

DOI: 10.20535/2411-1031.2018.6.2.153491

УДК 004(94+41)

ВОЛОДИМИР СОКОЛОВ

**СТРУКТУРНИЙ АНАЛІЗ СПОЛУК ІНТЕГРАЛЬНИХ ОБ'ЄКТІВ**

У статті представлено результати досліджень методів аналізу структури програмного забезпечення, побудованого за архітектурою на основі інтегральних об'єктів. Для визначення операцій структурного аналізу обрано представлення сполук інтегральних об'єктів у вигляді формул сполук та таблиць, а також формалізовані операції інтеграції та дезінтеграції сполук. Інтеграція сполуки з утворенням нового класу полягає в інкапсуляції сполуки в оболонку класу з присвоєнням степені інтеграції класу на одиницю більше, ніж у сполуки, а дезінтеграція полягає у зворотному процесі видалення сполуки з оболонки класу зі зменшенням степені інтеграції на одиницю. Табличне представлення сполук об'єктів, що включає як представлення об'єктів, так і зв'язків між ними, дозволяє застосувати формальні методи аналізу їх структури. Визначено поняття структурної еквівалентності двох сполук як таких, що мають однаковий склад та структуру з'єднання об'єктів, а також поняття функціональної еквівалентності двох сполук як таких, що реалізують однакову композицію функцій. В якості основних операцій структурного аналізу представлено дві базові операції – визначення структурної та функціональної еквівалентності двох сполук. Визначення еквівалентності сполук зводиться до порівняння табличного представлення сполук з урахуванням перестановки рядків та стовпчиків таблиць. Показано, що визначення структурної еквівалентності можливо лише для сполук однакової степені інтеграції, а для визначення функціональної еквівалентності потрібно приведення сполук до атомарного рівня. Визначення функціональної еквівалентності зведено до визначення структурної еквівалентності атомарних формул сполук. В якості похідних операцій виділено операції розпізнавання структурно та функціонально еквівалентних класів для заданої сполуки, які ґрунтуються на базових операціях визначення еквівалентності, а також чотири операції еквівалентних перетворень сполук: структурно та функціонально еквівалентна заміна сполуки на об'єкт еквівалентного класу та часткова структурно та функціонально еквівалентна заміна фрагменту сполуки на об'єкт еквівалентного класу. Операції структурного аналізу є основою для еквівалентних перетворень сполук в процесі проектування, верифікації та оптимізації системи класів розв'язку задач предметної області під час розробки програмного забезпечення.

**Ключові слова:** структурний аналіз, структурна еквівалентність, функціональна еквівалентність, інтегральний об'єкт, сполуки інтегральних об'єктів.

**Постановка проблеми.** Однією з центральних проблем теорії програмування є проблема визначення еквівалентності та еквівалентних перетворень алгоритмів та програм під час їх розробки, проектування та верифікації [1] - [5]. Наприклад, в теорії схем програм використовуються графові та алгебраїчні моделі для дослідження схем процедурних програм, однак проблема до кінця не вирішена. Крім того, в об'єктно-орієнтованому програмуванні взагалі немає формального апарату для опису та проектування системи класів, який би запобігав дублюванню структурно та функціонально однакових класів та дозволяв би досліджувати отримані програмні структури. Основною причиною цього є відсутність повноцінних формальних методів опису об'єктно-орієнтованих програм.

Однак при використанні архітектури програмного забезпечення на основі інтегральних об'єктів [6] така можливість існує. Тому для аналізу таких програм потрібно, по-перше,

у визначити способи їх формального опису, придатні не тільки для формального аналізу, а й для застосування алгоритмів, а, по-друге, визначити базовий набір операцій структурного аналізу.

**Аналіз останніх досліджень та публікацій.** Основні ідеї структурного аналізу програм були досліджені в рамках теорії схем програм та об'єктно-орієнтованого проектування. Сучасні дослідження в більшості стосуються окремих аспектів структурного аналізу процедурних програм та формалізації представлення класів в об'єктно-орієнтованому програмуванні. Наприклад, в [1] - [3] розглядаються алгебраїчні моделі програм з процедурами та будується система еквівалентних перетворень для їх структурного аналізу. В [4] запропоновано підхід для визначення ступеню еквівалентності алгоритмів в умовах невизначеності за допомогою операторного методу. Зміст цієї статті оснований на формальному представленні програм, створених за архітектурою на основі сполук інтегральних об'єктів [5], [6].

**Метою статті** є створення формального апарату структурного аналізу схем програм, побудованих за архітектурою на основі інтегральних об'єктів.

