УДК 004.415.2

АЦИКЛІЧНІСТЬ ТА ЗАМКНЕНІСТЬ МАКРОКОМПОЗИЦІЙ

В.Ю. Вінник, Т.С. Парфірова

Київський національний університет імені Тараса Шевченка,
01601 Київ, вул. Володимирська, 60,
тел.: (+38044) 521 3345
E-mail: vadim.vinnik@gmail.com, tetiana.parfirova@gmail.com

Задачу експлікації проектування програмних систем розглянуто з точки зору сутнісної платформи. Обгрунтовано адекватність вибору ациклічних композицій як базовий засіб експлікації логіки будови проектів. Для ациклічних композицій встановлено замкненість відносно суперпозиції. Показано, що встановлена в попередніх роботах замкненість відносно множення та накладання є наслідком замкненості відносно суперпозиції.

The problem of explication of software design is treated from the perspective of entity platform. It is established that acyclic compositions are adequate basic means for explication of design structures and logics. For acyclic compositions, closure against superopsition is proved. It is shown that closure against multiplication and overlapping is a corollary of that against superposition.

Вступ

Загальна задача експлікації програмування [1–3], яка вирішується шляхом покрокового прагматико-обумовленого розгортання концептуально єдиного сутнісного ядра, включає в себе низку більш вузьких задач експлікації тих чи інших часткових випадків програмування а також метазадачу підтримки системної єдності через встановлення зв'язків між ними, тобто відповідно задач аналізу та синтезу. Зосередимося на перших. Декомпозиція програмування може здійснюватись „по горизонталі" – відповідно до предметних областей та індукованих ними відгалужень прикладного програмування, або „по вертикалі" – відповідно до ступеню абстракції чи конеретизації у розгляді предмету. Що стосується конкретних мов програмування, цілих парадигм (імперативної, об'єктно-орієнтованої, функціональної тощо), а також різноманітних відгалужень (бази даних, задачі символьної обробки та ін.), відповідні семантичні структури та логіка побудови програм добре вивчені завдяки численним роботам [4–7]. Разом з тим, якщо розглядати програмуваня більш абстрактно, відволікаючись від специфіки мов, середовищ та технологій, як структуровану та цілеспрямовану діяльність з побудови програмних систем, в якій центральне місце займає проектування, наявні моделі виглядають недостатньо переконливими і повними. Це і не дивно: якщо у першому випадку предмет розгляду – програма, що виконується на реальній чи ідеальній машині, – безпосередньо допускає строгий опис математичними засобами, то у другому випадку предмет надто обтяжений залежністю від психологічних, ринкових, соціокультурних та інших мінливих факторів, що важко піддаються моделюванню.

Для експлікації проектування програмних систем у першому наближенні потрібно почати з того єдиного принципу, що може вважатися безсумнівним: будь-які артефакти людської діяльності (твори мистецтва, наукові теорії, політичні доктрини, в тому числі й проекти програмних систем) мають композиційну природу. По-перше (так би мовити, у просторовому вимірі) композиційна платформа дозволяє розглядати сутність як певне поєднання інших сутностей, по-друге, що більш важливо, в часовому вимірі дозволяє бачити процес її становлення як покрокове введення та виключення композицій, а по-третє – власну роль та призначення сутності дозволяє розглядати з точки зору її включеності в ту чи іншу композицію.

Відповідно до загальних принципів композиційного, експлікативного програмування та сутнісної платформи, експлікувати певний клас сутностей – значить описати характерний для нього клас композицій, тобто виділити ті композиції, що визначають прагматико-обумовлену суть цих сутностей, задають їх сутесутнісні відношення. Домовимося композиції проектного рівня, щоб відрізняти їх від композицій рівня програмної реалізації, називати макрокомпозиціями.

Не намагаючись охопити всі визначальні властивості макрокомпозицій, зосередимо увагу на одній з них, а саме ациклічності. Дійсно, цикли в програмуванні хоча й відіграють надзвичайно важливу роль, але зосереджені переважно на мікрорівні (на рівні конкретних алгоритмів) і тим самим інкапсульовані з точки зору проектного рівня. Проект як абстраговане відображення складових програмної системи та зв'язків між ними має вигляд або ієрархічної моделі (відношення надоб'єкт-підоб'єкт, надклас-підклас тощо), або однонаправленого, без зворотних ланок, потоку передачі даних між підсистемами (наприклад, діаграма послідовностей в мові UML).

