуральних чисел $D \equiv N^2$. Багатомісні функції парного аргументу та значення (предикати парного аргументу) називатимемо парними функціями (предикатами) або N^2– функціями (N^2– предикатами) та позначатимемо їх прописними буквами $F, G, H, \ldots$ ($P, Q, R, \ldots$). Багатомісні функції та предикати натурального аргументу та значення будемо називати арифметичними функціями та предикатами або (N – функціями (N – предикатами)) та позначатимемо їх відповідно рядковими буквами $f, g, h, \ldots$ ($p, q, r, \ldots$). Рядковими буквами $x, y, z, \ldots$ ($\xi, \pi, \rho, \ldots$) позначатимемо змінні з областю визначення N (N^2). Пари будемо позначати буквами $u, v, w, \ldots$, а натуральні числа – буквами $a, b, c, \ldots$ Відштовхуючись від результату про I_m^n -

базис ППА $A_N^{чр}$ арифметичних чр-функцій та чр-предикатів (система функцій $\sigma_N \equiv \{0, s, <, I_m^n\}_{m=1,\ldots,n}^{n=1,2,\ldots}$ – I_m^n-базис ППА $A_N^{чр}$) [9] і використовуючи результати попереднього розділу, розглянемо ППА $A_{N^2}^{чр}$ та побудуємо алгебраїчну характеристику для класу парних чр-функцій та чр-предикатів. Для цього спершу побудуємо просту та зручну для подальших викладок ефективну нумерацію множини N^2: $\alpha : N \to N^2$.

Розглянемо на множині N^2 наступне відношення часткового порядку $\succ$: дві пари $\langle a_1, b_1\rangle, \langle a_2, b_2\rangle \in N^2$ знаходяться у відношенні $\succ$, тобто $\langle a_1, b_1\rangle \succ \langle a_2, b_2\rangle$, якщо виконується хоча б одна з двох наступних умов: або $a_1 + b_1 > a_2 + b_2$, або $(a_1 + b_1 = a_2 + b_2) \,\&\, b_1 > b_2$. Таке упорядкування, очевидно, дозволяє «витягнути» множину N^2 у зростаючу послідовність виду: (0,0); (1,0); (0,1); (2,0); (1,1); (0,2); (3;0); (2,1); (1,2); (0,3),…, а значить, виходячи з визначення відношення $\succ$, ефективно задати необхідну нумерацію на N^2. Тобто, для будь-якого натурального числа c відповідна йому пара чисел $\langle a, b\rangle$ є такою, що $c = \sum_{i=0}^{a+b} i + b$. І навпаки, для знаходження пари чисел $\langle a, b\rangle$ за номером s необхідно, насамперед знайти таке натуральне l, що $\sum_{i=0}^{l} i \le s, \sum_{i=0}^{l+1} > s$. Тоді легко знаходиться другий елемент пари: $b = s - \sum_{i=0}^{l} i$, а відтак і перший: $a = l - b$. Таким чином, нумерація α дійсно є ефективною бієкцією.

Виходячи з того, що нумерація $\alpha : N \to N^2$ за побудовою є бієкцією, ін'єктивне відображення $\varphi : N^2 \to N$ представляє собою обернену до нумерації функцію, тобто $\varphi \equiv \alpha^{-1}$. Що стосується кодуючого відображення $\Phi : N \to N^2$, то воно задається, наприклад, так: $\Phi(a) \equiv \langle 0, a\rangle$, для будь-якого $a \in N$. Крім того, очевидно, що множини $\varphi(N^2)$ та $\Phi(N)$ є рекурсивними.

З огляду на вищезазначене має місце

Лема 4. N-образ N^2-чр-функції (N^2-чр-предикату) є $\varphi(N^2)$-чр-функція ($\varphi(N^2)$-чр-предикат).

З рекурсивності множини $\varphi(N^2)$ випливає, що будь-яка $\varphi(N^2)$-чр-функція є N-чр-функція і аналогічно для $\varphi(N^2)$-чр-предикатів. Таким чином, справедливий

Наслідок 2. N-образ N^2-чр-функції (N^2-чр-предикату) – є N-функцією (N-предиктом).

Тепер розглянемо наступні парні функції, що вводяться аналогічно функціям для натуральних чисел — функція константи нуля $0^2(\langle a_1, a_2\rangle) = \langle a_1, 0\rangle$; функція слідування $s^2 = (\langle a_1, a_2\rangle) = \langle a_1, a_2 + 1\rangle$, зокрема для $\langle 0, a\rangle \in \Phi(N)$, $s^2(\langle 0, a\rangle) = \langle 0, a + 1\rangle$.

