

УДК 621.3

М.В. ЧеркаськийНаціональний університет “Львівська політехніка”,
кафедра “Електронні обчислювальні машини”**ЕВОЛЮЦІЯ ТЛУМАЧЕННЯ ПОНЯТТЯ “АЛГОРИТМ”**

© Черкаський М.В., 2003

Розглянуто еволюцію поняття “алгоритм”, його моделей, властивостей, параметрів. Показана необхідність побудови моделі алгоритмів на основі апаратно-програмного обчислювача. Дано його вербальне формулювання.

The article examines evolution of the term “algorithm”, its models, features, and parameters. It demonstrates a need for construction of algorithm models using a hardware-software calculator. It provides its verbal formulation.

Вступ. Те, що зараз ми розуміємо під словом “алгоритм”, використовувалося в глибокій давнині. Задовго до утворення і роботи Олександрійської бібліотеки, яка, за словами академіка А.Колмогорова, відіграла в античному світі роль першого науково-дослідного інституту, були відомі правила виконання арифметичних операцій в Індії, знаходження чисел за їх залишками в Китаї. Але особливе значення для зародження поняття “алгоритм” як фундаментального об’єкта математики мають праці Евкліда (біля 300 р. до Р.Х.) і аль-Хорезмі (біля 850 р.).

Праці “Начала” Евкліда були настільною книгою математиків протягом більш як двох тисячоліть. І зараз аксіоми з “Начал” є необхідним елементом загальної культури людини. У “Началах” вперше описано широковідому процедуру знаходження найбільшого спільного дільника двох чисел, яка значно пізніше отримала назву “алгоритм Евкліда”. Його значення для теорії алгоритмів не обмежується лише тим, що він повністю задовольняє властивості алгоритму, відповідає всім його параметрам. Цей алгоритм є прикладом високоефективного за часовою складністю алгоритму. Порівняно з альтернативним алгоритмом перебору можливих розв’язань, складність якого дорівнює $O(N)$, алгоритм Евкліда характеризується складністю лише $\sim O(\log_2 N)$ [1].

Мухамед ібн Муса з Хорезму, за арабським ім’ям – аль-Хорезмі (походженням з середньоазіатського міста Хорезм), видатний багдадський вчений, що працював у 9-му столітті н.е., у своїй книжці-трактаті “Про індійський рахунок” описав позиційну десяткову систему числення і алгоритми реалізації за її допомогою арифметичних операцій множення і ділення, підсумування, віднімання та інших. Сьогодні збереглися лише переклади трактату. Перші з них відносяться до початку 12-го століття. Кожний розділ перекладу, а іноді навіть абзаци, починалися словами “Сказав Альгорізм...”. Це словосполучення використовували у своїх лекціях професори середньовічних університетів. Поступово ім’я аль-Хорезмі набуло звучання “алгорізм”, “алгоритм” і навіть перетворилися у назву нової арифметики. За довгу еволюцію слова “алгоритм” було втрачено джерело його виникнення. І тільки у 1849 році сходознавець Ж.Рейно повернув нам ім’я аль-Хорезмі [2].

Алгоритми Евкліда і аль-Хорезмі є прикладами ефективного розв’язання арифметичних задач. Вони ілюструють використання алгоритмів, але не дають тлумачення поняття “алгоритм”, яке існує як самостійний, незалежний від конкретних прикладів,

математичний об'єкт. Пройшло декілька століть перед тим, як поступово почало викристалізовуватися це важливе поняття. Але й тепер не можна сказати, що цей процес завершився. Проблема полягає у наближенні поняття “алгоритм” до здобутків прикладної теорії обчислень, перетворення і зберігання даних. Все те, що було зроблено у межах класичної теорії алгоритмів в докомп'ютерну епоху, потрібно пов'язати з досягненнями сучасних комп'ютерних наук. Саме цій проблемі, що розглядається в еволюційному плані, присвячена стаття.