Зазначимо: беручи за основу розгляду ациклічність проектів, ми не відкидаємо і не ігноруємо цикли як такі, а замість цього вказуємо їх місце. З великого досвіду формалізації семантики програм [4, 6] відомо, що будь-яка циклічна (чи хоча б потенційно циклічна) сутність має незрівнянно складніші властивості, ніж ациклічна. Проекти, за самим своїм призначенням, мають бути фундаментом для створення системи, яка сама

має високий рівень складності. Але ж немає сенсу в якості фундаменту для цілого класу складних систем брати сутності, які через власну складність в свою чергу потребували б фундаменту.

Оскільки для самих по собі проектів важно запропонувати просту та водночас точну логіко-математичну модель, для поточної задачі дослідження властивостей ациклічних композицій залучимо допоміжні сутності – програми, що не містять циклів. Це дозволить застосувати добре розроблений формальний апарат композиційного програмування (КП). Виходячи з наведених міркувань введено поняття ациклічної функції, обґрунтовано його адекватність для відображення передач керування та даних між підсистемами великої системи, розглянуто близькі за сенсом поняття послідовно-паралельних та ланцюгових функцій, вирішено питання про взаємну звідність цих трьох класів функцій. Також показано, що клас ациклічних функцій замкнений відносно операцій множення і накладання, які становлять в КП моделі послідовного та паралельного виконання програм.

У даній роботі покажемо, що клас ациклічних функцій замкнений також відносно суперпозиції, та виявимо зв'язки цієї властивості з властивостями, дослідженими раніше.

Основні поняття і означення композиційного програмування

Тотожне відображення множини X позначатимемо $\mathrm{id}_X = \{(x,x) \mid x \in X\}$, область визначення та область значень функції f – $\mathrm{dom}\, f$ та $\mathrm{rang}\, f$ відповідно. X^Y позначаємо множину всіх відображень множини Y в множину X. $\alpha \mid U$ позначено обмеження бінарного відношення α на множину U. Проекцію відношення ρ по k-й компоненті позначаємо $\mathrm{pr}_k\, \rho$.

Відповідно до композиційного програмування (КП) [1, 2] і розробленої на його основі сутнісної платформи [3], дані моделюються іменними множинами (ІМ), а програми – іменними функціями (ІФ). Позначимо $\mathbf{V}$ і $\mathbf{D}$ множину всіх імен і денотатів відповідно. V-іменною множиною називають скінченне функціональне бінарне відношення $\alpha \in \mathbf{D}^V$, $V \subseteq \mathbf{V}$. Клас всіх іменних множин позначимо $\mathbf{N}$.

Операція накладання ІМ визначається так:

$$\alpha \nabla \beta = \beta \cup \{(u,d) \mid (u,d) \in \alpha, u \notin \mathrm{pr}_1(\beta)\}.$$

Можна довести асоціативність накладання.

За означенням, ІФ – це унарна часткова функція вигляду $f : \mathbf{N} \xrightarrow{\sim} \mathbf{N}$. ІФ f називається V-арною, якщо $\mathbf{dom}\, f \subseteq \mathbf{D}^V$, і (V,W)-арною, якщо до того ж $\mathbf{rang}\, f \subseteq \mathbf{D}^W$, $V, W \subseteq \mathbf{V}$. Не кожна іменна функція має визначену арність. Функція, яка має деяку арність, називається поліарною. Всюди далі розглядаються лише поліарні іменні функції.

Говорячи про еквівалентність ІФ, всюди маємо на увазі еквівалентність за поведінкою, яка полягає у тому, що на однакових аргументах дві ІФ або мають однаковий результат, або разом невизначені:

$$\forall \alpha \in \mathbf{N}\ f(\alpha) \simeq g(\alpha).$$

Операції над ІФ називають композиціями. Змістовно, композиції моделюють збирання складної програми з більш простих програм (підпрограм). Для подальшого розгляду потрібні дві композиції: множення ІФ (відповідає послідовному виконанню) та накладання ІФ (відповідає паралельному виконанню), означених наступним чином:

$$(f \circ g)(\alpha) \simeq g(f(\alpha)),$$
$$(f \nabla g)(\alpha) \simeq f(\alpha) \nabla g(\alpha).$$

Слід звернути увагу на те, що в останній формулі знаком ∇ позначено операцію над ІФ у лівій частині й операцію над ІМ у правій; це не повинно призводити до непорозумінь, оскільки з контексту завжди ясно, який з двох сенсів мається на увазі.