**Виклад основного матеріалу дослідження.**

**Представлення сполук.** Як відомо з [5], сполуки можна представляти у вигляді функціональної моделі, реляційної моделі та графічної схеми. Однак для структурного аналізу більше підходить представлення сполук у вигляді формул та таблиць. Наприклад, розглянемо сполуку  $S$ , що задана графічною схемою (див. рис.1) та наступною функціональною моделлю:

$$S = \langle Cls, Obj, Con, Inp, Out \rangle, \tag{1}$$

- Де  $Cls = \{A : x, y \rightarrow z, B : x, y \rightarrow z\}$  – множина формул класів;  
 $Obj = \{A :: o_1, A :: o_2, B :: o_3\}$  – множина об'єктів;  
 $Con = \{o_3.x = o_1.z, o_3.y = o_2.z\}$  – схема з'єднання об'єктів;  
 $Inp = \{o_1.x, o_1.y, o_2.x, o_2.y\}$  – входи сполуки;  
 $Out = \{o_3.z\}$  – виходи сполуки.

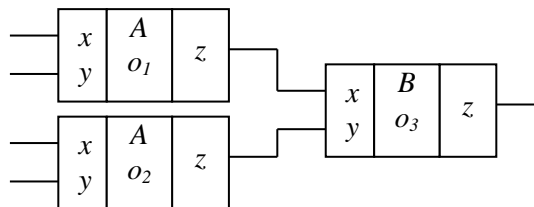


Рисунок 1 – Графічна схема сполуки  $S$

Представлення сполуки  $S$  у вигляді формул має наступний вигляд.

$S = A_2B$  – формула сполуки, що визначає склад сполуки;

$S : o_1.x, o_1.y, o_2.x, o_2.y \rightarrow o_3.z$  – валентна формула сполуки;

$S : o_1.x, o_1.y, o_2.x, o_2.y \{o_3.x = o_1.z, o_3.y = o_2.z\} \rightarrow o_3.z$  – структурна формула;

$S = A_2B : o_1.x, o_1.y, o_2.x, o_2.y \{o_3.x = o_1.z, o_3.y = o_2.z\} \rightarrow o_3.z$  – повна об'єктна формула;

$S = A_2B : A[1](x, y), A[2](x, y) \{B[3]x = A[1]z, B[3]y = A[2]z\} \rightarrow B[3]z$  – повна формула.

Відповідно до [6] кожний інтегральний клас містить дескриптор, який можна представити відповідно до (1) або з використанням формул наступним чином (для прикладу класу  $C$ , наведеного на рис. 2):

$$D_C = \langle S_C, F_C \rangle,$$

де  $S_C$  – формула внутрішньої сполуки, що інтегрована в класі:

$$S_C = A_2B : A[1](x, y), A[2](x, y)\{B[3]x = A[1]z, B[3]y = A[2]z\} \rightarrow B[3]z$$

$F_C$  – структурна формула класу з урахуванням з’єднання внутрішньої сполуки з входами та виходами класу:

$$F_C = C^1 : x_1, y_1, x_2, y_2 \{A[1]x = x_1, A[1]y = y_1, A[2]x = x_2, A[2]y = y_2, z = B[3]z\} \rightarrow z$$

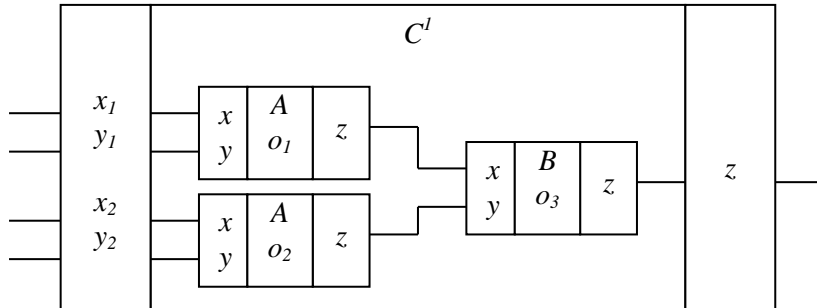


Рисунок 2 – Схема інтегрального класу  $C$

Таке представлення інтегральних класів дозволяє формалізувати операції інтеграції та дезінтеграції. Наприклад, формула інтеграції сполуки  $S$  (рис. 1) з утворенням нового інтегрального класу  $C$  може бути представлена як

$$C^1 : x_1, y_1, x_2, y_2 \rightarrow z = I(S)\{A[1]x = x_1, A[1]y = y_1, A[2]x = x_2, A[2]y = y_2, z = B[3]z\}$$

Дезінтеграція класу  $C$  на одну степінь є виведенням сполуки зі складу класу з утворенням сполуки  $S$  і може бути представлено формулою

$$S = dI(C)$$

або з урахуванням степені інтеграції

$$S^{n-1} = dI(C^n)$$

Дезінтеграція класу  $C^n$  на  $m$  степенів можна записати

$$S^{n-m} = dI^m(C^n)$$

Дезінтеграція сполуки полягає у дезінтеграції кожного інтегрального об’єкту сполуки на одну степінь.