Перші формулювання. Перший період формулювання поняття алгоритму належить до часу від 13-го до кінця 19-го століть. Д.Кнут в [3] посилається на одне з ранніх джерел, у якому наведено таке тлумачення алгоритму: “Під цією назвою об'єднані поняття о чотирьох арифметичних діях, а саме о сумуванні, множенні, відніманні і діленні”, що безпосередньо пов'язано з працями аль-Хорезмі. Ця назва не несе в собі посилянь на властивості, характеристики, параметри, пізніше вона була замінена на значно ширше поняття – “арифметика”. Перше тлумачення сутності терміну “алгоритм” належить до періоду 16–17-го століть і пов'язано з працями Хр.Рудольфа (1525 р.) і Лейбніца (1684 р.). Потреба в тлумаченні алгоритму була викликана, зокрема, пошуком єдиного універсального способу розв'язання будь-якої задачі [2]. У адаптованій до сучасної термінології формі це тлумачення виглядає так: “Алгоритм позначає будь-який регулярний обчислювальний процес, який за кінцеву кількість кроків розв'язує задачі визначеного класу”. Тут чітко простежуються три властивості алгоритму: детермінованість (регулярність), дискретність (кінцева кількість кроків) і масовість, що впливає зі слів “...розв'язує задачі визначеного класу”. Крім того, фіксується параметр алгоритму – правило закінчення.

Отже, завершальне тлумачення першого етапу базувалося на відображенні властивостей обчислювального процесу – детермінованості, дискретності і масовості. Властивість “елементарність” не згадувалась.

Уточнення поняття “алгоритм”. Наприкінці 19-го і на початку 20-го століть в математиці виникло ряд проблем, розв'язання яких вимагало уточнення розуміння, що таке алгоритм.

У процесі дослідження теорії множин, математичної логіки були відкриті антимонії: парадокс Кантора (1899р), парадокс Рассела (1902 р.). Одночасно Гілберт сформулював двадцять проблем, серед яких десята проблема вимагала доведення наявності або відсутності алгоритму розв'язання діафантового рівняння в цілочислових коефіцієнтах [4]. Потім було виявлено нові класи задач, що теж вимагали дати відповідь, чи мають задачі цього класу розв'язок. Отже, поряд з проблемою антимоній інтенсивно досліджувалась проблема розв'язності. Проблеми виявилися складними. Їх розв'язання звичайними математичними методами були ненадійними з точки зору можливості виникнення помилок і неефективними за витратами часу. Виникла необхідність побудови нового математичного апарата і засобів досліджень. Були прийняті дві принципові умови, загальні для будь-яких конкретних математичних побудов:

- 1) кількість операцій, яка необхідна для розв'язання конкретної проблеми, повинна бути кінцевою;
- 2) операції повинні бути елементарними, щоби уникнути можливих помилок на складному ланцюгу інтуїтивних переходів в процесі розв'язання проблеми.

Слово “елементарність” математично не визначалось і в даному контексті еквівалентне поняттю “простота і локальність”. А.Марков, крім того, надає цьому терміну

зміст “загальна зрозумілість” [5]. “Загальна зрозумілість”, “простота і локальність” не мають математичного змісту. Саме ця обставина сприяла появі багатьох різних напрямків подальшого уточнення поняття “алгоритм” і використання його для побудов систем дослідження проблем антимоній і розв’язності. Першою такою системою був апарат рекурсивних функцій.

Рекурсивні функції були введені Гьоделем (1931 р.) спеціально для дослідження антимоній. До їх складу входять найпростіші функції: слідування за, проектування тотожної рівності нулю, а також ще три базових функції: підстановки, примітивної рекурсії та μ -операції [6]. Ці функції прийнято вважати елементарними. З досвіду спілкування зі студентами, що вивчають теорію алгоритмів, можна дійти висновку, що засвоєння апарата рекурсивних функцій потребує значно більших зусиль, ніж засвоєння операцій з використанням таких моделей алгоритмів, як машина Тюрінга, Поста, формальні алгоритми Маркова та інші. Разом з тим доведено, що дослідження антимоній за допомогою рекурсивних функцій є значно легшим порівняно із звичайним математичним апаратом. Через простоту операцій ймовірність появи помилок при виявленні і дослідженні антимоній зменшується.

Моделі алгоритмів. Другий напрямок дослідження проблем розв’язності ґрунтувався на використанні моделей алгоритму – формальних алгоритмічних систем (ФАС). Одну з перших моделей запропонував Алан Тюрінг. Вона отримала назву “машина Тюрінга”.