Нехай ІФ f та g – відповідно (U_1,V_1)- та (U_2,V_2)-арні. Тоді ІФ $f \circ g$ є (U_1,V_2)-арною, якщо $V_1 = U_2$, та всюди невизначеною в іншому випадку. ІФ $f \nabla g$ є $(U, V_1 \cup V_2)$-арною, якщо $U_1 = U_2 = U$, інакше – всюди невизначеною.

З останнього видно, що для застосування тих чи інших композицій до наперед заданих ІФ потрібно узгодити їх арності. Для цього існують описані далі допоміжні функції.

Обмеженням назвемо параметричну $(V, U \cap V)$-арну ІФ $\uparrow_U^V$ (де $U, V \subseteq \mathbf{V}$), таку, що, при $\alpha \in \mathbf{D}^V$,

$$\uparrow_U^V(\alpha) = \alpha \mid U = \{(u,d) \mid (u,d) \in \alpha \wedge u \in U\}.$$

Перейменуванням назвемо параметричну (V,U)-арну ІФ $\updownarrow^{\xi}$, де $U,V \subseteq \mathbf{V}$; $\xi : U \to V$, таку, що, при $\alpha \in \mathbf{D}^{V}$,

$$\updownarrow^{\xi}(\alpha) = \{(u, \alpha(\xi(u))) \mid u \in U\}.$$

Змістовно, ця операція заміняє в ІМ α імена $v \in V$ на нові імена $u \in U$, такі, що $\xi(u) = v$. Відзначимо, що взаємна однозначність відображення ξ не вимагається, тобто одне ім'я з α може після перейменування перейти в декілька нових імен. Безпосередньо з означення слідує, що якщо $\mathrm{pr}_1 \alpha = V \supseteq \mathrm{pr}_2 \xi = W$, то $\left(\uparrow_W^V \circ \updownarrow^{\xi}\right)(\alpha) = \xi \circ \alpha$, де $\circ$ позначено множення унарних функцій (слід мати на увазі, що ІМ α саме є функцією, що іменам ставить у відповідність денотати).

Ациклічні функції та їх властивості

Нагадаємо означення. Нехай програма складається з n підпрограм f_i ($i = 1,\ldots,n$). Вхід програми в цілому розглядається як вихід її уявної 0-ї підпрограми; дуальним чином, вихід ациклічної програми розглядається як вхід її уявної $(n+1)$-ї підпрограми. Характеристична особливість ациклічних програм полягає у тому, що відношення „діяти після", або „використовувати дані від" є частковим порядком на множині підпрограм. Цей порядок можна доповнити до лінійного: для всіх i, j, де $0 \le i < j \le n+1$, канал передачі даних з виходу i-ї підпрограми на вхід j-ї існує тоді й тільки тоді, коли $i < j$.

Припустимо, функції f_i мають арність (U_i, V_i) при $i = 1,\ldots,n$. Нехай дана множина імен V, яку також будемо позначати V_0, та множина імен U, яку також позначимо U_{n+1}. Нехай для всіх i, j, де $0 \le i < j \le n+1$, дано відображення ξ_{ij}.

Сукупність цих відображень можна розглядати як трикутну матрицю Ξ, що має $n+1$ рядок і стільки ж стовпчиків, причому нумерація рядків починається з 0, а стовпчиків – з 1. Накладаються обмеження:

$$V_i \supseteq \bigcup_{j=i+1}^{n+1} \mathrm{pr}_2(\xi_{ij}),\ i = \overline{0,n},$$

$$U_j = \bigcup_{i=0}^{j-1} \mathrm{pr}_1(\xi_{ij}),\ j = \overline{1,n+1}.$$

Тоді, за означенням, ациклічна ІФ в базисі Ф – це (V,U)-арна ІФ $f = \mathrm{T}^{\Xi}_{V,U}(f_1,\ldots f_n)$, така, що $f_1,\ldots,f_n \in \Phi$, та для будь-якої V-іменної множини α значення $\beta = f(\alpha)$ визначається за наступними рекурентними співвідношеннями, де $\alpha_0 = \alpha$, $\beta = \beta_{n+1}$:

$$\beta_k = \nabla_{i=0}^{k-1}(\xi_{ik} \circ \alpha_i), \text{ при } k = \overline{1,n+1},$$

$$\alpha_k = f_k(\beta_k), \text{ при } k = \overline{1,n}.$$

Зазначимо, що α_i при $0 \le i \le n$ є значенням функції f_i (змістовно, результатом роботи i-ї підпрограми), а β_i при $1 \le i \le n+1$ є аргументом функції f_i.