Повна дезінтеграція сполуки або інтегрального класу  $C$  до атомарного рівня сполуки  $S^0$  може бути представлена формулою

$$S^0 = dI^0(C)$$

Крім того, атомарну формулу сполуки будемо позначати

$$dI^0(C) = A(C)$$

Повна формула сполуки разом з формулами класів містить вичерпні дані про сполуку, але для формальних методів аналізу найбільш підходить наступне табличне представлення: заголовки стовпчиків містять атрибути об’єктів сполуки, а заголовки рядків – назви класів об’єктів або признак з’єднання (позначено символом “=”), при цьому один рядок відповідає одному об’єкту або одному з’єднанню. Елементи таблиці містять “-1”, якщо атрибут є входом об’єкту або з’єднання, “1”, якщо він є виходом об’єкту або з’єднання, а решта дорівнює “0” (можна не вказувати). Наприклад, для сполуки  $S$  (див. рис. 1) табличне представлення наведене в табл. 1.

Таблиця 1 – Табличне представлення сполуки  $S$

<b>S</b>	x	y	z	x	y	z	x	y	z
<b>A</b>	-1	-1	1						
<b>A</b>				-1	-1	1			
<b>B</b>							-1	-1	1
<b>=</b>			-1				1		
<b>=</b>						-1		1	

**Операції структурного аналізу сполук.** При структурному аналізі сполук найбільш суттєвими є питання структурної еквівалентності сполук, функціональної еквівалентності сполук та еквівалентні перетворення сполук.

**Визначення 1. Структурна еквівалентність сполук.** Дві сполуки  $S_1$  та  $S_2$  є структурно еквівалентними (СЕ) ( $S_1 \equiv S_2$ ), якщо вони мають однаковий склад об'єктів та структуру їх з'єднання.

Необхідною умовою СЕ сполук є еквівалентність формул сполук (склад об'єктів відповідних класів), а достатньою умовою СЕ є еквівалентність повних формул сполук. Дві сполуки різної степені інтеграції не можуть бути структурно еквівалентними внаслідок неможливості збігу класів різної степені інтеграції у формулах сполук.

Практичне визначення СЕ сполук ускладнюється різною синтаксичною послідовністю запису повних формул сполук і різним порядком нумерації або іменування об'єктів. Крім випадку елементарної СЕ двох сполук, коликожна сполука складається з одного об'єкту одного класу, загалом потрібні перетворення форми представлення сполук для визначення СЕ. Для таких перетворень, як буде показано нижче, найбільш підходить табличне представлення сполук.

**Визначення 2. Функціональна еквівалентність сполук.** Дві сполуки  $S_1$  та  $S_2$  є функціонально еквівалентними (ФЕ) ( $f(S_1) \equiv f(S_2)$ ), якщо вони реалізують одну й ту саму композицію функцій.

Із СЕ двох сполук витікає їх ФЕ, так як кожна сполука реалізує одну й ту саму композицію функційодних й тих самих класів.

$$S_1 \equiv S_2 \Rightarrow f(S_1) \equiv f(S_2)$$

Однак, якщо сполуки мають різну ступінь інтеграції і не є СЕ, вони можуть реалізувати однакову композицію функцій. Таке можливе, якщо при приведенні сполук до однакової степені інтеграції (шляхом дезінтеграції сполуки більшої степені до рівня сполуки меншої степені або шляхом дезінтеграції обох сполук включно до атомарного рівня) вони стануть СЕ. Наприклад, сполука S (див. рис.1), клас C (див. рис.2) та клас E (див. рис.3) є ФЕ.

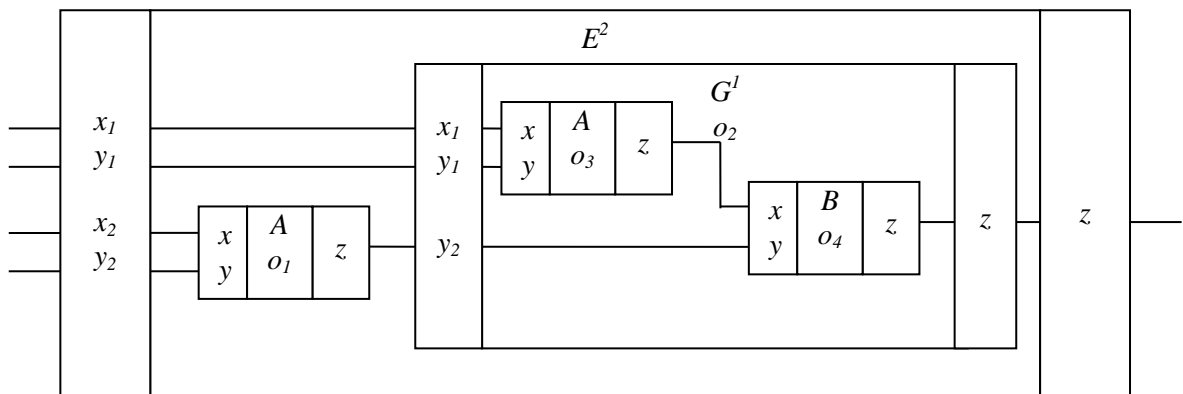


Рисунок 3 – Схема інтегрального класу E

Тобто, визначення ФЕ сполук зводиться до визначення СЕ сполук після приведення їх до одної степені інтеграції(можливо, до атомарного рівня).

