

Про процедурну платформу для теорії алгоритмів

Віталій Зубенко

к.ф.-м.н., доцент, КНУ ім. Тараса Шевченка, вул. Володимирська, 60, Київ, 01033, e-mail: vvz1952@gmail.com

Робота присвячена спробі математичного уточнення загального поняття алгоритму. Пропонується відійти від існуючої в теорії алгоритмів практики визначати якийсь конкретний представницький клас алгоритмів¹ і пов'язувати з ним загальне поняття алгоритму. Наш підхід базується на ідеї математичного уточнення більш абстрактного і ширшого поняття - обчислювальної процедури [2], а потім у звуженні його до поняття алгоритму. Вводиться поняття оперативної таблиці процедур і в основу звуження покладена розв'язність цих таблиць. Розглядається місце основних математичних моделей алгоритмів та обчислень в межах запропонованої процедурної платформи таких як скінченні автомати, машини Тюринга, нормальні алгоритми, дискретні перетворювачі Глушкова, формальні граматики, системи Поста та інші. Як з'ясувалося, майже всім цим моделям відповідають процедури зі скінченними оперативними таблицями. Обговорюються питання упорядкування та систематизації всіх існуючих на сьогодні моделей алгоритмів та числень в межах даної платформи.

Ключові слова: алгоритм, обчислювальна процедура, алгоритмічна процедура, алгоритмічне числення, оперативна таблиця.

Вступ. Причина відсутності в теорії алгоритмів загального визначення алгоритму – на поверхні. Існуючі моделі алгоритмів нерозривно пов'язані з конкретними фінітними мовами їхнього подання, а скільки моделей – стільки й мов. Щоб обійти цю першопричину, введемо абстрактне поняття процедури, а потім сформулюємо місце алгоритмів серед процедур.

1. Покладемо $T = \{0, 1, \dots\}$, $T_n = [0..n]$, S та $S_0 \subseteq S$ - сукупності станів та вхідних станів. Трійка $\Pi = (T, S, S_0)$ називається обчислювальним простором.

Нехай S^* , та S^∞ – сукупності відповідно усіх скінченних та нескінченних процесів $p: T_n \rightarrow S, p: T \rightarrow S$, які подаються словами $s_0 s_1 \dots s_n$ або $s_0 s_1 \dots$ алфавіті S . Початкові скінченні відрізки процесів будемо називати їхніми префіксами.

Серед усіх процесів виділимо обчислювальні процеси, в яких поява чергового стану не довільна, а є результатом певної елементарної операції над його крайнім станом. Зафіксуємо певну сукупність $\Delta = \{f_i: S \rightarrow S, i = 0, 1, \dots\}$ часткових елементарних операцій на станах. Позначимо 2^Δ множину всіх скінченних підмножин Δ . Оперативна функція має вигляд $\delta: S^+ \rightarrow 2^\Delta$ і

¹ Тобто такий, що в ньому є еквівалентний алгоритм для кожного інтуїтивного алгоритму [1].

регулює порядок застосування операцій в обчисленнях. *Обчислювальною процедурою* у просторі Π із сукупністю елементарних операцій Δ та оперативною функцією δ називається трійка $P = (\Pi, \Delta, \delta)$.

Кожна обчислювальна процедура породжує певну сукупність обчислень за процедурою P , які визначаються рекурентно: $p_0 = s$ і для всіх $t > 0$ $p_t = f_t(p_{t-1})$, де $s \in S$ - початковий стан обчислення, а f_t - одне зі значень оперативної функції δ на префіксі $p_0 \dots p_{t-1}$.

До найпростіших видів процедур відносяться *темпоральні*, в яких значення оперативної функції $\delta(p_0 \dots p_{t-1})$ визначається просто за номером кроку обчислення та *автоматні*, в яких це значення залежить тільки від останнього стану обчислення, тобто $\delta(p_0 \dots p_{t-1}) = \delta(p_{t-1})$. Більшість класичних алгоритмічних систем є саме автоматними.

Основних механізмів завершення обчислень - два. Коли в процесі обчислення оперативна функція чи елементарна операція не визначена або крайній стан належить до виділеної підмножини $S_f \subseteq S$ заключних станів.

