

зміни кількості зовнішніх зв'язків новоутвореної частини $B+\alpha$ порівняно з частиною B достатньо визначити кількість зв'язків типів e_1, e_2, e_3 та e_5 . Відповідний алгоритм визначення множин цих зв'язків має вигляд:

1. $E_1 = E_\alpha^{ex} \cap E_A^{ex}$.
2. $E_2 = \{e \in (E_\alpha^{ex} \cup E_1 \cup E_B^{ex}) \mid P(e) \cap P(A-\alpha) \neq \emptyset\}$.
3. $E_3 = E_\alpha^{ex} \setminus (E_1 \cup E_2 \cup E_B^{ex})$.
4. $E_5 = \{e \in (E_\alpha^{ex} \setminus (E_1 \cup E_2 \cup E_3)) \mid (P(e) \subset P(\alpha) \cup P(B))\}$.
5. $E_6 = \{e \in (E_\alpha^{ex} \setminus (E_1 \cup E_2 \cup E_3 \cup E_5)) \mid P(e) \cap P(A-\alpha) = \emptyset\}$.

Висновки

Як бачимо з описаного алгоритму, на кількість зв'язків новоутвореної схеми N^* з частин $A-\alpha$ та $B+\alpha$ впливають тільки два типи зв'язків: e_1 та e_5 . Ідентифікація зв'язків тільки цього типу суттєво спрощує та прискорює алгоритм. Якщо ж виникає додаткова задача визначення кількості зовнішніх зв'язків новоутворених частин $A-\alpha$ чи $B+\alpha$, то необхідно також ідентифікувати зв'язки типів E_3 та E_6 (для частини $A-\alpha$) або E_2 та E_5 (для частини $B+\alpha$), або E_2, E_1 та E_6 , коли необхідно знати обидві ці величини та визначити їх кількість.

1. Базилевич Р.П. Декомпозиционные и топологические методы автоматизированного конструирования электронных устройств. Львов, 1981.
2. Роман Базилевич. Ієрархічні кластеризація, декомпозиція та багаторівневе макромодельовання - ефективні засоби розв'язування комбінаторних задач схемного типу великої та надвеликої розмірності. Зб. наук. праць "Сучасні проблеми в комп'ютерних науках", ДУ "Львівська політехніка", 2000, -С.15 - 30.
3. Базилевич Р.П., Рибак О.Г. Оптимізації пакування програмованих логічних матриць на основі оптимального згоргання схеми. Вісник ДУ "Львівська політехніка", № 385, 1999.

УДК 621.3

SH - МОДЕЛЬ АЛГОРИТМУ

© М. Черкаський

Національний університет "Львівська політехніка"

Дано визначення SH-моделі алгоритму. Ця модель враховує у своєму визначенні апаратні засоби, що дозволяє розширити перелік характеристик складності за рахунок апаратної і структурної. Модель дозволяє досліджувати складність побудови комп'ютерних засобів на різних ієрархічних рівнях.

The definition for SH-model algorithm is presented. The model considers hardware in its definition, which allows to expand the list of complexity characteristics at the expense of instrumental and structural complexities. The model gives a possibility to reseach complexity of software/hardware on various hierachal levels.

Вступ

У першій половині 20-го сторіччя розвиток теорії алгоритмів був зумовлений проблемою розв'язності задач. Для дослідження цієї проблеми було зроблено декілька уточнень стосовно поняття "алгоритм". Було сформульовано вербальне тлумачення алгоритму, введені параметри, описані властивості.

Одне з останніх тлумачень алгоритму є таке: "Алгоритм – це точне приписання, що задає обчислювальний процес, який починається з деякої системи вихідних даних і спрямований на одержання результату, повністю визначеному цими вихідними даними" [1].

Це тлумачення алгоритму не дає відповіді на те, якими засобами реалізувати "точне приписання, що задає обчислювальний процес". Щоб здійснити цей процес, необхідні деякі матеріальні засоби перероблення в дискретному часі вхідних даних у вихідні згідно з заданим наперед правилом безпосереднього перероблення. Для цього необхідна модель програмних і апаратних засобів, яка реалізує це правило (тут під апаратними розуміються будь-які засоби, що здійснюють обчислювальний процес, наприклад, людина-обчислювач, автомати, обчислювальні машини, які працюють під управлінням програми та інш.).