Машина Тюрінга уточнює інтуїтивне поняття алгоритму, надає йому точного математичного визначення. Машина Тюрінга повністю відповідає наведеному тлумаченню алгоритму, вона володіє всіма властивостями і має вісім параметрів – правило початку, правило вводу, правило виводу, правило безпосереднього перероблення, правило закінчення, систему вхідних даних, потенційно нескінченну систему проміжних результатів, систему кінцевих результатів. Машина Тюрінга моделюється декількома простими операціями: читання символу з рядка, знаходження і читання команди в програмі, реалізація команд, запис або стирання нового символу, пересування головки вправо, вліво або залишання на місці. Перевагою над рекурсивними функціями з точки зору виконання вимоги елементарності є, крім зрозумілості підоперацій, також однотипність кроків для будь-яких алгоритмів.. Фіксованість кроку машини Тюрінга дозволила порівнювати різні алгоритми, сформулювати початкові визначення теорії складності – дати поняття часової, місткісної і програмної складностей. Машина Тюрінга, як і більшість інших моделей (багато з них є варіантами машини Тюрінга, наприклад, машина з необмеженими регістрами), орієнтована на дослідження обчислювальних операцій. Складання програми для машини Тюрінга є простим і зрозумілим. Але моделювання не зв’язаних з обчисленнями задач, наприклад задач кодування/декодування, є відносно складним.

Нормальні алгоритми Маркова також уточнюють поняття “алгоритм”, дають визначення кроку алгоритму через дві взаємозв’язаних операції, що повторюються через кінцеву кількість циклів – розпізнавання і підстановки. Простота і зрозумілість очевидні. Операції розпізнавання і підстановки нагадують дитячі ігри – складання кубиків. Тому ці дві операції можна вважати загальнозрозумілими. Намагання зробити крок алгоритму найближчим до визначення: “простота і локальність”, а також “загальна зрозумілість” не суперечать простоті розв’язання задач кодування/декодування, але, на відміну від машини Тюрінга, ускладнює логіку обчислень.

Крім описаних, існує декілька інших математичних моделей, що уточнюють поняття “алгоритм”, але вони не вносять суттєвої новизни в предмет досліджень.

Отже, у першій половині 20-го століття розвиток теорії алгоритмів був зумовлений проблемою розв'язності задач. Було зроблено декілька уточнень стосовно поняття алгоритм, створено низку математично точно описаних моделей, сформульовано вербальне тлумачення алгоритму, введено параметри, описано властивості, у тому числі “елементарність” та “спрямованість”, було введено характеристики складності: часова, програмна, місткісна. Завершальним на цьому шляху можна прийняти таке тлумачення, яке визначається переліком властивостей алгоритму і підтримувалося А.Колмогоровим [7] і А. Марковим [5].:

“а) алгоритм – це процес послідовної побудови величин, який проходить в дискретному часі так, що в початковий момент задається початкова скінченна система величин, а в кожний наступний момент система величин втримується за певним законом (програмою) із системи величин, які були в попередній момент часу (дискретність алгоритму).

б) система величин, які отримуються в якийсь (не початковий) момент часу, однозначно визначається системою величин, отриманих в попередні моменти часу (детермінованість алгоритму).

в) закон отримання наступної системи величин із попередньої повинен бути простим і локальним (елементарність кроків алгоритму).

*г) якщо спосіб отримання наступної величини із якої-небудь заданої величини не дає результату, то повинно бути вказано, **що** потрібно вважати результатом алгоритму (спрямованість алгоритму).*

д) початкова система величин може вибиратися із деякої потенційно нескінченної множини (масовість алгоритму)”.

Алгоритм – припис. Бухливий розвиток комп'ютеризації в останню чверть 20-го століття дав потужний поштовх утворенню нового напрямку інженерної діяльності – програмуванню. В цих умовах програма як одна з форм алгоритму не могла сприйматися як процес. Більш природно сприймається термін “припис”. У результаті тлумачення алгоритму набуло такого вигляду:

”Алгоритм – точний припис, який задає обчислювальний процес (який називається в цьому випадку алгоритмічним), що починається з довільного початкового даного (з деякої сукупності можливих для даного алгоритму початкових даних) і спрямований на отримання результату, який повністю визначається цим початковим даним“[5].

У цьому тлумаченні суттєвим є те, що алгоритм розглядається не як процес, а як “припис, що задає ...процес...” Але, як і в інших тлумаченнях минулого століття, алгоритм передбачає використання абстрактного обчислювача, який у своєму складі не передбачає технічних засобів. Головка, механізм її пересування, вузли пошуку команди в пам'яті програми, сама пам'ять – це уявні елементи, що не існують в визначенні машини Тюрінга. Тому реально побудувати машину Тюрінга неможливо, немає з чого будувати. Те ж саме стосується і нормальних алгоритмів Маркова. Вузли – розпізнавач і перетворювач – існують лише як математичні абстракції.

