

ПСЕВДО SH-МОДЕЛЬ

© Черкаський М.В., Абдалла Саїд Садек, 2004

Дано визначення псевдо SH-моделі програми. Показано, що псевдо SH-модель відрізняється від SH-моделі і універсальних математичних моделей алгоритмів відсутністю керуючої складової. Розглянуто принципи побудови псевдо SH-моделі з точки зору теорії складності. Подано перелік характеристик складності. Наведено приклад розрахунку структурної складності.

The definition for SH-model algorithm is presented. The pseudo SH- model differ from SH-model and universal mathematical model algorithm that he don't have a control signal. Here is also shown the principal theory of pseudo SH-model. Expand the list of complexity characteristics. Given an example that calculates the structural complexity.

Вступ. Поширеною оцінкою програмного продукту є кількість операторів, записаних мовою високого рівня [1]. Але вона не може претендувати на вичерпність. Традиційно важливою характеристикою програмно-апаратних комп'ютерних засобів розв'язання задач є часова складність, яка вимірюється кількістю операцій над даними, необхідних для розв'язання. Кількість інструкцій програми і кількість операцій, необхідних для її реалізації, не одне і те саме. Лише деякою мірою кількість інструкцій непрямо може використовуватись для оцінки часових співвідношень різних складових програми. Те саме стосується оцінки інформаційності програмних розробок. Хоча кількість операторів в середньому без врахування особливості структурних побудов дає уявлення про інформаційний вміст програмного продукту, але не дає змогу пов'язати його із структурними особливостями програм. Для дослідження цієї характеристики більше придатна не сама програма, а її блок-схема. Тут пропонується спосіб оцінки структурної складності програми по блок-схемі за допомогою характеристик SH-моделі.

SH-модель алгоритму. Для аналізу програмно-апаратних засобів може використовуватись SH-модель алгоритму (SH – Software/Hardware) [2].

SH-модель – це сімка

$$B = \langle D, Q, q_0, q_f, G, P, M \rangle, \quad (1)$$

де D – кінцева множина символів зовнішнього алфавіту; Q – кінцева множина станів SH-моделі; q_0 і q_f – початковий і кінцевий стани, $q_0, q_f \in Q$; G – конфігурація апаратних засобів моделі;

$$G = (X, U), \quad (2)$$

де X – множина елементарних перетворювачів; U – множина між'єднань; P – така програма

$$P = \{D\} \times \{Q\} \times \{X\} \times \{\rightarrow\} \times \{D\} \times \{Q\}, \quad (3)$$

що не може мати трьох однакових перших символів; M – зовнішня віртуальна пам'ять.

SH-моделі не мають раз і назавжди встановленої структури. Однак кожна конкретна модель алгоритму стосовно апаратної побудови має точно окреслену структуру, яка складається з двох множин: множини елементарних перетворювачів і множини між'єднань:

$$\begin{aligned} X &= \{x_1, x_2 \dots x_n\}; \\ U &= \{u_1, u_2 \dots u_m\}. \end{aligned} \quad (4)$$

Кожний елементарний перетворювач виконує деяку операцію. Ці операції можуть бути різними для різних перетворювачів.

Властивості SH-моделі. SH-модель має п'ять властивостей, які впливають з вербального інтуїтивного тлумачення алгоритму.

- Дискретність. Робота SH-моделі здійснюється множиною обмежених у часі кроків. Кожний крок може містити елементарні операції перетворення, передачу даних від одного елементарного перетворювача до іншого, а також операцію запису даних в елементи пам'яті.
- Елементарність. Операції перетворення, передачі і запису даних в елементи пам'яті SH-моделі є простими і локальними в просторі і часі. Слову “елементарність” надається ієрархічний зміст.
- Детермінованість. Кожний крок алгоритму повністю визначений функцією елементарного перетворювача і командою програми. Напрямок передачі даних від одного елементарного перетворювача до іншого точно визначений напрямком, що задають з'єднання або команди програми.
- Масовість. Ця властивість має дворівневий зміст:
 - 1) універсальність – одна і та сама SH-модель може бути застосована для розв'язання будь-якої задачі, що формалізується;
 - 2) масовість у традиційному тлумаченні – одна і та сама SH-модель може бути застосована для будь-якої кількості задач, які відрізняються набором вхідних даних при постійному правилі безпосереднього перероблення.
- Ієрархічність. Кожний елементарний перетворювач може бути представлений SH-моделлю нижчого ієрархічного рівня. З іншого боку, кожна SH-модель може бути використана як елементарний перетворювач вищого ієрархічного рівня.