Крайній стан обчислення з початковим станом s позначається $P^*(s)$. Він може бути не один. Про відповідності $P^*: S_0 \rightarrow S$ та $P^*: S_0 \rightarrow S_f$ говорять, що їх *обчислює* процедура P , а саму процедуру називають *процедурою-функцією*.

В гіперобчисленнях процес може продовжуватися і після переходу в заключний стан. Тоді результат $P^{**}(s)$ складають усі його вихідні стани.

Часто цікавлять не самі стани, а тільки ті чи інші їхні складові і тоді розглядають не саму відповідність, яку обчислює процедура, а її проєкцію на ці складові. Зустрічаються такі обчислення, коли для отримання наступного його стану є необхідність звернутися не тільки до поточного стану, а й до деяких інших. Це спонукає перейти від унарних операцій до k -арних, $k \geq 1$, вигляду $f^k: S \times \dots \times S \rightarrow S$, які, на відміну від елементарних операцій, можуть бути і багатозначними. Нехай множина Δ є сукупністю k_i -арних операцій $f_i^{k_i}$. Вхідні стани тепер називаються – *аксіомами*, а функція керування $\delta: S^* \rightarrow 2^\Delta$ тепер визначена і на порожньому префіксі. Четвірка $P = (\Pi, \Delta, \delta, S_f)$ називається *процедурою-численням*. Наступний стан обчислення p_t в численнях визначається так. Для $f_i^{k_i} \in \delta(p_0 \dots p_{t-1})$, ($f_i^{k_i} \in \delta(\varepsilon)$ для нульового кроку) з попередніх станів p_0, p_1, \dots, p_{t-1} та аксіом формується довільний кортеж (q_1, \dots, q_{k_i}) з області визначення функції $f_i^{k_i}$ і тоді $p_t = f_i^{k_i}(q_1, \dots, q_{k_i})$. Мають місце гіперобчислення. Крайні стани і проміжні заключні складають теорію $Th(P)$ числення P .

Ми побудували загальну теоретико-множинну модель поняття обчислювальної процедури. Тепер, щоб отримати математичне визначення

алгоритму, необхідно формально виокремити алгоритми серед всіх обчислювальних процедур.

2. Алгоритми мають справу тільки з конструктивними об'єктами і самі є такими об'єктами. Отже, всі стани алгоритмічної процедури, елементарні операції та оперативна функція мають бути конструктивними в межах певної фінітної дескриптивної системи. Конструктивність функції означає, що існує її фінітний опис, який дозволяє за скінченну кількість кроків знайти значення функції за даними аргументами.

Нагадаємо, що два елементи функції ядро еквівалентні, якщо значень функції на цих елементах збігаються. Індексом відношення еквівалентності називається потужність множини його фактор-класів. Ядерна еквівалентність оперативної функції завжди має скінченний індекс, якщо множина елементарних операцій Δ скінченна. Операційну функцію можна задати таблично – *оперативною таблицею*, в першому стовпчику якої перераховані всі класи ядерної еквівалентності, у другому – її значення на префіксах обчислень даного класу. Отже, щоб знайти значення оперативної функції достатньо знайти рядок таблиці, до якого належить префікс і взяти одну з операцій з другого стовпчика. Оперативну таблицю назвемо *розв'язною*, якщо існує алгоритм, який дозволяє знайти в ній рядок, до якого належить довільний префікс або дати відповідь, що такого рядка немає. Розв'язність оперативної таблиці гарантує існування скінченного опису оперативної функції. Отже, ми готові сформулювати, що таке алгоритм в термінах обчислювальних процедур.

Алгоритм – це обчислювальна процедура з конструктивними станами і такими ж елементарними операціями та розв'язною оперативною таблицею.

Тут не виникає «зачароване коло», пов'язане з розв'язністю оперативної таблиці, тому що мова йде не про загальну розв'язність множин, а тільки про окремі її часткові випадки, кожен з яких в разі потреби може бути математично уточненим без апеляції до загального поняття. Далі алгоритми як алгоритмічні процедури будемо називати *A-процедурами*.