У попередніх роботах встановлено низку властивостей ациклічних функцій.

Твердження 1. Скінченний добуток, тобто ІФ вигляду $f = f_1 \circ \ldots \circ f_n$ можна представити як ациклічну ІФ в базисі $\{f_1,\ldots,f_n\}$. Справді, нехай ІФ f_i має арність (U_i, V_i) для всіх $i = 1,\ldots,n$. Перепозначимо U_1 через V_0. Тоді за умови узгоджених арностей $f = \mathrm{T}^{\Xi}_{V_0,V_n}(f_1,\ldots,f_n)$ при

$$\Xi = \begin{bmatrix} \mathrm{id}_{V_0} & \varnothing & \dots & \varnothing \\ & \mathrm{id}_{V_1} & \dots & \varnothing \\ & & \ddots & \vdots \\ & & & \mathrm{id}_{V_n} \end{bmatrix}.$$

Твердження 2. Скінченне накладання, тобто ІФ вигляду $f = f_1 \nabla \dots \nabla f_n$ можна представити як ациклічну ІФ в базисі $\{f_1,\dots,f_n\}$. Справді, нехай ІФ f_i має арність (U,V_i) для всіх $i = 1,\dots,n$. Позначимо $V = \bigcup_{i=1}^{n} V_i$. Тоді $f = \mathrm{T}^{\Xi}_{U,V}(f_1,\dots,f_n)$ при

$$\Xi = \begin{bmatrix} \mathrm{id}_U & \mathrm{id}_U & \dots & \mathrm{id}_U & \varnothing \\ & \varnothing & \dots & \varnothing & \mathrm{id}_{V_1} \\ & & \ddots & \vdots & \vdots \\ & & & \varnothing & \mathrm{id}_{V_{n-1}} \\ & & & & \mathrm{id}_{V_n} \end{bmatrix}.$$

Твердження 3. Добуток $f \circ g$ довільних ациклічних функцій $f = \mathrm{T}^{\Xi}_{V,U}(f_1,\dots,f_m)$ та $g = \mathrm{T}^{\Psi}_{U,W}(g_1,\dots,g_n)$ може бути перетворений на еквівалентну ациклічну функцію в базисі $\{f_1,\dots,f_m,g_1,\dots,g_n\}$.

Твердження 4. Накладання $f \nabla g$ довільних ациклічних ІФ $f = \mathrm{T}^{\Xi}_{U,V}(f_1,\dots,f_m)$ та $g = \mathrm{T}^{\Psi}_{U,W}(g_1,\dots,g_n)$ може бути перетворене на еквівалентну ациклічну функцію в базисі $\{f_1,\dots,f_m,g_1,\dots,g_n\}$.

Ациклічність суперпозиції ациклічних функцій

Основна задача даної роботи полягає у доведенні наступної властивості.

Теорема. Нехай дано дві ациклічні ІФ, $f = \mathrm{T}^{\Xi}_{U,V}(f_1,\dots,f_m)$ та $g = \mathrm{T}^{\Psi}_{X,Y}(g_1,\dots,g_n)$, дано таке число k, що $1 \le k \le m$, причому функція f_k є (X,Y)-арною. Тоді функція h, отримана підстановкою функції g замість базисної функції f_k, тобто функція

$$h = \mathrm{T}^{\Xi}_{U,V}\left(f_1,\dots f_{k-1}, \mathrm{T}^{\Psi}_{X,Y}(g_1,\dots,g_n), f_{k+1},\dots,f_m\right),$$

може бути представлена у вигляді

$$h = \mathrm{T}^{\mathrm{Z}}_{U,V}\left(f_1,\dots f_{k-1}, g_1,\dots,g_n, f_{k+1},\dots,f_m\right)$$

при деякій матриці перейменувань Z.