Однією з властивостей ФЕ сполук є те, що при досягненні СЕ в процесі покрокової дезінтеграції сполук до певної степені всі нижчі степені інтеграції сполук також будуть СЕ.

**1. Структурна еквівалентність сполук однакової степені інтеграції.** Прийmemo, що ступінь інтеграції сполуки дорівнює максимальній степені інтеграції об'єктів сполуки.

Постановка задачі.

Дано:

1. Повні формули сполук  $S_1$  та  $S_2$ :

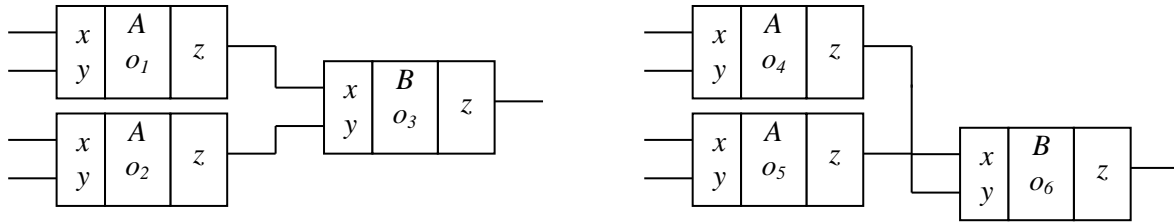
$$S_1 = A_2B : A[1](x, y), A[2](x, y)\{B[3]x = A[1]z, B[3]y = A[2]z\} \rightarrow B[3]z$$

$$S_2 = A_2B : A[4](x, y), A[5](x, y)\{B[6]x = A[5]z, B[6]y = A[4]z\} \rightarrow B[6]z$$

2. Формули класів об'єктів сполук:

$$Cls = \{A: x, y \rightarrow z, B: x, y \rightarrow z\}$$

Відповідні графічні схеми сполук представлені на рис. 4.



а) схема сполуки  $S_1$ .

б) схема сполуки  $S_2$ .

Рисунок 4 – Схеми сполук

Потрібно: визначити структурну еквівалентність сполук степені інтеграції  $n$ :  $S_1^n \equiv S_2^n$ .

Для прикладу, що розглядається,  $n=0$ .

Рішення:

1. Представляємо сполуки  $S_1$  та  $S_2$  у вигляді таблиць  $T(S_1)$  та  $T(S_2)$  (з нумерацією рядків та стовпчиків для демонстрації рішення) відповідно до повних формул.

Таблиця 1 –  $T(S_1)$

$S_1$		1	2	3	4	5	6	7	8	9
		x	y	z	x	y	z	x	y	z
1	A	-1	-1	1						
2	A				-1	-1	1			
3	B							-1	-1	1
4	=			-1				1		
5	=						-1		1	

Таблиця 2 –  $T(S_2)$

$S_2$		1	2	3	4	5	6	7	8	9
		x	y	z	x	y	z	x	y	z
1	A	-1	-1	1						
2	A				-1	-1	1			
3	B							-1	-1	1
4	=			-1					1	
5	=						-1	1		

2. Якщо перестановкою рядків та стовпчиків таблицю  $T(S_2)$  можна привести до форми таблиці  $T(S_1)$  або навпаки, то обидві сполуки є СЕ.

Таблиця 3 –  $T(S_2)$  після перестановки

$S_2$		4	5	6	1	2	3	7	8	9
		x	y	z	x	y	z	x	y	z
2	A	-1	-1	1						
1	A				-1	-1	1			
3	B							-1	-1	1
5	=			-1				1		
4	=						-1		1	

Як видно, в результаті обидві таблиці є однаковими з точністю до імен класів та атрибутів об'єктів, з чого випливає, що  $S_1^n \equiv S_2^n$ .

**2. Функціональна еквівалентність сполук різної степені інтеграції.** Постановка задачі.

Дано:

1. Повні формули сполук  $S_1$  та  $S_2$  :

$$S_1 = A_2B : A[1](x, y), A[2](x, y)\{B[3]x = A[1]z, B[3]y = A[2]z\} \rightarrow B[3]z$$

$$S_2^1 = AG^1 : A[4](x, y), G[5](x_1, y_1)\{G[5]y_2 = A[4]z\} \rightarrow G[5]z$$

Відповідні графічні схеми сполук представлені на рис. 5.

