МЛТА Лекція 15.02.2021

Теорія алгоритмів. Становлення

Теорія алгоритмів – наука, яка вивчає загальні властивості закономірності алгоритмів, різноманітні формальні моделі їх представлення. Теорія алгоритмів є теоретичною основою програмування та інформатики. До задач теорії алгоритмів відносяться формальне доведення алгоритмічної нерозв'язності задач, асимптотичний аналіз складності алгоритмів, класифікація алгоритмів відповідно до класів складності, розробка критеріїв порівняльної оцінки якості алгоритмів тощо.

Теорія алгоритмів як окремий розділ математики виникла в 30-х рр. XX ст. Необхідність точного математичного коректування інтуїтивного поняття алгоритму стала неминучою після усвідомлення неможливості існування алгоритмів розв’язку багатьох масових проблем, у першу чергу, пов’язаних з арифметикою та математичною логікою (проблеми істинності арифметичних формул і формул числення предикатів першого порядку, 10-та проблема Гілберта про розв'язність діофантових рівнянь тощо).

Для доведення відсутності алгоритму розв’язання певної задачі необхідно мати точне математичне визначення алгоритму, тому після сформування поняття алгоритму як нової та окремої сутності першочерговою постала проблема знаходження адекватних формальних моделей алгоритму й дослідження їх властивостей. Таким чином, формальні моделі були запропоновані як для первісного поняття алгоритму, так і для похідного поняття алгоритмічно обчислюваної функції.

Першими формальними моделями алгоритмічно обчислюваних функцій були ***λ-значні функції*** (А. Чорч, 1932) та ***загальнорекурсивні функції*** (К. Гедель, 1934). Вказані класи визначались як функції, графіки яких породжуються відповідно до числення λ-конверсій і числення Ербрана-Геделя.

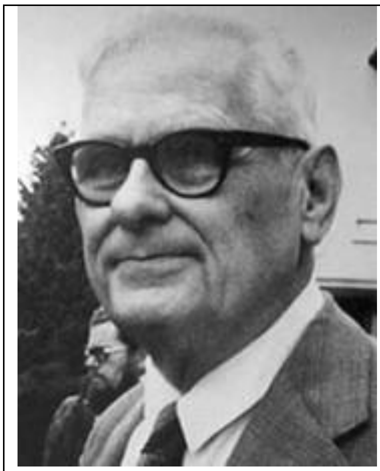
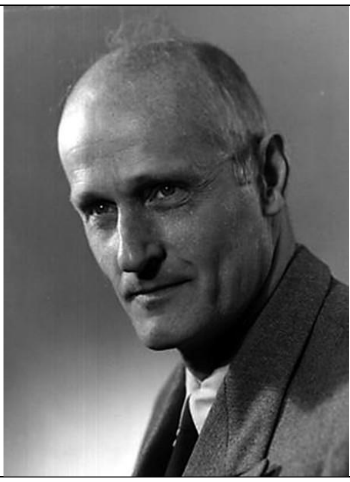
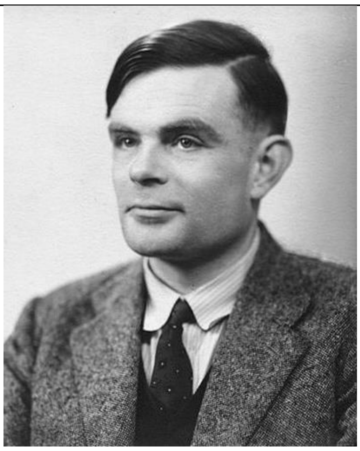
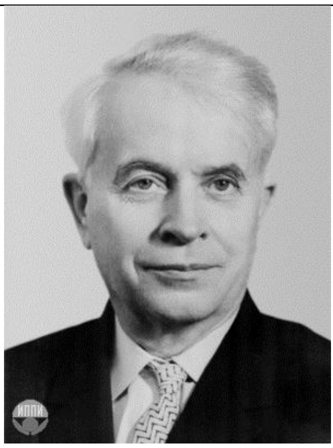
У 1936 р. С. Кліні поширив поняття загальнорекурсивної функції на випадок часткових функцій, увівши поняття ***частково рекурсивної функції*** та описавши клас таких функцій у чисто функціональних термінах.

У 1943 р. Е. Пост запропонував модель обчислюваних функцій на основі введеного ним числення спеціального вигляду – ***канонічних систем***.

Для формалізації самого поняття алгоритму були запропоновані точні математичні описи алгоритмічної машини й обчислюваності на ній. Першою формальною моделлю алгоритмічної машини була ***машина Тьюрінга*** (А. Тьюрінг, Е. Пост, 1936).

Із пізніших моделей відзначимо ***нормальні алгоритми*** (А. Марков, 1952) та ***регістрові машини*** (Д. Шепердсон, Г. Стерджіс, 1963).