Позначимо кількість базисних функцій h_i функції h: $r = m + n - 1$. Тепер перепишемо співвідношення з означень ациклічної функції для кожної з функцій f, g, h:

$$f(\alpha_0) = \beta_{m+1}, \tag{1}$$

$$\beta_i = \prod_{j=0}^{i-1} (\xi_{ji} \circ \alpha_j) \text{ при } 1 \le i \le m+1, \tag{2}$$

$$\alpha_i = f_i(\beta_i) \text{ при } 1 \le i \le m, \tag{3}$$

$$g(\gamma_0) = \delta_{n+1}, \tag{4}$$

$$\delta_i = \prod_{j=0}^{i-1} (\psi_{ji} \circ \gamma_j) \text{ при } 1 \le i \le n+1, \tag{5}$$

$$\gamma_i = g_i(\delta_i) \text{ при } 1 \le i \le n, \tag{6}$$

$$h(\mu_0) = \nu_{r+1}, \tag{7}$$

$$\nu_i = \prod_{j=0}^{i-1} (\zeta_{ji} \circ \mu_j) \text{ при } 1 \le i \le r+1, \tag{8}$$

$$\mu_i = h_i(\nu_i) \text{ при } 1 \le i \le r\,. \tag{9}$$

Аргумент і значення функції f ототожнимо з аргументом і значенням функції h:

$$\alpha_0 = \mu_0\,, \tag{10}$$
$$\beta_{m+1} = \nu_{r+1}\,. \tag{11}$$

Крім того, випишемо умову, що аргумент та значення функції f_k ототожнюються з аргументом та значенням функції g:

$$\gamma_0 = \beta_k\,, \tag{12}$$
$$\delta_{n+1} = \alpha_k\,. \tag{13}$$

Тоді для базисних функцій h_i, їх аргументів та значень маємо такі означення:

$$h_i = \begin{cases} f_i, & 1 \le i \le k-1, \\ g_{i-k+1}, & k \le i \le k+n-1, \\ f_{i-n+1}, & k+n \le i \le m+n-1. \end{cases} \tag{14}$$

$$\mu_i = \begin{cases} \alpha_i, & 0 \le i \le k-1, \\ \gamma_{i-k+1}, & k \le i \le k+n-1, \\ \alpha_{i-n+1}, & k+n \le i \le m+n-1, \end{cases} \tag{15}$$

$$\nu_i = \begin{cases} \beta_i, & 1 \le i \le k-1, \\ \delta_{i-k+1}, & k \le i \le k+n-1, \\ \beta_{i-n+1}, & k+n \le i \le m+n. \end{cases} \tag{16}$$

Тепер розглянемо аргументи та значення функцій h_i. Можливі три випадки.

Випадок 1. $1 \le i \le k-1$. Оскільки, згідно (14, 15, 16), $h_i = f_i$, $\mu_i = \alpha_i$ та, $\nu_i = \beta_i$, з огляду на (10, 11), єдино можливим значенням ζ_{ji} при $0 \le j \le i-1$ є

$$\zeta_{ji} = \xi_{ji}\,. \tag{17}$$

Випадок 2. $k \le i \le k+n-1$. Згідно (16), аргументом ν_i функції h_i є

$$\nu_i = \delta_{i-k+1} = \sum_{j=0}^{i-k} \psi_{j,i-k+1} \circ \gamma_j\,,$$

тобто

$$\nu_i = \left(\psi_{0,i-k+1} \circ \gamma_0\right) \nabla \left(\sum_{j=1}^{i-k} \psi_{j,i-k+1} \circ \gamma_j \right).$$

Маючи на увазі (12), підставимо вираз (9) для β_k у перший член накладання, а в другому, згідно (15), виразимо γ_j ($1 \le j \le i-k$) через μ_j ($k \le j \le i-1$):

$$\nu_i = \left(\psi_{0,i-k+1} \circ \sum_{j=0}^{k-1} \xi_{jk} \circ \alpha_j \right) \nabla \left(\sum_{j=k}^{i-1} \psi_{j-k+1,i-k+1} \circ \mu_j \right).$$

Тепер в першому члені замінимо α_j на μ_j згідно (15), а спільний множник внесемо до оператора накладання:

$$\nu_i = \left(\sum_{j=0}^{k-1} \psi_{0,i-k+1} \circ \xi_{jk} \circ \mu_j \right) \nabla \left(\sum_{j=k}^{i-1} \psi_{j-k+1,i-k+1} \circ \mu_j \right).$$