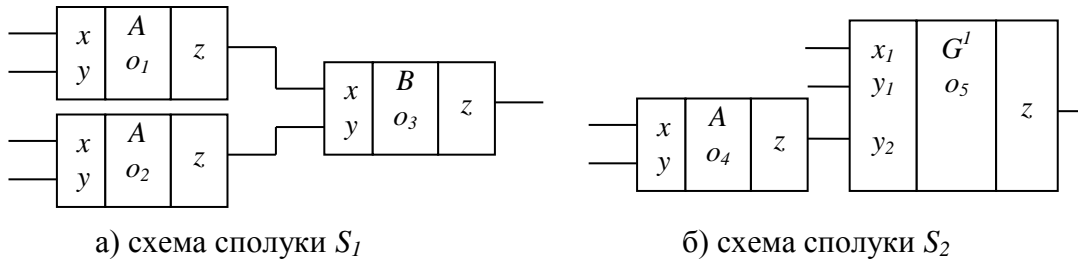


Рисунок 5 – Схеми сполук

2. Формули класів об'єктів сполук:

$$Cls = \{A : x, y \rightarrow z, B : x, y \rightarrow z, G^1 : x_1, y_1, y_2 \rightarrow z\}$$

3. Дескриптори інтегральних класів, об'єкти яких входять до складу сполук (в прикладі, що розглядається, це дескриптор інтегрального класу G):

$$D_G = \langle S_G, F_G \rangle,$$

$$S_G = AB : A[1](x_1, y_1), B[2]y_2\{B[2]x = A[1]z\} \rightarrow B[2]z$$

$$F_G = G^1 : x_1, y_1, y_2\{A[1]x = x_1, A[1]y = y_1, B[2]y = y_2, z = B[2]z\} \rightarrow z$$

Потрібно: визначити функціональну еквівалентність сполук:  $f(S_1) \equiv f(S_2)$ .

Рішення:

1. Внаслідок того, що кожна сполука реалізує композицію функцій виключно атомарних класів, то приведення сполук до однакової, але не нульової, степені інтеграції не гарантуватиме їх ФЕ при відсутності СЕ, тому потрібно виконати повну дезінтеграцію обох сполук та визначити їх СЕ:

$$dI^0(S_1) \equiv dI^0(S_2) \text{ або } A(S_1) \equiv A(S_2)$$

2. Так як сполука  $S_1$  є атомарною, виконуємо повну дезінтеграцію сполуки  $S_2$ , використовуючи дескриптор інтегрального класу  $D_G$ , атомарні формули представляємо таблицями та визначаємо СЕ  $A(S_1) \equiv A(S_2)$ .

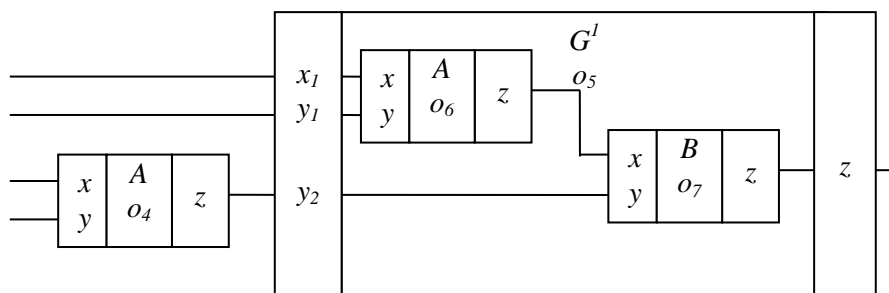


Рисунок 6 – Схеми сполуки  $S_2$  з урахуванням  $D_G$

Замінюємо сполуку  $S_2$  на її атомарну сполуку (див. рис.7).

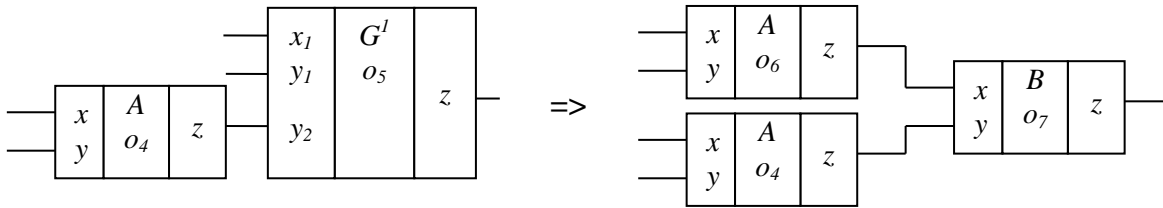


Рисунок 7 – Повна дезінтеграція сполуки  $S_2$

Формуємо таблиці сполук (табл.4, табл.5, табл.6).

Таблиця 4 – Таблиця  $T(S_1)$

$S_1$		1	2	3	4	5	6	7	8	9
		x	y	z	x	y	z	x	y	z
1	A	-1	-1	1						
2	A				-1	-1	1			
3	B							-1	-1	1
4	=			-1				1		
5	=						-1		1	

Таблиця 5 – Таблиця  $T(S_2)$

$S_2$		1	2	3	4	5	6	7
		x	y	z	$x_1$	$y_1$	$y_2$	z
1	A	-1	-1	1				
2	G				-1	-1	-1	1
3	=			-1			1	

Таблиця 6 – Таблиця  $T(S_2)$  після повної дезінтеграції

$S_2$		1	2	3	4	5	6	7	8	9
		x	y	z	x	y	z	x	y	z
1	A	-1	-1	1						
2	A				-1	-1	1			
3	B							-1	-1	1
4	=			-1				1		
5	=						-1		1	

Як видно,  $T(S_1) = T(S_2)$ , з цього випливає еквівалентність атомарних сполук і як наслідок – ФЕ сполук:

$$A(S_1) \equiv A(S_2) \Rightarrow f(S_1) \equiv f(S_2)$$

### 3. Операції розпізнавання класів сполук.

#### 3.1 . Розпізнавання (пошук) структурно еквівалентних класів (СЕК) сполуки.

Постановка задачі.