Зауваження. Далі наведені короткі відомості про вчених, які внесли значний вклад в теорію алгоритмів.

		
Алонзо ЧЁРЧ (**Alonzo Church**) (1903–1995) – американский логик, математик. Внес значительный вклад в математическую логику, теорию алгоритмов, компьютерную математику. Его система лямбда-исчислений легла в основу функциональных языков программирования, в частности семейства Лисп (например, Scheme).	**Курт Фридрих Гёдель (Kurt Friedrich Gödel)** (1906-1978) – австрийский логик, математик и философ математики, доказал знаменитую теорему о неполноте (1931). С теоремой Геделя связано открытое в XX веке чрезвычайно важное явление алгоритмической неразрешимости. Заложил основы теоретической информатики.	**Стивен Коул Клини** (правильнее – **Кле́йни**, **Stephen Cole Kleene**) (1909–1994) – американский математик и логик. Его работы совместно с работами Алонзо Чёрча, Курта Гёделя и Алана Тьюринга дали начало разделу математической логики – теории вычислимости. Известен изобретением регулярных выражений, его именем названы алгебра Клини, звёздочка Клини, теорема Клини о рекурсии, теорема Клини о неподвижной точке. Работал также в области интуиционистсткой математики Брауэра. Внёс важный вклад в теорию конечных автоматов (см. теорема Клини).
		
Алан Тьюринг (***Alan Mathison Turing***) (1912–1954)– английский математик, логик, криптограф, оказавший существенное влияние на развитие информатики. Предложил в 1936 году абстрактную вычислительную «Машину Тьюринга», которую можно считать моделью компьютера общего назначения, позволила формализовать понятие алгоритма и до сих пор используется во множестве теоретических и практических исследований. Теория искусственного интеллекта.	**Андрей Андреевич Марков** (1903—1979) Американский математик и логик. Читал лекции по математике и логике в Колумбийском, Нью-Йоркском и других университетах США. Основные труды по математической логике: алгебра Поста, классы Поста функций алгебры логики; предложил абстрактную вычислительную машину — машину Поста.	**Эмиль Леон Пост** (*Post Emil Leon*) (1897—1954) Американский математик и логик. Читал лекции по математике и логике в Колумбийском, Нью-Йоркском и других университетах США. Основные труды по математической логике: алгебра Поста, классы Поста функций алгебры логики; предложил абстрактную вычислительную машину — машину Поста.

У 1936 р. А. Чорч і С. Кліні довели збіжність класів загальнорекурсивних та λ-значних функцій. На основі цього факту й аналізу ідей, які привели до вказаних понять, А. Чорч висунув тезу про збіжність класу алгоритмічно обчислюваних функцій із класом загальнорекурсивних функцій. С. Кліні узагальнив цю тезу для випадку часткових функцій.

Доведена А. Тьюрінгом у 1937 р. збіжність класів частково рекурсивних функцій та функцій, обчислюваних на машинах Тьюрінга, стала ще одним підтвердженням тези А. Чорча. Пізніше такі збіжності були встановлені для всіх відомих формальних моделей алгоритмічно обчислюваних функцій. Тому є всі підстави вважати, що кожна з названих вище формальних моделей адекватно уточнює інтуїтивне поняття алгоритмічно обчислюваних функцій.

Теорія алгоритмів виникла як розділ математичної логіки, поняття алгоритму тісно пов'язане з поняттям числення. Найширше застосування теорія алгоритмів має саме в математичній логіці. У наш час апарат теорії алгоритмів використовують всюди, де є алгоритмічні проблеми (основи математики, теорія інформації, теорія керування, конструктивний аналіз, обчислювальна математика, теорія ймовірності, лінгвістика, економіка та ін.).

Теорія автоматів – це розділ теорії керуючих систем, що вивчає математичні моделі перетворювачів дискретної інформації, які називають автоматами. Такими перетворювачами є як реальні пристрої (обчислювальні машини, живі організми), так і абстрактні системи (наприклад, формальна система – це сукупність абстрактних об'єктів, не пов'язаних із зовнішнім світом, в якому представлені правила оперування множиною символів у строгому синтаксичному трактуванні без урахування смислового змісту, тобто семантики; аксіоматичні теорії, що описують певну сукупність явищ у їхньому причинно-наслідковому зв'язку одне з одним).

Найбільш тісно теорія автоматів пов'язана з теорією алгоритмів. Це можна пояснити тим, що автомат перетворює дискретну інформацію по кроках у дискретні моменти часу і формує результуючу інформацію по кроках заданого алгоритму. Ці перетворення можливі за допомогою технічних та/або програмних засобів. Автомат можна зобразити як деякий пристрій (чорний ящик), на який подають вхідні сигнали і знімають вихідні і який може мати деякі внутрішні стани. Такий автомат називають абстрактним, оскільки абстрагуються від реальних фізичних вхідних і вихідних сигналів, розглядаючи їх просто як букви деякого алфавіту і в зв'язку з ідеалізованим дискретним часом. Особливе місце в теорії автоматів займає поняття скінченного автомата.