Предикат «менше» $<^2$ вводиться як:

$$\langle a_1, a_2\rangle <^2 \langle b_1, b_2\rangle = \begin{cases} T,\ якщо\, a_1 + a_2 < b_1 + b_2, \\ T,\ якщо\, a_1 + a_2 = b_1 + b_2, a_2 < b_2, \\ F,\ в\, інших\, випадках \end{cases}$$

Також знадобляться додаткові функції $Inv^2, Dc^2, \oplus^2, +^2, -^2$. Функція Inv^2 міняє місця елементів пари, тобто $Inv^2(\langle a_1, a_2\rangle) = \langle a_2, a_1\rangle$. Функція Dc^2 зменшує на одиницю другий елемент пари: $Dc^2(\langle a_1, a_2\rangle) = \langle a_1, a_2 - 1\rangle$. Функція $\oplus^2$: $\oplus^2(\langle a_1, a_2\rangle) = \langle 0, a_1 + a_2\rangle$. Функція $+^2$ ($-^2$): $\langle a_1, a_2\rangle +^2 \langle b_1, b_2\rangle = \langle a_1, a_2 + b_2\rangle$ ($-^2 : \langle a_1, a_2\rangle -^2 \langle b_1, b_2\rangle = \langle a_1, a_2 - b_2\rangle$).

Позначимо $\sigma'_{N^2} = \{0^2, s^2, <^2, Inv^2, Dc^2, \oplus^2, +^2, -^2, \psi, \chi, I_m^n\}_{m=1,\dots,n}^{n=1,2,\dots}$ – множину N^2-чр-функцій та N^2-чр-предикатів.

По аналогії з попереднім розділом розглянемо N^2-функції ψ та χ – кодуючу та розкодуючу функції відповідно такі, що $\psi \equiv \varphi \cdot \Phi$, а χ – деяке розширення відображення ψ^{-1}.

Очевидно, мають місце наступні результати:

Лема 5. N^2-модель *N*-функції (*N*-предикату), що належить до множини σ_N може бути побудована з функцій множини σ_{N^2} за допомогою операцій ППА.

Доведення. Наведемо викладки для *N*-функції константи нуля 0:

$$\left.\begin{aligned} 0(\varphi(\langle 0, a_2\rangle)) = 0(a_2) = 0 \\ \varphi(o^2(\langle 0, a_2\rangle)) = \varphi(\langle 0,0\rangle) = 0 \end{aligned}\right\} \Rightarrow 0(\varphi(\langle 0, a_2\rangle)) = \varphi(0^2(\langle 0, a_2\rangle))$$

N-функція слідування *s* та *N*-предикат «менше» < можуть бути змодельовані аналогічно.

Лема 6. Функції ψ та χ можуть бути побудовані з функцій множини σ_{N^2} скінченим застосуванням операцій ППА.

Доведення.

$$\psi(\langle a_1, a_2\rangle) = = Inv^2\Big(Dc^2\Big(Inv^2\big(f_1(\langle 0,0\rangle, \langle 0,0\rangle, \langle a_1, a_2\rangle)\big) -^2 Inv^2\big(\langle a_1, a_2\rangle -^2 f_2(\langle 0,0\rangle, \langle 0,0\rangle, f_1(\langle 0,0\rangle, \langle 0,0\rangle, \langle a_1, a_2\rangle))\big)\Big)\Big)$$

Тут під f_1 розуміється $f_1(\langle a_1, a_2\rangle, \langle b_1, b_2\rangle, \langle c_1, c_2\rangle) = = *^4(p_{11}, f_{11}, f_{12}, f_{13})(\langle a_1, a_2\rangle, \langle b_1, b_2\rangle, \langle c_1, c_2\rangle)$, де $p_{11}, f_{11}, f_{12}, f_{13}$ представлені наступним чином: $p_{11}(\langle a_1, a_2\rangle, \langle b_1, b_2\rangle, \langle c_1, c_2\rangle) = \langle b_1, b_2\rangle <^2 \langle c_1, c_2\rangle$, $f_{11}(\langle a_1, a_2\rangle, \langle b_1, b_2\rangle, \langle c_1, c_2\rangle) = s^2(\langle a_1, a_2\rangle)$, $f_{12}(\langle a_1, a_2\rangle, \langle b_1, b_2\rangle, \langle c_1, c_2\rangle) = s^2(\langle a_1, a_2\rangle +^2 \langle b_1, b_2\rangle)$, $f_{13}(\langle a_1, a_2\rangle, \langle b_1, b_2\rangle, \langle c_1, c_2\rangle) = \langle c_1, c_2\rangle$. f_2 представляється аналогічним чином.

По аналогії з функцією ψ може бути представлена і функція χ. Таким чином, справедливою є

Лема 7. $\langle N, N^2, \sigma_N, \sigma'_{N^2}, \psi, \chi\rangle$ – П-система.

Теорема 4. $\sigma_{N^2} = \{0^2, s^2, <^2, Inv^2, I_m^n\}_{m=1,\dots,n}^{n=1,2,\dots}$ – I_m^n-базис ППА $A_{N^2}^{\div\eth}$

Повнота системи σ'_{N^2} випливає безпосередньо з леми 7. Базисність σ_{N^2} слідує з того, що, по-перше, функції $+^2, -^2, Dc^2$ можна очевидно отримати із функцій $0^2, Inv^2, s^2$, предикату $<^2$, а функцію $\oplus^2$ – з функцій $0^2, Inv^2, +^2$ за допомогою операцій ППА, по-друге, $\sigma_{N^2} = \sigma'_{N^2} \setminus \{+^2, -^2, Dc^2, \oplus^2\}$ і, по-третє, ні одна з функцій з системи σ_{N^2} не може бути видалена з неї без втрати властивості повноти. Адже, предикат «менше» $<^2$ не можна видалити через те, що це єдиний предикат системи σ_{N^2}, функція 0^2 не зберігає множину $N^2 \setminus \bigcup_{n=0}^{\infty} \langle n, 0\rangle$, s^2 – множину $\bigcup_{n=0}^{\infty} \langle n, 0\rangle$, а Inv^2 – множину $\bigcup_{n=0}^{\infty} \langle 0, n\rangle$.