Дано:

1. Повна формула сполуки  $S$ .

2. Множина інтегральних класів  $C = \{C_i | i = \overline{1, I}\}$ , кожний з яких має дескриптор інтегрального класу  $D_i$ .

Потрібно: знайти СЕК сполуки  $S$ :

$$SEC(S, C) = \{C_j \in C | S \equiv S_j, j = \overline{1, J}, J \leq I\}$$

Рішення:

1. Знаходимо підмножину ізомernih класів  $K$ , що мають однакову формулу зі сполукою  $S$ .

$$K = \{C_k \subseteq C \mid S = S_k, k \leq I\}$$

2. Якщо  $K \neq \emptyset$ , то представляємо сполуку  $S$  та ізомерні класи  $K$  у формі таблиць та знаходимо всі СЕК.

$$SEC(S, K) \subseteq K \mid T(S) = T(S_k)$$

### 3.2. Розпізнавання (пошук) функціонально еквівалентних класів (ФЕК) сполуки.

Постановка задачі.

Дано:

1. Повна формула сполуки  $S$ .
2. Множина інтегральних класів  $C = \{C_i \mid i = \overline{1, I}\}$ , кожний з яких має дескриптор інтегрального класу  $D_i$ .

Потрібно:

Знайти ФЕКсполуки  $S$ .

$$FEC(S, C) = \{C_j \in C \mid f(S) \equiv f(S_j), j = \overline{1, J}, J \leq I\}$$

Рішення:

1. Знаходимо атомарні формули для класів  $C$  та сполуки  $S$ .
2. Знаходимо підмножину ізомерних атомарних класів  $K$ , що мають однакову формулу зі сполукою  $S$ .

$$K = \{C_k \subseteq C \mid A(S) = A(S_k), k \leq I\}$$

3. Якщо  $K \neq \emptyset$ , представляємо сполуку  $S$  та ізомери  $K$  у формі таблиць та знаходимо всі ФЕК.

$$FEC(S, K) \subseteq K \mid T(A(S)) = T(A(S_k))$$

## 4. Операції еквівалентного перетворення сполук.

### 4.1. Пошук та підстановка СЕК (СЕ заміна).

Постановка задачі.

Дано:

1. Повна формула сполуки  $S$ .
2. Множина інтегральних класів  $C = \{C_i \mid i = \overline{1, I}\}$ , кожний з яких має дескриптор інтегрального класу  $D_i$ .

Потрібно: знайти СЕКсполуки  $S$  та замінити сполуку  $S$  на об'єкт класу  $C_j$ :

$$S \leftarrow C_j :: o \mid C_j \in SEC(S, C), SEC(S, C) \neq \emptyset$$

Рішення:

1. Застосовуємо операцію 3.1 для множини класів  $C$  та замінюємо сполуку  $S$  на об'єкт СЕК  $C_j$ , якщо він знайдений. Питання вибору СЕК, якщо їх більше одного, не розглядається. Наприклад, сполуку  $S$  (див. рис.1) можна замінити на об'єкт класу  $C$  (див. рис.2).

2. Змінюємо представлення сполуки  $S$ .

### 4.2. Пошук та підстановка ФЕК (ФЕ заміна).

Постановка задачі.

Дано:

1. Повна формула сполуки  $S$ .
2. Множина інтегральних класів  $C = \{C_i \mid i = \overline{1, I}\}$ , кожний з яких має дескриптор інтегрального класу  $D_i$ .

Потрібно: знайти ФЕК сполуки  $S$  та замінити сполуку  $S$  на об'єкт класу  $C_j$ :

$$S \leftarrow C_j :: o \mid C_j \in FEC(S, C), FEC(S, C) \neq \emptyset$$

Рішення:

1. Застосовуємо операцію 3.2 для множини класів  $C$  та замінюємо сполуку  $S$  на об'єкт ФЕК  $C_j$ , якщо він знайдений. Питання вибору ФЕК, якщо їх більше одного, не розглядається. Наприклад, сполуку  $S$  (див. рис.1) можна замінити на об'єкт класу  $E$  (див. рис.3).

2. Змінюємо представлення сполуки  $S$ .



#### 4.3. Частковий пошук та підстановка СЕК (часткова СЕ заміна).

Постановка задачі.