Якщо звернутися до існуючих алгоритмічних систем та числень, то можна помітити що: 1) вони, як A-процедури є в основному автоматними (за винятком неавтоматних числень) і 2) для переходів визначальними є стани не в усьому їхньому об'ємі, а тільки вмісти певних обмежених («активних» за [1]) їхніх структурних фрагментів. Збіг вмістів цих обмежених фрагментів лежить в основі ядерної еквівалентності більшості оперативних функцій A-процедур.

Розглянемо для прикладу машини Тюрінга (МТ). МТ має скінченну кількість внутрішніх станів і вхідну стрічку, в якій розміщується вхідне слово з X^* , один з символів якого є доступним для обробки. В процесі обчислення МТ може змінити внутрішній стан, замінити доступний символ стрічки на інший і залишити його доступним або зробити доступним символ справа чи зліва від нього. МТ завжди зупиняється при переході в заключний внутрішній стан (відсутні гіперобчислення). Заключне слово на стрічці є результатом обробки МТ вхідного слова. Станами A-процедури МТ є пари (внутрішній стан МТ, слово на стрічці з виділеною доступною коміркою). В початкових станах

внутрішній стан МТ є початковим, в заключних – заключним. Функція-проекція повертає вміст стрічки стану. Оперативна функція А-процедури відповідає функції переходів МТ і є скінченою та автоматною. Два стани А-алгоритму МТ ядерно еквівалентні, якщо їхні внутрішні стани та вмісти доступних комірок на стрічках збігаються. Подібні А-процедурні еквіваленти можуть бути побудовані і для решти існуючих алгоритмічних систем.

Загальні властивості А-процедур мають свою інтерпретацію в усіх конкретних алгоритмічних системах і не потребують в останніх додаткових доведень. Наприклад, відомо про існування певної регулярної алгебраїчної форми для функцій, обчислювальних А-процедурами [4]. Існування подібних форм для випадку функцій, обчислюваних машинами Тюрінга і дискретними перетворювачами свого часу було доведено в роботах Якопіні та Глушкова. Це також стосується замкнення класів обчислювальних функцій відносно основних функціональних композицій та теоретико множинних операцій.

Висновки. Отже, розглянута процедурна платформа для теорії алгоритмів, в основу якої покладено поняття А-процедури. Проілюстровано як дана платформа може бути використана для упорядкування існуючих алгоритмів та числень та вивчення їхніх загальних властивостей. А-процедури можуть бути покладені в основу розбудови та дослідження нових моделей обчислень.

Література

- [1] Успенский В.А. Теория алгоритмов: ее основные открытия и приложения/ Семенов А.Л.- В кн. Алгоритмы в современной математике и ее приложениях/ под.ред. А.П.Ершова, Д. Кнута.-Н.: ВЦ СО АН СССР.- 1982.-Ч.1.- с. 99-342.
- [2] Зубенко В. В. Програмування: Навчальний посібник /Л.Л. Омельчук.– К.: ВПЦ «Київський університет», 2011. - 625 с.
- [3] Зубенко В.В. Темпоральні процедури та алгоритми // Проблеми програмування , 2006, № 2-3, с. 53-59.

On a procedural platform for the theory of algorithms

Vitaly Zubenko

The work is devoted to an attempt to mathematically clarify the general concept of an algorithm. It is proposed to move away from the practice existing in the theory of algorithms to define a specific representative class of algorithms and associate the general concept of an algorithm with it. Our approach is based on the idea of mathematically specifying a more abstract and broader concept - a computational procedure[2], and then in narrowing it down to the notion of an algorithm. The concept of operational table of procedures is introduced, and the basis of narrowing is the solvability of these tables. The place of basic mathematical models of algorithms and calculations within the proposed procedural platform, such as finite automata, Turing machines, normal algorithms, discrete Glushkov transforms, formal grammars, Post systems, and others, is considered. As it turns out, procedures with finite operational tables correspond to almost all of these models. Issues of arrangement and systematization of all currently existing models of algorithms and calculus within the framework of this platform are discussed.

Отримано 15.03.23