Обов'язковими умовами побудови моделі алгоритму є: її відповідність інтуїтивному тлумаченню поняття "алгоритм", наявність у моделі всіх параметрів алгоритму і виконання моделлю вимог, що впливають з властивостей алгоритму. Властивості: дискретність, детермінованість, спрямованість, масовість можуть бути визначені точно математично. Властивість "елементарність" в існуючих моделях алгоритму математично точного визначення не має. Прийнято [2] елементарність тлумачити як простоту і локальність, іноді [3] як загальну зрозумілість. Щоб запобігти помилкам при дослідженні складних задач, кроки алгоритмів не повинні мати інтегральний характер. Таке нестроге припущення щодо властивості "елементарність" сприяло виникненню декількох формальних алгоритмічних систем (ФАС). Найбільш розповсюдженою машино-орієнтованою моделлю алгоритму, виходячи з частоти літературних посилань, є машина Тьюрінга.

Особливості моделі алгоритму машини Тьюрінга

Модель однорядкової детермінованої машини Тьюрінга задається шісткою:

$$M = \langle A, Q, q_0, q_f, a_0, P \rangle, \quad (2)$$

де A – кінцева множина символів зовнішнього алфавіту; Q – кінцева множина символів внутрішнього алфавіту; q_0 і q_f – початковий і кінцевий стани, $q_0, q_f \in Q$; a_0 – позначення порожньої комірки рядка, $a_0 \in Q$; P – така програма, яка не може мати двох команд, у яких би збігалися два перші символи:

$$P = \{A\} \times \{Q\} \rightarrow \{A\} \times \{L, R, S\} \times \{Q\}, \quad (3)$$

де L – зсувати головку вліво; R – зсувати головку вправо; S – залишити головку на місці.

Машини Тьюрінга мають одну і ту ж конфігурацію засобів реалізації алгоритму для розв'язання будь-якої задачі. В конфігурацію входять: нескінченний нерухомий рядок, що поділений на окремі комірки, в які можна розмістити не більше одного символу зовнішнього алфавіту; рухома головка, яка може стирати, записувати і зчитувати символи зовнішнього алфавіту в комірках рядка; програма із кінцевою кількістю команд.

Ці елементи і лінії передавання повідомлень, що їх пов'язують, утворюють структуру машини Тьюрінга, яка не залежить від структури конкретного алгоритму [4]. Ця важлива особливість машини Тьюрінга дозволяє кількісно порівнювати різні алгоритми з часової, місткісної складності і складності програм.

Машина Тьюрінга як модель алгоритму відповідає інтуїтивному тлумаченню алгоритму. В явному вигляді тут означені всі сім параметрів. Особливості роботи машини Тьюрінга не суперечать властивостям алгоритму. Кроки машини Тьюрінга дискретні і детерміновані, мають властивість масовості. Єдина властивість, яка приймається умовно, це елементарність кроку. В машині Тьюрінга крок алгоритму супроводжується декількома операціями: читання символу в комірці рядка, пошук необхідної команди, виконання команди – операція із змістом комірки (залишити попередній символ, стерти його, записати новий), операція переміщення головки (залишити на місці, зсунути вліво чи вправо). Всі ці операції, що становлять крок алгоритму, є неінтегрованими, загальнозрозумілими. Використання машини Тьюрінга дозволяє розширити список характеристик складності алгоритмів і тим самим започаткувати метричну теорію складності. Часова складність задається послідовністю миттєвих станів машини Тьюрінга. Місткісна складність вимірюється кількістю комірок рядка, яка необхідна для реалізації алгоритму. Складність програми визначається кількістю команд.

Найбільше практичне значення для економії обчислень має часова складність. В метричній теорії способам побудови алгоритмів з мінімізованою часовою складністю приділяється головна увага. Мінімізація часової складності машині Тьюрінга пов'язана з використанням таких способів:

- зміна розташування початкових даних на рядку;
- вибір місця розташування проміжних результатів;
- вибір стратегії руху головки;
- вибір початкового положення головки;
- збільшення символів зовнішнього алфавіту;
- застосування паралелізму (багаторядкова машина Тьюрінга).

Наведені способи мінімізації часової складності, крім останнього, мають теоретичне застосування, але їх практичне значення для комп'ютерної реалізації невелике.

Машина Тьюрінга є абстрактною ідеалізованою моделлю алгоритму. На відміну від моделей *software/hardware*, вона не враховує апаратні витрати, необхідні для реалізації алгоритму. Ця особливість абстрактних моделей не дозволяє у повній мірі використовувати досягнення теорії ФАС у проектуванні апаратно-програмних засобів. А у деяких випадках цей недолік приводить до практично неприйнятних висновків. Прикладом тому є теорема про лінійне прискорення. Для моделей *software/hardware* ця

теорема не коректна.

Критика абстрактних моделей

Відмінність сучасних комп'ютерних засобів від розглянутих моделей алгоритмів полягає у враховуванні апаратної реалізації як однієї із складових визначення апаратно-програмної моделі алгоритму. Правило безпосереднього перероблення в цих моделях виконується апаратними засобами повністю або разом з програмним керуванням.