Висновки. У роботі досліджено процес генезису рішень програмістських задач на основі композицій примітивної програмної алгебри, які є репрезентативними представниками загального поняття композиції. Представлено метод пошуку повних систем у ППА шляхом породження та застосування композицій. Обгрунтовано використання методу як інструменту підтримки програмування без втрати інформаційних та інтелектуальних інвестицій. На основі поняття повної системи та критерію повноти показана можливість вирішення проблеми повноти у класах функцій різних зліченних носіїв.

Розвиток досліджень у цьому напрямку спрямований на розробку засад відкрито-замкнених адаптивних середовищ програмування, загальнозначущим фундаментом яких є загальне поняття композиції, та на розробку редукційних методів дослідження функцій як засобів прагматико-обумовленої декомпозиції програмістських задач.

Список використаних джерел

1. Редько И.В. Экзистенциальный базис сущностных сред// Системні дослідження та інформаційні технології.–2008.– № 3 .– С.16-31
2. Редько И.В. Открыто-замкнутые основания сред интеграции. Часть 1. // Системні дослідження та інформаційні технологіїю.-№4.-2010.-С.3-14
3. Редько В.Н., Редько И.В. Экзистенциальные основания композиционной парадигмы //Кибернетика и системный анализ.– 2008.– №2.– С. 3-12
4. Губский Б.В., Крапива Е.В., Редько И.В. Вычислимые функции на реляциях и таблицах в счетном и конечном алфавитах // Докл.АН Украины.–1988.–№10.– С.74-76.
5. Буй Д.Б., Редько И.В. Примитивные программные алгебры функций, сохраняющих денотаты // Докл.АН УССР.–1988.–№9.– С.66-68.
6. Мальцев А.И. Алгоритмические системы//М.: Наука.– 1970.–392 с.
7. Мальцев А.И. Конструктивные алгебры. 1//Успехи мат.наук.–1961.– 16.–№3.– С.3–60.
8. Мальцев А.И. Алгоритмы и рекурсивные функции//М.: Наука.– 1965.–391 с.
9. Буй Д.Б., Редько И.В. Примитивные программные алгебры вычислимых функцій//Кибернетика.–1987.–№3.–С.68-74

КОМПОЗИЦИОННЫЕ ОСНОВЫ ПРОГРАММИСТСКОЙ ДЕЯТЕЛЬНОСТИ

ГОРЕЛОВ А.В., РЕДЬКО И.В., ЯГАНОВ П.А.
Национальный технический университет Украины "КПИ"

Цель. Разработка технологических основ генезиса решений программистских задач.

Методика. Проведенные в работе исследования базируются на алгебраических методах исследования программ и методах композиционного программирования. Основу последних составляют программные алгебры, носителями которых есть специальные классы функций, а операциями – композиции, которые представляют собой абстракции от средств синтеза программ.

Результаты. В рамках программных алгебр строго ставятся и решаются проблемы полноты в классах вычислимых функций, которые занимают одно из первых мест в программистской проблематике. Одним из путей решения данной проблемы является представленный в работе метод, который изложен в виде ряда оригинальных утверждений, лемм и теорем. С помощью предложенного в работе метода была получена полная система алгебры для множества пар натуральных чисел.

Научная новизна. В работе предложено универсальный метод решения проблем полноты в примитивных программных алгебрах (ППА) над разными классами вычислительных функций.

Практическая значимость. Результаты могут быть использованы при исследовании алгебраических характеристик разных классов вычислимых функций в задачах формализации семантик языков программирования.

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

COMPOSITIONAL BASICS IN PROGRAMMER ACTIVITY

HORIELOV A.V., REDKO I.V., YAGANOV P.O.
National Technical University of Ukraine “KPI”

Purpose. The purpose of research is development of technological basics for solutions of programming problems.

Methodology. Carried out investigations are based on algebraic research methods of programs and compositional programming methods. Basis of last ones consists of software algebras, with special classes of functions as carriers and compositions that represent abstractions from program synthesis tools as operations.

Findings. Problems of fullness in classes of calculable functions that took one of the first places in programming problems are definitely set and solved in the context of program algebras. One of the ways to solve these problems is method, presented in the article as series of original statements, lemmas and theorems. Using this method a complete system of algebra for set of pairs of natural numbers was achieved.

Originality. Universal method of solving the problem of fullness in primitive program algebras (PPA) on different classes of calculable functions proposed in the article.

Practical value. Results could be applied in algebraic characteristics research of different calculable functions classes in problems of programming language semantics formalization.

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