Варіанти наведеного тлумачення широко використовуються в сучасній літературі [9, 10, 11].

Апаратно-програмне тлумачення. Теорія алгоритмів, побудована на основі абстрактного обчислювача, не дозволяє повною мірою використати досягнуті здобутки для синтезу, аналізу і оптимізації комп'ютерних засобів. Більше того, деякі висновки суперечать практиці комп'ютерних обчислень, наприклад, теорема про лінійне прискорення [12]. Наближення теорії алгоритмів до потреб комп'ютерної техніки є необхідним – у цьому

напрямку зроблено багато. Останнє тлумачення “алгоритм – це припис...” дозволяє під словом “припис” розуміти апаратні або апаратно-програмні засоби, як прийнято у фахівців з комп’ютерної інженерії. Звідси випливає твердження.

Твердження. *“Алгоритм можна реалізувати апаратними або апаратно-програмними засобами, але не можна реалізувати лише програмно”.*

Розглянуті формулювання поняття “алгоритм” і попереднє твердження приводять до такого тлумачення:

“Алгоритм – це фіксована для розв’язання деякого класу задач конфігурація апаратно-програмних засобів перетворення, передавання і зберігання даних, що задає обчислювальний процес, який починається з деякої системи початкових даних (потенційно нескінченної) і скерований на отримання результату, повністю визначеного цими початковими даними”.

Це визначення відрізняється від попереднього тлумачення тим, що замість слів “точний припис” використані слова комп’ютерної термінології – “...фіксована для деякого класу задач конфігурація апаратних засобів перетворення, передавання і зберігання даних...”. Останнє тлумачення не суперечить переліку властивостей алгоритму, воно має всі визначені параметри і дозволяє будувати моделі алгоритму для дослідження і оптимізації апаратно-програмних засобів сучасних комп’ютерних систем. Воно дозволяє побудувати апаратно-програмну модель (SH-модель) алгоритму, розширити перелік властивостей і характеристик складності комп’ютерних засобів [13].

Разом з тим останнє тлумачення алгоритму стосується лише комп’ютерного розв’язання задач. Воно не враховує різноманітних алгоритмів, що реалізуються за допомогою природного інтелекту. Тому формула “Алгоритм – припис...” залишається зручним інструментом математичних досліджень.

Висновки

1. Еволюція інтуїтивного тлумачення алгоритму як фундаментального поняття математики протягом тисячоліть була пов’язана з необхідністю розв’язання конкретних математичних проблем.

2. Принциповими пунктами розвитку поняття “алгоритм” було формування переліку властивостей, параметрів і характеристик алгоритму.

3. Апаратно-програмне тлумачення поняття “алгоритм” може стати корисним в процесі розроблення ефективних апаратно-програмних комп’ютерних засобів.

1. Черкаський М.В., Мітьков В.С. *Історичний аспект складності алгоритму*// Вісник Нац. ун-ту “Львівська політехніка”. – 2002. – №463. – С.111–118. 2. Юркевич А.П. *История математики в средние века.* – М., 1961. 3. Кнут Д. *Искусство программирования для ЭВМ, Т. 1* – М., 1976. 4. Трахтенброт Б.А. *Алгоритмы и вычислительные автоматы.* – М., 1974. 5. Марков А.А. *Теория алгорифмов.* – М. – Л.; 1954. – (Труды МИАН. Т. 42). 6. Мальцев А.И. *Алгоритмы и рекурсивные функции.* – М., 1986. 7. Успенский В.А., Семенов А.Л. *Теория алгоритмов: основные открытия и приложения.* – М., 1987. 8. *Математическая энциклопедия.* / Гл. ред. И.М. Виноградов. Т. 1. – М., 1977. 9. Microsoft Press. *Толковый словарь по вычислительной технике: Пер. с англ.* – М., 1995. 10. Кормен Т., Лейзерсон Ч., Ривест Р. *Алгоритмы: построение и анализ: Пер. с англ.* – М., 2001. 11. Седжвик, Р. *Фундаментальные алгоритмы на C++. Алгоритмы на графах: Пер. с англ.* – М., С-П., К., 2002. 12. Катленд Н. *Вычислимость. Введение в теорию рекурсивных функций.* – М., 1983. 13. Черкаський М.В. *SH-модель алгоритму*// Вісник Нац. ун-ту “Львівська політехніка”. – 2001. – №433. – С.127–134.