Скінченний автомат – математична абстракція, що дозволяє описувати зміни стану об'єкта залежно від його поточного стану та вхідних даних за умови що загальна можлива кількість станів і множина вхідних сигналів скінченні. Головною перевагою скінченних автоматів є те, що в них природним чином описано системи, керовані зовнішніми подіями.

Теорію автоматів застосовують як в математиці, так і у вирішенні практичних задач. Наприклад, засобами теорії автоматів можна довести розв’язність деяких формальних числень. Застосування методів і понять теорії автоматів до вивчення формальних і природних мов сприяло виникненню математичної лінгвістики (математична лінгвістика – математична дисципліна, предметом якої є розробка формального апарата для опису будови природних і деяких штучних мов.) Поняття автомата може служити модельним об'єктом у найрізноманітніших задачах, завдяки чому можливе застосування теорії автоматів у різних наукових і прикладних дослідженнях.

Поняття алгоритму. Основні властивості алгоритмів

Поняття алгоритму належить до первісних понять математики і є концептуальною основою процесів обробки інформації.

Обчислювальні процеси алгоритмічного характеру відомі людству з глибокої давнини. Такими є, наприклад, арифметичні дії над цілими числами, знаходження найбільшого спільного дільника двох чисел тощо. На початку XX ст. сформовано строге поняття алгоритму.

Під ***алгоритмом*** розуміють скінченну множину точно визначених правил для чисто механічного розв’язування задач певного класу. Така множина правил задає обчислювальний процес, названий ***алгоритмічним***, що починається з довільного початкового даного (вибраного з деякої фіксованої для даного алгоритму множини початкових даних) і спрямований на отримання повністю визначеного цим початковим даним результату.

Термін «алгоритм» походить від латинізованого імені *Algorithmi* великого Персидського вченого Мухаммеда аль-Хорезмі, який жив у першій половині IX століття (точні роки його життя невідомі, приблизно 780 – 850). «Аль-Хорезмі» означає «з Хорезма» (історичної області в нинішньому Узбекистані, центром якої було місто Хіва). Він вперше дав опис придуманої в Індії позиційної десяткової системи числення, вперше використав цифру 0 для позначення пропущеної позиції в записі числа, сформулював правила розв’язання квадратного рівняння.

Інтуїтивно ***алгоритм*** визначають як ***«порядок (послідовність) чітких недвозначних інструкцій, які зрозумілі виконавцеві та приводять до певного результату за скінчений час»***.

Алгоритм – це будь-яка система обчислень, виконуваних по строго визначеним правилам, яка після будь-якого числа кроків завідомо призводить до розв'язання поставленої задачі (А. Н. Колмогоров).

Алгоритм – це скінчений набір правил, який визначає послідовність операцій для розв'язання конкретної множини задач і має п'ять важливих рис: скінченність, визначеність, вхід, вихід, ефективність (Д. Е. Кнут).

Алгоритм – це точне розпорядження, що визначає обчислювальний процес, що йде від варійованих вихідних даних до шуканого результату (Марков А.А.).

Точне визначення алгоритму дати неможливо, тому що унаслідок первісності не можна виразити його іншими поняттями в математиці. Але можна сформулювати ряд інтуїтивних вимог до алгоритмів, виділивши характерні ***властивості алгоритму***, якими зазвичай є:

1) ***скінченність (фінітність)***: результат отримують за скінченну кількість кроків;

2) ***масовість***: початкові дані для алгоритму можна вибирати з певної (можливо, нескінченної) множини даних; це означає, що алгоритм призначений не для однієї конкретної задачі, а для класу однотипних задач;

3) ***дискретність***: алгоритм становить послідовність кроків, на кожному з яких виконують ту чи іншу інструкцію; кожну наступну інструкцію виконують після завершення попередньої;

4) ***елементарність***: кожна інструкція є елементарна для виконавця і не вимагає від нього ніякої винахідливості;

5) ***детермінованість***: після завершення чергового кроку завжди відомо, що робити на наступному кроці;

Ця властивість означає, що застосування алгоритму до тих же самих даних має привести до одного і того ж результату.

6) ***результативність***: має бути визначено, що саме слід вважати результатом роботи алгоритму. Виконання алгоритму повинно або закінчуватися результатом, або інформацією про те, чому не може бути отриманий результат. При цьому множина кроків, з яких складається алгоритм, є скінченою.