Характеристики складності SH-моделі. У процесах синтезу, аналізу і оптимізації SH-моделей пропонується використовувати п'ять характеристик складності: апаратну, часову, програмну, структурну та ємнісну.

Апаратна складність – кількість елементарних перетворювачів і елементів тимчасової пам'яті деякого ієрархічного рівня апаратних засобів SH-моделі:

$$A = |X|, \quad (5)$$

де X – множина елементів схеми.

Визначення відображає ієрархічну побудову комп'ютерних засобів. Якщо під SH-моделлю розуміти операційний пристрій, то як елементарні перетворювачі розглядаються однорозрядні комірки або вентиля для рівня регістрових передач елементами є операційні пристрої, на системному рівні SH-моделлю є комп'ютер та інше.

Наведене визначення апаратної складності не суперечить поняттю “об'єм обладнання”, що використовується в обчислювальній техніці.

Часова складність. У метричній теорії під часовою складністю розуміють кількість елементарних операцій, наприклад, кроків машини Тюрінга. У теорії software/hardware часова складність визначається дещо інакше.

Часова складність SH-моделі визначається кількістю елементів схеми, розташованих вздовж максимального критичного шляху розповсюдження сигналу:

$$L = |\max X_i|, \quad (6)$$

де $\max X_i$ – кортеж елементів SH-моделі, що належать до максимального критичного шляху розповсюдження сигналу, включаючи повторні проходження елементів в циклі.

Одиницею часової складності є елементарний перетворювач деякого ієрархічного рівня схеми. Перехід від часової складності до визначення часу спрацювання схеми наведений у такій формулі:

$$T = \sum_{\tau_{e_i} \in \max |e_i|} \tau(e_i), \quad (7)$$

де $\tau(e_i)$ – реальна затримка сигналу на елементі e_i .

Ємнісна складність SH-моделі дорівнює кількості комірок зовнішньої пам'яті, яка потрібна для розв'язання цієї задачі.

Апаратна, часова і ємнісна складності комп'ютерних засобів належать до технічних характеристик. Крім них, в процесі аналізу SH-моделі використовують додатково програмну і структурну складності, які належать до інформативних характеристик.

Програмна складність. Часова діаграма – це двовимірна таблиця в координатах: дискрети часу; входи керування. На процесорному рівні кожній асемблерній команді відповідає власна часова діаграма мікропрограми. Програмна складність оцінюється ступенем нерегулярності (ентропії) часової діаграми:

$$P = -F \log_2 \frac{F}{n \cdot m}, \quad (8)$$

де $F = \sum_L f_l$; n – кількість входів керування; m – кількість дискрет часу часової діаграми; f_l – кількість сигналів керування l -го фрагмента часової діаграми для обраного рівня ієрархії побудови апаратних засобів; L – кількість фрагментів часової діаграми, конфігурації яких не повторюються.

Структурна складність відображає ступінь нерегулярності міжзв'язків схеми деякого рівня ієрархії побудови апаратних засобів. Структурна складність алгоритмічного пристрою – це ентропія матриці інциденцій:

$$S = -E \log_2 \frac{E}{q \cdot r}, \quad (9)$$

де E – кількість елементів матриці інциденцій системи; $q \cdot r$ – розмір матриці.

Структурна складність SH-моделі визначається аналогічним способом, що і програмна. Відмінність лише у тому, що об'єктом розрахунків є матриця інциденцій. Структурну складність визначається в три етапи:

1. Схема SH-моделі перетворюється в орграф.
2. Орграф кодується у вигляді матриці інциденцій.
3. Розраховується значення нерівномірності матриці інциденцій.

Блок-схема програми. Блок-схема програми, як і SH-модель, алгоритму не має, подібно до машини Тюрінга, жорстко заданої структури. Блок-схема програми також є множиною “елементарних” інструкцій, об'єднаних орієнтованими з'єднаннями. Блок-схема програми є статичним об'єктом. Вона, як і програма мовою високого рівня, є лише приписом, набором інструкцій для розв'язання задачі іншими засобами. Ці засоби не входять до її складу. Блок-схема програми відрізняється від інших моделей алгоритму, від функціональних комп'ютерних схем тим, що вона не є носієм процесу. Для неї, на відміну від SH-моделі, не можна побудувати пристрій керування.