Порівнявши отриманий вираз з (8), робимо висновок:

$$\zeta_{ji} = \begin{cases} \psi_{0,i-k+1} \circ \xi_{jk}, & 0 \le j \le k-1, \\ \psi_{j-k+1,i-k+1}, & k \le j \le i-1. \end{cases} \tag{18}$$

Випадок 3. $k+n \le i \le m+n$. Згідно (16), аргумент ν_i функції h_i є

$$\nu_i = \left(\bigvee_{j=0}^{k-1} \xi_{j,i-n+1} \circ \alpha_j \right) \nabla \left(\xi_{k,i-n+1} \circ \alpha_k \right) \nabla \left(\bigvee_{j=k+1}^{i-n} \xi_{j,i-n+1} \circ \alpha_j \right).$$

У першому члені накладання можна замінити α_j на μ_j за (15), у другому члені, згідно (13), підставимо δ_{n+1} замість α_k, а в третьому члені α_j ($k+1 \le j \le i-n$) можна замінити на μ_j ($k+n \le j \le i-1$):

$$\nu_i = \left(\bigvee_{j=0}^{k-1} \xi_{j,i-n+1} \circ \mu_j \right) \nabla \left(\xi_{k,i-n+1} \circ \delta_{n+1} \right) \nabla \left(\bigvee_{j=k+n}^{i-1} \xi_{j-n+1,i-n+1} \circ \mu_j \right). \tag{19}$$

Розглянемо окремо множник δ_{n+1} з другого члена цього виразу. Застосувавши його означення (9), винесемо перший доданок з оператора накладання:

$$\delta_{n+1} = \bigvee_{j=0}^{n} \psi_{j,n+1} \circ \gamma_j = \left(\psi_{0,n+1} \circ \gamma_0 \right) \nabla \left(\bigvee_{j=1}^{n} \psi_{j,n+1} \circ \gamma_j \right).$$

З γ_0 тут можна обійтися таким же чином, як і у випадку 2, тобто замінити на означення β_k (2), після чого всі α_j та γ_j замінити на відповідні μ_j згідно (15):

$$\delta_{n+1} = \left(\psi_{0,n+1} \circ \bigvee_{j=0}^{k-1} \xi_{jk} \circ \alpha_j \right) \nabla \left(\bigvee_{j=1}^{n} \psi_{j,n+1} \circ \gamma_j \right),$$

тобто

$$\delta_{n+1} = \left(\bigvee_{j=0}^{k-1} \psi_{0,n+1} \circ \xi_{jk} \circ \mu_j \right) \nabla \left(\bigvee_{j=k}^{k+n-1} \psi_{j-k+1,n+1} \circ \mu_j \right),$$

Підставивши це у (19), отримуємо:

$$\nu_i = \left(\bigvee_{j=0}^{k-1} \left(\left(\xi_{j,i-n+1} \nabla \xi_{k,i-n+1} \circ \psi_{0,n+1} \circ \xi_{jk} \right) \circ \mu_j \right) \right) \nabla \left(\bigvee_{j=k}^{k+n-1} \xi_{k,i-n+1} \circ \psi_{j-k+1,n+1} \circ \mu_j \right) \nabla \left(\bigvee_{j=k+n}^{i-1} \xi_{j-n+1,i-n+1} \circ \mu_j \right).$$

Порівнюючи цей вираз з (8), доходимо висновку, що при $k+n \le i \le m+n$

$$\zeta_{ji} = \begin{cases} \xi_{j,i-n+1} \nabla \xi_{k,i-n+1} \circ \psi_{0,n+1} \circ \xi_{jk}, & 0 \le j \le k-1, \\ \xi_{k,i-n+1} \circ \psi_{j-k+1,n+1}, & k \le j \le k+n-1, \\ \xi_{j-n+1,i-n+1}, & k+n \le j \le i-1. \end{cases} \tag{20}$$

Таким чином, вирази (17, 18, 20) повністю визначають елементи матриці Z.