Дано:

1. Повна формула сполуки  $S$  та її виділений фрагмент у вигляді сполуки  $T$ .

2. Множина інтегральних класів  $C = \{C_i | i = \overline{1, I}\}$ , кожний з яких має дескриптор інтегрального класу  $D_i$ .

Потрібно: знайти СЕК фрагменту  $T$  сполуки  $S$  та замінити його на об'єкт класу  $C_j$ :

$$S[T \leftarrow C_j :: o | C_j \in SEC(T, C), SEC(T, C) \neq \emptyset]$$

Рішення:

1. Застосовуємо операцію 4.1 для фрагмента  $T$  та замінюємо фрагмент  $T$  сполуки  $S$  на об'єкт СЕК з урахуванням з'єднання входів та виходів. Наприклад, фрагмент сполуки  $S$  (рис.1), що складається з об'єктів  $o_1$  та  $o_3$  можна замінити на об'єкт класу  $G$  (рис.3).

2. Змінюємо представлення сполуки  $S$ .

#### 4.4. Частковий пошук та підстановка ФЕК (часткова ФЕ заміна).

Постановка задачі.

Дано:

1. Повна формула сполуки  $S$  та її виділений фрагмент у вигляді сполуки  $T$ .

2. Множина інтегральних класів  $C = \{C_i | i = \overline{1, I}\}$ , кожний з яких має дескриптор інтегрального класу  $D_i$ .

Потрібно: знайти ФЕК фрагменту  $T$  сполуки  $S$  та замінити його на об'єкт класу  $C_j$ :

$$S[T \leftarrow C_j :: o | C_j \in FEC(T, C), FEC(T, C) \neq \emptyset]$$

Рішення:

1. Застосовуємо операцію 4.2 для фрагмента  $T$  та замінюємо фрагмент  $T$  сполуки  $S$  на об'єкт ФЕК  $C_j$  з урахуванням з'єднання входів та виходів.

2. Змінюємо представлення сполуки  $S$ .

**Висновки.** Розроблено множину базових операцій формального структурного аналізу програм, створених за архітектурою інтегральних об'єктів, на основі табличного представлення, що дозволяє виконувати як еквівалентні перетворення сполук об'єктів, так і застосовувати формальні методи проектування системи класів під час розробки програм розв'язку задач предметної області.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] А. Э. Молчанов, “Разрешимость проблемы эквивалентных преобразований в классе примитивных схем программ”, *Труды Института системного программирования РАН*, т. 27, № 2, с. 173-188, 2015.
- [2] Р. И. Подловченко, “Исследование примитивных схем программ с процедурами”, *Моделирование и анализ информационных систем*, т. 21, № 4, с.116-131, 2014.
- [3] Р. И. Подловченко, “К вопросу о полиномиальной сложности проблемы эквивалентности в алгебраических моделях программ”, *Кибернетика и системный анализ*, т. 48, № 5, с.17-24, 2012.
- [4] В. Дубовой, О. Никитенко, та О. Глонь, “Еквівалентність невизначених алгоритмів”, *Наукові праці Вінницького національного технічного університету*, № 2, с.1-8, 2009.
- [5] В. В. Соколов. “Застосування функціональної та реляційної моделей в об'єктно-орієнтованому програмуванні”, *Information Technology and Security*, vol. 5, iss. 1, pp. 42-54, 2017.
- [6] В. В. Соколов. “Архітектура програмного забезпечення на основі інтегральних об'єктів”, *Information Technology and Security*, vol. 5, iss. 2, pp. 51-59, 2017.

Стаття надійшла до редакції 6 вересня 2018 року.

## REFERENCE

- [1] A. Molchanov, "A Solution to the Equivalent Transformation Problem in a Class of Primitive Program Schemes", *Proc. Institute of System Programming of the Russian Academy of Sciences*, vol. 27, no. 2, pp. 173-188, 2015.
- [2] R. I. Podlovchenko, "Primitive Program Schemes with Procedures", *Modeling and analysis of information systems*, vol. 21, no. 4, pp.116-131, 2014.
- [3] R. I. Podlovchenko, "On the polynomial complexity of equivalence checking problem in algebraic models of programs", *Cybernetics and Systems Analysis*, vol. 48, no. 5, pp.17-24, 2012.
- [4] V. Dubova, O. Nikitenko, and O. Glon, "Equivalence of uncertain algorithms", *Proc. Vinnytsia National Technical University*, no. 2, pp.1-8, 2009.
- [5] V. Sokolov, "Application of functional and relational models in object-oriented programming", *Information Technology and Security*, vol.5, iss. 1, pp. 42-54, 2017.
- [6] V. Sokolov, "Architecture of software based on integrated objects ", *Information Technology and Security*, vol. 5, iss. 2, pp. 51-59, 2017.