Частково апаратні засоби можуть бути замінені програмними засобами виконання алгоритму, і в цьому сенсі апаратне виконання еквівалентне програмному. Використання апаратно-програмної моделі алгоритму дозволяє в процесі синтезу алгоритмів використовувати додаткову характеристику - апаратну складність.

Розглянуті абстрактні моделі алгоритму містять ті чи інші апаратні засоби або припускають їх наявність. Наприклад, в машині Тьюрінга припускається наявність рухомих головок, засобів читання і запису, дешифраторів при виборі тих чи інших символів зовнішнього та внутрішнього алфавітів. Але під час побудови ефективних алгоритмів вони не враховуються. Нормальні алгоритми Маркова також містять підмножину елементарних операторів, розпізнавачів і перетворювачів, але в систему характеристик апаратні затрати не входять.

Отже, для переходу від абстрактних ФАС до комп'ютерних моделей необхідне врахування апаратних засобів.

Елементарні перетворювачі. Апаратні засоби містять один або декілька елементарних перетворень. Елементарний i -й перетворювач x_i є одиницею апаратної складності.

$$\forall x_i = 1. \quad (4)$$

Елементарний перетворювач x_i перетворює деяку сукупність початкових даних в сукупність вихідних даних d_i :

$$x_i : \{d_i\} \rightarrow \{d_i\}. \quad (5)$$

Часова складність l_i перетворювача прийнята рівною одиниці:

$$\forall i, l_i = 1. \quad (6)$$

Елементарний перетворювач представляється у вигляді чорної скриньки. Формальне визначення операції перетворення даних елементарним перетворювачем можливе із застосуванням будь-якої відомої моделі алгоритму, включаючи *SH*-модель. Звідси випливає, що поняття "*SH*-модель" і "елементарний перетворювач" мають ієрархічний зміст. Тобто, для *SH*-моделі справедлива властивість ієрархічності. Відзначимо, що абстрактні моделі обчислювачів такої властивості не мають.

Використання елементарних перетворювачів в складі моделі алгоритму робить необхідним врахування системи зв'язків $U = \{U_{ij} | i = 1, j\}$ між ними. Без цієї системи зв'язків дослідження апаратної реалізації алгоритму буде неповним.

Конфігурація зв'язків впливає на інші характеристики комп'ютерних засобів, зокрема, змінює значення апаратної і програмної складностей. Звідси наступна вимога: врахування конфігурації зв'язків або структурної складності - додаткової характе-

тики *SH*-моделі.

Визначення *SH*-моделі

SH-модель є фіксованою для деякого класу задач конфігурацією апаратно-програмних засобів перетворення, передавання і зберігання даних, що задає обчислювальний процес, який починається з деякої системи початкових даних і скерований на отримання результату, повністю визначеного цими початковими даними.

Це визначення відрізняється від вербального тлумачення алгоритму тим, що замість слів "точно приписання" використані слова комп'ютерної термінології – "...фіксована для деякого класу задач конфігурація апаратних засобів перетворення, передавання і зберігання даних..."

Формальний опис *SH*-моделі наступний: *SH*-модель – це сімка

$$B = \langle D, Q, q_0, q_f, G, P, M \rangle, \quad (7)$$

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;

$$G = (X, U), \quad (8)$$

де X – множина елементарних перетворювачів; U – множина між'єднань;

$$P = \{ D \} \times \{ Q \} \times \{ X \} \times \{ \rightarrow \} \times \{ D \} \times \{ Q \}, \quad (9)$$

P – така програма, що не може мати трьох однакових символів в (9); M – зовнішня віртуальна пам'ять.

Структура *SH*-моделі

Перевагою машини Тьюрінга є наявність незмінної структури (каркасу), яка зберігається для всіх алгоритмів, що моделюються. Це зумовило використання у визначенні моделі алгоритму слова "машина". Використання "каркасу" є зручним для побудови правила безпосереднього перероблення. Інші моделі алгоритмів не мають постійної структури. *SH*-моделі також 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:

$$E = \{ e_1, e_2 \dots e_n \}; \quad U = \{ u_1, u_2 \dots u_m \}. \quad (10)$$

Кожний елементарний перетворювач виконує деяку операцію. Ці операції можуть бути різними для різних перетворювачів. До множини елементарних перетворювачів належать також елементи тимчасової пам'яті. Вони відображають реальні схмотехнічні і системотехнічні пристрої – тригери, регістри, кеш та інш. Вибір ієрархічного рівня елементів тимчасової пам'яті відповідає ієрархічному рівню комп'ютерних схем, на якому реалізується алгоритм.

Властивості SH-моделі

SH-модель має всі властивості, які випливають 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. Слово "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.
- Спрямованість. Якщо процес перетворення даних не може відбуватися за конфігурацією зв'язків і програмою, то повинна бути вказівка, що робити в такій ситуації.
- Масовість. Ця властивість має дворівневий зміст:
 - 1) універсальність - одна і та ж SH-модель може бути застосована для розв'язання будь-якої задачі, що формалізується;
 - 2) масовість у традиційному тлумаченні – одна і та ж 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.