Зазначимо, що представити суперпозицію функції g замість функції f_k у вигляді ациклічної функції можна й іншим способом, з простішим виглядом матриці перейменувань, але ціною неістотного розширення базису двома допоміжними функціями, а саме:

$$h = \mathrm{T}^{Z}_{U,V}\left(f_1, \ldots f_{k-1}, \mathrm{id}_{\mathbf{D}^X}, g_1, \ldots, g_n, \mathrm{id}_{\mathbf{D}^Y}, f_{k+1}, \ldots, f_m \right),$$

де елементи матриці Z визначаються за такими правилами:

- $\zeta_{ji} = \xi_{ji}$ при $0 \le j < i \le k$;
- $\zeta_{ji} = \psi_{j-k,i-k}$ при $k \le j < i \le k+n+1$;
- $\zeta_{ji} = \xi_{j-n-1,i-n-1}$ при $k+n+1 \le j < i \le m+n+2$;
- $\zeta_{ji} = \xi_{j,i-n-1}$ при $0 \le j \le k-1$ та $k+n+2 \le i \le m+n+2$.

- В усіх інших випадках $\zeta_{ji} = \varnothing$.

Іншими словами, якщо матрицю Ξ представити в блочній формі

$$\Xi = \begin{bmatrix} \Xi_{01} & \Xi_{02} \\ & \Xi_{12} \end{bmatrix},$$

де матриці Ξ_{01} та Ξ_{12} верхньо-трикутні, а Ξ_{02} – прямокутна, причому Ξ_{01} містить k рядків і стовпчиків, то матриця Z має вигляд

$$Z = \begin{bmatrix} \Xi_{01} & \varnothing & \Xi_{02} \\ & \Psi & \varnothing \\ & & \Xi_{12} \end{bmatrix}.$$

Таким чином, клас ациклічних функцій замкнений відносно суперпозиції. Варто уваги, що встановлені раніше властивості замкненості цього класу відносно множення та накладання (твердження 3 та 4) виявляються наслідками даної властивості.

Справді, нехай дано ациклічні функції f та g. Згідно твердження 1, ациклічною у базисі $\{h_1, h_2\}$ є функція $h = h_1 \circ h_2$. За теоремою, ациклічною буде функція, отримана з неї підстановкою f замість h_1 та g замість h_2. Такі ж міркування справедливі для накладання.

Висновок

У контексті основної проблематики – експлікації проектів програмних систем – зазначені властивості підтверджують адекватність обраних засобів розкриття логіки та структури проектів. Дійсно, якщо до ациклічної композиції підставити ті чи інші аргументи – підпрограми f_i $(i = 1, \ldots, n)$, буде отримано деяку конкретну програму; якщо ж в якості аргументів підставити інші проекти, результатом підстановки буде знову проект. Іншими словами, подібно до того, як чиста логіка предикатів першого порядку є спільною основою для необмеженого класу прикладних логік, параметризація запропонованої нами моделі базисними функціями дає деякі прикладні програмування, тоді як без параметризації вона являє собою „чисту", абстраговану від змісту, форму, спільну для цих програмувань.

1. *Редько В.Н.* Композиции программ и композиционное программирование // Программирование. – 1978. – № 5. – С. 3-24.
2. *Редько В.Н.* Композиционная структура программологии // Кибернетика и системный анализ. – 1998. – № 4. – С. 47-66.
3. *Редько В.Н., Редько И.В., Гришко Н.В.* Программологические основания сущностной платформы // Проблеми програмування. – 2008. – № 2-3 (Спец. випуск). – С. 75-83.
4. *Hoare, C.A.R., Jifeng, He.* Unifying Theories of Programming. – Prentice Hall Europe. – 1998 – 298 p.
5. *Кауфман В.Ш.* Языки программирования. Концепции и принципы. – М.: Радио и связь, 1993. – 430 с.
6. *Дейкстра Э.* Дисциплина программирования. – М.: Мир, 1978. – 280 с.
7. *Басараб И.А., Никитченко Н.С., Редько В.Н.* Композиционные базы данных. – К.: Либідь, 1992. – 192 с.
8. *Parfirova T., Vinnyk V* Compositional Model of Acyclic Programs // CSE'2010 International Scientific Conference on Computer Science and Engineering, September 20-22, 2010, Košice - Stará Ľubovňa, Slovakia.
9. *Parfirova T., Vinnyk V* Some properties of acyclic compositional programs // Information Models of Knowledge. – ITHEA № 19. – P. 460-464.
10. *Парфірова Т.С., Вінник В.Ю.* Семантичні структури програм без циклів // Матеріали 7-ї Міжнародної конференції «Теоретичні та прикладні аспекти побудови програмних систем» – TAAPSD'2010, Київ, Україна, 4-8 жовтня 2010 р. – С. 378-384.