Разом з тим, блок-схема програми має спільні риси з SH-моделлю. Вона також складається з множини вершин інструкцій, які можна ототожнити за змістом з елементарними перетворювачами SH-моделі і надати їм умовну назву “елементарна” інструкція. Кожна вершина уявляє собою чорну скриньку з входом та виходом, що може перетворювати вхідні дані у вихідні. Але, на відміну від SH-моделі, ця чорна скринька не має входу керування, не має множини сигналів керування – програми.

Ці міркування дозволяють дати формальний опис блок-схеми програми, подібний до опису SH-моделі, але без системи керування. Будемо називати формальний опис блок-схеми програми псевдо SH-моделлю.

Визначення псевдо SH-моделі. Псевдо SH-модель – це п'ятірка

$$\langle D, Q, q_0, q_f, G, M \rangle, \quad (10)$$

де D – кінцева множина символів зовнішнього алфавіту; Q – кінцева множина станів моделі; q_0 і q_f – початковий і кінцевий стани, $q_0, q_f \in Q$; G – конфігурація блок-схеми програми:

$$G = (X, U), \quad (11)$$

де X – множина вершин блок-схеми; U – множина орієнтованих з'єднань;

$$\begin{aligned} X &= \{x_1, x_2, \dots, x_n\}; \\ U &= \{u_1, u_2, \dots, u_m\}. \end{aligned} \quad (12)$$

M – об'єм зовнішньої пам'яті, необхідний для розв'язання задачі.

Властивості псевдо SH-моделі. Псевдо SH-модель, як і SH-модель, має п'ять властивостей: дискретність, детермінованість, елементарність, масовість, ієрархічність. Властивість елементарність тут визначається також, як для SH- моделі алгоритму, ототожнюванням кожної вершини блок-схеми програми з чорною скринькою. У цьому випадку інструкції, які відповідають вершинам блок-схеми, можуть бути інтегрованими. Звідси випливає наявність для псевдо SH-моделі властивості “ієрархічність”. Решта властивостей тотожні властивостям SH-моделі.

Характеристики складності псевдо SH-моделі. Характеристики складності псевдо SH-моделі суттєво відрізняються за змістом і кількістю. Для псевдо SH-моделі їх є три: часова, об'єктна, структурна.

Часова складність. Одиницею часової складності псевдо SH-моделі є одна інструкція програми. Часова складність визначається потужністю інструкцій $\max|X|$, необхідних для розв'язання задачі. Отримання витрат реального часу, потрібних для розв'язання задачі розрахунками з використанням псевдо SH-моделі, неможливе. Дуже великий вплив на ці витрати мають характеристики комп'ютерних засобів. Тому для оцінки витрат часу використовують вимірювання продуктивності програм – профілювання [3].

Об'єктна складність. Псевдо SH-модель не оцінюється апаратною складністю. Замість неї будемо використовувати назву “об'єктна складність”. Її одиницею є вершина блок-схеми. Кількість вершин деякого ієрархічного рівня псевдо SH-моделі є об'єктною складністю.

Структурна складність. Ця оцінка псевдо SH-моделі збігається з аналогічною оцінкою SH-моделі. Структурна складність псевдо SH-моделі є ентропією матриці інциденцій, побудованої на основі блок-схеми програми:

$$S = -X \log_2 \frac{X}{q \cdot r}, \quad (13)$$

де X – кількість елементів матриці інциденцій системи; $q \cdot r$ – розмір матриці.

У матриці можуть існувати фрагменти, що повторюються. По визначенню вони не повинні враховуватися у значенні структурної складності. У такому випадку $X = \sum_L \varphi_l$, де φ_l – кількість з'єднань l -го неповторюваного фрагмента матриці інциденцій схеми.

Вибір фрагментів проводиться з логічних умов:

$$\forall i, j; i \neq j \{ [\psi_i \cap \psi_j \neq \psi_i \vee \psi_j] \Rightarrow \varphi = |\psi_i| + |\psi_j| \} \vee \{ [\psi_i \cap \psi_j = \psi_i \vee \psi_j] \Rightarrow \varphi = |\psi_i| \}, i, j = \overline{1, N}, \quad (14)$$

де ψ_i, ψ_j – i -й та j -й фрагменти матриці інциденції.

Структурна складність псевдо SH-моделі визначається так само, як для SH-моделі. Відмінність є лише в тому, що об'єктом дослідження є матриця інциденцій блок-схеми послідовних програм. Одержання структурної складності проводиться в чотири етапи:

1. Блок-схема програми перетворюється в оргграф.
2. Оргграф стискується за рахунок скорочення кількості послідовно з'єднаних операційних вершин.
3. Отриманий оргграф кодується у вигляді матриці інциденцій.
4. Розраховується значення нерівномірності матриці інциденцій.