Параметри SH-моделі

Параметри алгоритму – правило початку, правило закінчення, – задаються програмою. Системи проміжних і кінцевих результатів, правило безпосереднього перероблення задаються апаратними або апаратно-програмними засобами. Система початкових даних задається зовнішніми відносно до SH-моделі пристроями пам'яті.

Характеристики складності SH-моделі. В процесах синтезу, аналізу і оптимізації SH-моделей пропонується використовувати чотири характеристики складності: апаратну, часову, програмну і структурну.

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

$$A = |X|, \quad (12)$$

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 ін.

Разом з тим при оцінці апаратної складності можливий інший підхід. Апаратну складність мікропроцесорів можна визначати кількістю транзисторів, розташованих на кристалі. Постійне збільшення апаратної складності (а зараз вона досягає 100 млн. транзисторів) створює сприятливі умови для розширення функціональних можливостей, для покращання практично всіх споживчих характеристик процесорів, зокрема точності обчислень, збільшення продуктивності (без збільшення фізичних розмірів комп'ютерів, розширення функцій).

Наведене визначення апаратної складності не протирічить поняттю "об'єм обладнання", що використовується в обчислювальній техніці.

Часова складність

В метричній теорії під часовою складністю розуміють кількість елементарних операцій, наприклад, кроків машини Тьюрінга. В теорії *software/hardware* часова складність визначається дещо інакше.

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

$$L = |\max X_i|, \quad (13)$$

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 формулі:

$$T = \sum_{\tau_i \in \max X_i} \tau(e_i), \quad (14)$$

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

Апаратна і часова складності описують властивості комп'ютерних засобів, які прийнято відносити до технічних характеристик. *SH*-моделі, крім апаратної і часової, характеризуються додатково програмною і структурною складностями.

Програмна складність

Проаналізуємо часову діаграму роботи деякого пристрою на рівні регістрових передач з точки зору інформації, що міститься в ній. Часова діаграма являє собою двомірну таблицю, на осі абсцис якої позначені дискрети часу, на осі ординат – входи керування пристроями функціональних схем.

На процесорному рівні кожній асемблерній команді відповідає власна мікропрограма (часова діаграма). Для кожної мікропрограми конфігурація положень сигналів керування фіксована. Використовуючи логарифмічну міру ступеня нерегулярності (ентропії) часової діаграми, запишемо:

$$P = -F \log_2 \frac{F}{n \cdot m}, \quad (15)$$

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.

Умова вибору фрагментів така:

$$\forall i, j; i \neq j \{ [\varphi, \Pi\varphi, \neq \varphi, \vee \varphi,] \Rightarrow |f_i| = |\varphi_i| + |\varphi_j| \} \vee \{ [\varphi, \Pi\varphi, = \varphi, \vee \varphi,] \Rightarrow |f_i| = |\varphi_i| \}, \quad (16)$$

де φ_i, φ_j - фрагменти часової діаграми.

Структурна складність SH -моделі визначається аналогічним способом, що і програма. Відмінність лише в об'єкті розрахунків. Одержання структурної складності проводиться в три етапи:

1. Схема SH -моделі перетворюється в оргграф.
2. Оргграф кодується у вигляді матриці інциденції.
3. Розраховується значення нерівномірності матриці інциденції.

Місткісна складність SH -моделі дорівнює кількості комірок зовнішньої пам'яті, яка потрібна для розв'язання даної задачі φ_i, φ_j .

1. Математическая энциклопедия. Под ред. И.М. Виноградова. - М.: Советская энциклопедия, 1977. - в пяти томах.
2. Марков А.А. Теория алгорифмов. - М.- Л.: Изд-во АН СССР, 1954 - 375 с. - (Труды МИАН. Т. 42).
3. Ахо А., Холкрофт Дж., Дж. Ульман Дж.. Построение и анализ вычислительных алгоритмов. - М.: Мир, 1979. - 536 с.
4. Успенский В.А., А.Л. Семенов А.Л.. Теория алгоритмов: основные открытия и приложения. - М.: Наука, 1987. - 288 с.

УДК 621.397.3

РЕДАКТОР СИНТЕЗУ ТА МОДЕЛЮВАННЯ СКЛАДНИХ ЗОБРАЖЕНЬ СИМЕТРИЧНОЇ СТРУКТУРИ

© К. Березька, О. Березький

Тернопільська академія народного господарства, м. Тернопіль

Описано спроектований авторами редактор орнаментів. Наведено приклад синтезу та здійснено оцінку архівування орнаментів за допомогою створеного редактора.