За допомогою алгоритму кожен конкретний результат можна отримати за скінченну кількість кроків зі скінченної множини даних. Отже, до таких початкових даних алгоритм ***застосовний***. Проте в деяких ситуаціях процес виконання алгоритму для певних початкових даних продовжується необмежено. Для таких початкових даних алгоритм ***незастосовний***.

Для опису алгоритму необхідно вказати множину його початкових даних та множину даних, до яких належать результати. Ці множини називають також ***множиною вхідних даних*** і ***множиною вихідних даних*** алгоритму. Алгоритм із множиною вхідних даних X та множиною вихідних даних Y називають X-Y-алгоритмом.

Нехай алгоритм $\mathcal{A}$ має множину вхідних даних X.

Областю застосування алгоритму $\mathcal{A}$ називають підмножину $D \subseteq X$ таку, що до кожного $d \in D$ алгоритм $\mathcal{A}$ застосовний. Якщо $\mathcal{A}$ видає результат b під час роботи над вхідним даним d, тоді позначатимемо $b = \mathcal{A}(d)$.

Кожний X-Y-алгоритм $\mathcal{A}$ визначає функцію $f : X \to Y$, узагалі кажучи, часткову, задану наступним чином.

Для кожного елемента $d \in X$ маємо $f(d) = \mathcal{A}(d)$, якщо $d \in D$, тобто $\mathcal{A}$ до d застосовний. Якщо $d \notin D$, то $f(d)$ невизначене, тобто $\mathcal{A}$ до d – незастосовний.

Отже, такий алгоритм $\mathcal{A}$ ***обчислює*** функцію f.

Функцію називають ***алгоритмічно обчислюваною*** (скорочено АОФ), якщо існує алгоритм, який її обчислює.

Множину L називають ***алгоритмічно перелічною***, якщо вона є областю значень для деякої алгоритмічно обчислюваної функції, тобто існує алгоритм, який перелічує елементи множини L і тільки їх.

Множину L називають ***алгоритмічно розв'язною*** множиною відносно множини U, якщо існує алгоритм, який дозволяє для кожного $x \in U$ визначати, $x \in L$ чи $x \notin L$.

Кожна алгоритмічно розв'язна непорожня множина є алгоритмічно перелічна.

Справді, нехай L розв'язна відносно U за допомогою алгоритму $\mathcal{A}$. Наведемо алгоритм $\mathcal{R}$, який перелічує множину L. Зафіксуємо довільний елемент $b \in L$. На вхід $\mathcal{R}$ подаємо довільний $x \in U$ і запускаємо $\mathcal{A}$ над x. Якщо $x \in L$, то $\mathcal{R}$ видає x як результат; якщо $x \notin L$, то $\mathcal{R}$ видає b як результат.

Із поняттям алгоритму тісно пов'язане поняття числення, яке є настільки ж фундаментальне, як і поняття алгоритму. Поняття числення віддзеркалює та узагальнює інтуїтивне уявлення про індуктивне породження об'єктів, яке широко вживане в математиці.

Під терміном «***числення»*** розуміють скінченну множину точно визначених ***правил виведення***, які дозволяють із певних заданих об'єктів отримувати інші об'єкти.

Об'єкти, до яких застосовують правила виведення, називають ***гіпотезами***. Отриманий з гіпотез об'єкт називають ***висновком***.

Множину породжених численням об'єктів задають індуктивно. На першому кроці процесу породження (виведення) початкові об'єкти можна задати правилами виведення із порожньою множиною гіпотез. Об'єкт можна вважати породженим на певному кроці, якщо він отриманий за допомогою правила виведення із об'єктів, породжених на попередніх кроках.

Якщо на першому кроці процесу породження дозволити брати початкові об'єкти з певної множини A, то дістанемо ***числення з входом***. Таке числення $\mathcal{J}$ перетворює множину A в множину об'єктів B, породжених із об'єктів множини A за допомогою числення $\mathcal{J}$.

Зв'язок понять алгоритму та числення полягає в такому:

1. Поняття числення можна звести до поняття алгоритму в значенні зведення розгалуженого процесу породження до послідовного процесу переліку так, щоб алгоритм, який задає цей перелік, відтворив усі породжені численням об'єкти і тільки їх.

Такий алгоритм послідовно додає до множини вже породжених об'єктів (спочатку порожньої) скінченну множину нових об'єктів, отриманих однократним застосуванням правил виведення до вже породжених об'єктів.

2. Поняття алгоритму можна звести до поняття числення в значенні зведення алгоритмічного процесу до процесу породження. Справді, кожний алгоритм можна трактувати як числення з входом, яке має такі правила виведення, що виконання кожного з них відповідає виконанню одного кроку алгоритму. Правила виведення мають одну гіпотезу, до кожного об'єкта застосоване не більше ніж одне правило виведення.