Ємнісна складність псевдо SH-моделі дорівнює кількості комірок зовнішньої пам'яті, яка потрібна для розв'язання цієї задачі. Вартісна вага ємнісної складності порівняно з часовою, об'єктною і структурною невелика. Тому під час оптимізації псевдо SH-моделі за характеристиками складності її у більшості випадків можна не враховувати.

Приклад розрахунку характеристик складності псевдо SH-моделі. На рис. 1, а наведено блок-схему абстрактної програми. Для неї неможливо підрахувати потрібну для її реалізації кількість операцій – часову складність. Немає також даних для підрахунку ємнісної складності. Ці підрахунки здійснюються для кожної задачі окремо. Визначення об'єктної складності тривіальне, вона дорівнює кількості вершин блок-схеми програми. У нашому випадку $|X| = 14$.

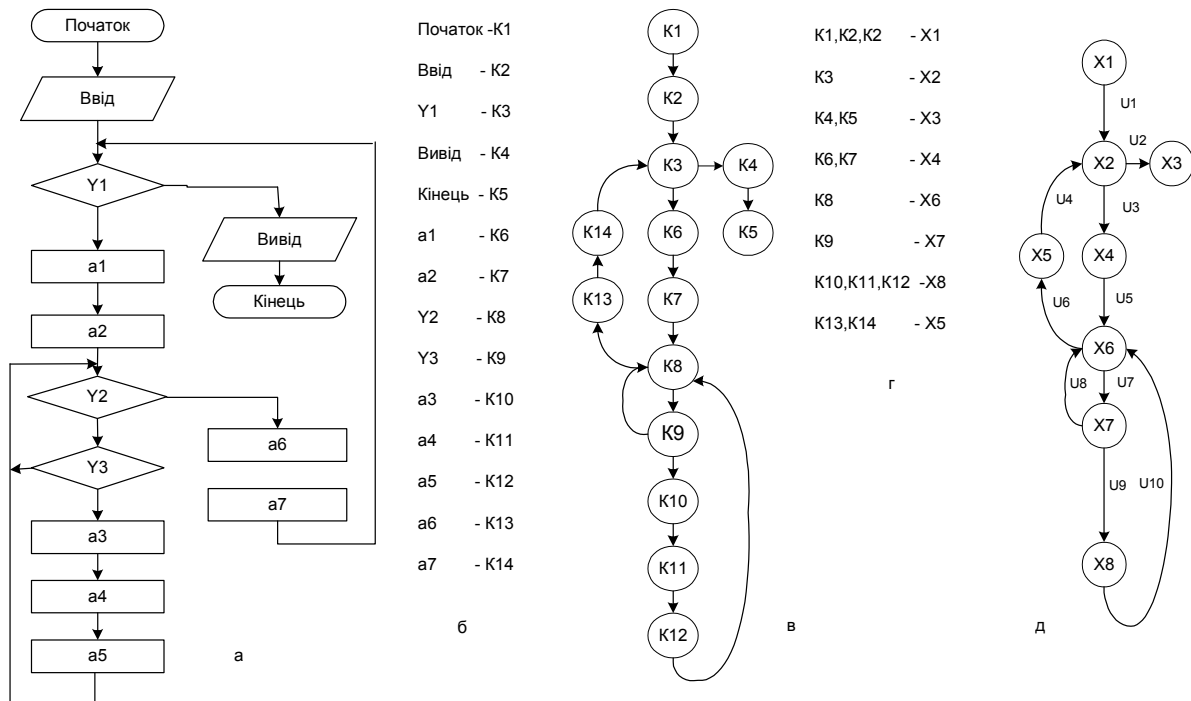


Рис. 1. Блок-схема абстрактної програми

Основну увагу приділимо визначенню структурної складності, основному показнику інтелектуальних витрат під час розроблення програм. Послідовність операцій розрахунку така. Спочатку блок-схему перетворюємо в оргграф (рис.1, в), потім стискаємо його (рис. 1, д), далі будуємо матрицю інциденцій (рис. 2) і на її основі розраховуємо значення структурної складності.