ВЛАДИМИР СОКОЛОВ

### СТРУКТУРНЫЙ АНАЛИЗ СОЕДИНЕНИЙ ИНТЕГРАЛЬНЫХ ОБЪЕКТОВ

В статье представлены результаты исследований методов анализа структуры программного обеспечения, построенного по архитектуре на основе интегральных объектов. Для определения операций структурного анализа выбрано представление соединений интегральных объектов в виде формул соединений и таблиц, а также формализованы операции интеграции и дезинтеграции соединений. Интеграция соединения с образованием нового класса заключается в инкапсуляции соединения в оболочку класса с присвоением степени интеграции классу на единицу больше, чем у соединения, а дезинтеграция заключается в обратном процессе извлечения соединения из оболочки класса с уменьшением степени интеграции на единицу. Табличное представление соединений объектов, которое включает как представление объектов, так и связей между ними, позволяет применить формальные методы анализа их структуры. Определено понятие структурной эквивалентности двух соединений как имеющих одинаковый состав и структуру соединения объектов, а также понятие функциональной эквивалентности двух соединений как реализующих одинаковую композицию функций. В качестве основных операций структурного анализа представлены две базовые операции- определение структурной и функциональной эквивалентности двух соединений. Определение эквивалентности соединений сводится к сравнению табличного представления соединений с учетом перестановки строк и столбцов таблиц. Показано, что определение структурной эквивалентности возможно только для соединений одинаковой степени интеграции, а для определения функциональной эквивалентности нужно приведения соединений до атомарного уровня. Определение функциональной эквивалентности сведено к определению структурной эквивалентности атомарных формул соединений. В качестве производных операций выделено операции распознавания структурно и функционально эквивалентных классов для заданного соединения, основанные на базовых операциях определения эквивалентности, а также четыре операции эквивалентных преобразований соединений: структурно и функционально эквивалентная замена соединения на объект эквивалентного класса и частичная структурно и функционально эквивалентна замена фрагмента соединения на объект эквивалентного класса. Операции структурного анализа являются основой для эквивалентных преобразований соединений в процессе проектирования, верификации и оптимизации системы классов решения задач предметной области при разработке программного обеспечения.

**Ключевые слова:** структурный анализ, структурная эквивалентность, функциональная эквивалентность, интегральный объект, соединения интегральных объектов.

VOLODYMYR SOKOLOV

## STRUCTURAL ANALYSIS OF THE COMPOUNDS OF INTEGRAL OBJECTS

The article presents the results of research on methods of analysis of the structure of software, built on architecture based on integral objects. To determine the structural analysis operations, the representation of the compounds of integral objects in the form of formulas of compounds and tables were chosen, and operations of integration and disintegration of compounds were formalized. The integration of compound with formation of a new class consists in encapsulating the compound into the class shell with the integration degree of the class one greater than that of the compound, and the disintegration consists in the reverse process of extracting the compound from the class shell with a decrease of integration degree by one. The table representation of the compounds of objects, which includes both the representation of objects and the connections between them, allows us to apply formal methods for analyzing their structure. The notion of structural equivalence of two compounds as having the same composition and structure of the objects compounds is determined, as well as the notion of functional equivalence of two compounds as those implementing the same composition of functions. As the main structural analysis operations, two basic operations are presented – the determination of the structural and functional equivalence of the two compounds. The determination of the equivalence of the compounds is reduced to comparing the table representation of the compounds, taking into account the permutation of the rows and columns of the tables. It is shown that the determination of structural equivalence is possible only for compounds of the same integration degree, and for the determination of functional equivalence, it is necessary to bring the compounds to the atomic level. The definition of functional equivalence is reduced to the determination of the structural equivalence of atomic formulas of compounds. The operations of recognition of structurally and functionally equivalent classes for a given compound are chosen as derivative operations, which are based on basic equivalence determination operations, as well as four operations of equivalent transformations of compounds: structurally and functionally equivalent replacement of a compound by an object of the equivalent class and partially structurally and functionally equivalent replacement of a fragment of a compound by an object of the equivalent class. Structural analysis operations are the basis for the equivalent transformation of compounds in the process of designing, verifying and optimizing the system of classes of the problem domain solution during software development.

**Keywords:** structural analysis, structural equivalence, functional equivalence, integral object, integral objects compounds.

**Володимир Володимирович Соколов**, кандидат технічних наук, доцент, доцент кафедри кібербезпеки та застосування автоматизованих інформаційних систем та технологій, Інститут спеціального зв'язку та захисту інформації національного технічного університету “Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна.

ORCID: 0000-0002-5779-7167.

E-mail: vsokolov@i.ua.

**Владимир Владимирович Соколов**, кандидат технических наук, доцент, доцент кафедры кибербезопасности и применения информационных систем и технологий, Институт специальной связи и защиты информации национального технического университета “Киевский политехнический институт имени Игоря Сикорского”, Киев, Украина.

**Volodymyr Sokolov**, candidate of technical sciences, associate professor, associate professor at the cybersecurity and application of information systems academic department, Institute of special communication and information protection of National technical university of Ukraine “Igor Sikorsky Kyiv polytechnic institute”, Kyiv, Ukraine.