$$I = \begin{matrix} & U1 & U2 & U3 & U4 & U5 & U6 & U7 & U8 & U9 & U10 \\ \begin{matrix} X1 \\ X2 \\ X3 \\ X4 \\ X5 \\ X6 \\ X7 \\ X8 \end{matrix} & \left| \begin{array}{cccccccccc} 1 & & & & & & & & & & \\ -1 & 1 & 1 & -1 & & & & & & & \\ & -1 & & & & & & & & & \\ & & -1 & 1 & & & & & & & \\ & & & 1 & -1 & & & & & & \\ & & & & -1 & 1 & 1 & -1 & & -1 & \\ & & & & & -1 & 1 & 1 & & & \\ & & & & & & & -1 & 1 & 1 & \\ & & & & & & & & -1 & 1 & \end{array} \right| \end{matrix}$$

Рис. 2. Матриця інциденцій

$$S = -E \log_2 \frac{E}{n * m} = -20 \log_2 \frac{20}{8 * 10} = 40.$$

Отже, структурна складність становить сорок біт.

Висновки. 1. Комп'ютерна програма є складовою алгоритму, разом із засобами її реалізації утворює SH-модель.

2. Псевдо SH-модель є моделлю комп'ютерної програми. Вона дозволяє розширити існуючий перелік характеристик складності для оцінки програмного продукту за рахунок структурної складності – інформаційного вмісту блок-схеми програми.

3. Псевдо SH-модель програми має властивість ієрархічності, що дозволяє використовувати її для аналізу структурної складності програмного продукту на різних рівнях детальності.

1. Кормен Т., Лейзерсон Ч., Ривест Р. *Алгоритмы: построение и анализ / Пер. с англ. – М.: МЦНМО, 2000*. 2. Черкаський М.В. *SH-модель алгоритму // Вісн. Нац. ун-ту “Львівська політехніка”*. – 2001. – № 433. – С. 127–134. 3. Касперски К. *Техника оптимизации программ. Эффективное использование памяти. – СПб.: БХВ-Петербург, 2003.*

УДК 621.3

М.В. Черкаський, Хусейн Халіл Мурад
Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

УНІВЕРСАЛЬНА SH-МОДЕЛЬ

© Черкаський М.В., Мурад Хусейн Халіл, 2004

Показано, що універсальна SH-модель відрізняється від універсальних математичних моделей алгоритмів за призначенням і способом синтезу. Розглянуті принципи побудови універсальної SH-моделі з погляду теорії складності апаратно-програмних алгоритмів. Сформульовані вимоги оптимізації до універсальної SH-моделі, що складається з двох SH-моделей – функціональної та керуючої.

Shown, that the universal SH-model differs from universal mathematical models of algorithms by the destination and the way of synthesis. The considered principles of construction the universal SH-model from the point of view of the theory of complexity of hardware-software algorithms. The formulated requirements of optimization to universal SH-model which will consist of two SH-models – functional and controlling.

Вступ. Поняття “універсальність” алгоритмічних систем використовується у математичній теорії абстрактних алгоритмів. Першою в історичному плані універсальною алгоритмічною системою був апарат рекурсивних функцій [1]. Універсальність допомагає довести наявність вирішення будь-яких математичних проблем за допомогою обмеженого набору спеціальних функцій. Досі рекурсивні функції залишаються основним інструментом дослідження проблем розв’язувальності.

Наступним кроком розвитку теорії алгоритмів стали дослідження машиноорієнтованих абстрактних моделей алгоритмів. Першою з них була машина Тюрінга [2]. Вона використовує одну і ту саму структуру (нескінченну стрічку, головку, програму) для реалізації будь-якого алгоритму, але для кожного алгоритму має власну програму і для кожної задачі має власну конфігурацію початкових даних, розташованих в комірках стрічки. Машина Тюрінга дозволила уточнити поняття алгоритму, описати перелік параметрів і характеристик, започаткувати теорію складності алгоритмів. Ця абстрактна модель дає майже повне уявлення про всі властивості алгоритму, в тому числі про властивість масовості, яка означає, що один і той самий алгоритм може розв’язувати множину задач деякого класу. Інакше кажучи, машина Тюрінга з однією і тією самою програмою може розв’язувати задачі з різними конфігураціями початкових даних, якщо зовнішні алфавіти програми і даних збігаються. Одна програма – один клас обчислювальних задач.

Універсальна машина Тюрінга повинна мати одну розширену програму, яка б дозволяла розв’язувати будь-які обчислювальні задачі. Під час побудови цієї машини виникають дві проблеми. По-перше, вибір способу кодування конфігурацій даних і програм; по-друге, вибір способу розміщення