МЛТА Лекція 22.02.2021

Поняття числення можна уточнити за допомогою поняття формальної системи.

Під ***формальною системою*** розуміють трійку (L,A,P), де L – мова формальної системи, A– множина аксіом, P – множина правил виведення. Мову задають алфавітом та правилами побудови слів, які називають формулами. Кожна аксіома є формулою. Правила виведення формальної системи діють на множині формул. Формулу, яку отримують з аксіом за допомогою правил виведення, називають ***теоремою***.

Кожну формальну систему можна трактувати як числення з входом, у якому зафіксована множина початкових об'єктів A, а породжувальні правила – суть правила виведення із P.

Формальну систему можна також розглядати як числення із правилами виведення різного типу, які дозволяють відповідно породжувати формули, аксіоми і теореми та виділяти теореми як результативні об'єкти.

До формальних систем, крім класичних аксіоматичних теорій, належать канонічні й нормальні системи Поста, формальні граматики тощо.

Вивчаючи формальні моделі алгоритмів, зазвичай розглядають числові алгоритми в розширеній множині натуральних чисел. Чому? Ми знаємо безліч алгоритмів, в яких фігурують об'єкти довільної природи. Виявляється, що будь-який логічний алгоритм можна досить простими методами звести до числового. Тому теорію числових алгоритмів можна вважати універсальним апаратом для дослідження всіх алгоритмічних проблем.

Теорема. Будь-яку алгоритмічну проблему можна звести до обчислення значень деякої цілочисельної функції цілочисельних аргументів.

Для доведення позначимо все початкові умови задачі $\mathcal{F}$, які переробляються алгоритмом $\mathfrak{A}$, у вигляді послідовності з цілими невід'ємними індексами

$$a_0\ , a_1, ..., a_n, ...$$

Розв'язки теж можна записати в вигляді занумерованої послідовності

$$b_0\ , b_1, ..., b_m, ...$$

Після введення нумерації будемо використовувати не самі записи умов і розв'язків, а їхні номери. Тепер можна сформулювати алгоритм, який перетворює номер запису умови в номер запису розв'язку. Цей алгоритм виконує обчислення значень числової функції

$$m=\varphi(n),$$

тобто є числовим алгоритмом.

Якщо існує алгоритм розв'язання вихідної задачі $\mathcal{F}$, то є і алгоритм, який визначає значення відповідної функції φ.

Формальні моделі алгоритмів та алгоритмічно обчислювальних функцій

Машини натуральнозначних регістрів. МНР-обчислюваність

Машина натуральнозначних регістрів (МНР) – це ідеалізована модель комп’ютера і була запропонована у 70-ті роки XX сторіччя з метою моделювання реальних обчислювальних машин та аналізу складності обчислень.

МНР містить нескінченну кількість регістрів, вмістом яких є натуральні числа $\{0,1,2,3,\dots\}$. Регістри нумеруємо (йменуємо) натуральними числами, починаючи з 0, і позначаємо їх R_0 , R_1, ..., R_n, ... Вміст регістру R_n позначаємо $'R_n$.

R_0	R_1	R_2	$\cdots$	R_n	$\cdots$

Послідовність $('R_0,'R_1,\dots,'R_n,\dots)$ вмісту регістрів МНР назвемо ***конфігурацією*** машини натуральнозначних регістрів**.**

Відзначимо, що тільки скінченна множина регістрів містить числа, відмінні від нуля. Всі інші регістри заповнені нулями. Це допущення передбачає, що кожен алгоритм використовує тільки скінченний обсяг пам'яті.

МНР може змінити вміст регістрів згідно з виконуваними нею ***командами*** в порядку їх написання. Скінченний список команд утворює ***програму*** МНР. Команди програми послідовно нумеруються натуральними числами, починаючи з 1. Номер команди в програмі називатимемо також ***адресою*** команди. Довжину (кількість команд) МНР-програми P позначатимемо $|P|$.

Команди МНР бувають 4-ох типів:

1. Обнулення n-го регістру:

$$Z(n):\ 'R_n = 0.$$

2. Збільшення вмісту n-го регістру на 1:

$$S(n):\ 'R_n = 'R_n + 1.$$

3. Копіювання вмісту регістру R_m у регістр R_n або переадресація:

$$T(m,n):\ 'R_n = 'R_m \text{ (у такому разі } 'R_m \text{ не змінюється).}$$

4. Умовний перехід: $J(m,n,q)$:

якщо $'R_n = 'R_m$, то перейти до виконання q-ї команди, інакше – виконувати наступну за списком команду програми. Число q у команді $J(m,n,q)$ назвемо адресою переходу.

Команди типів 1–3 називають арифметичними. Після виконання арифметичної команди МНР має виконувати наступну за списком команду програми.

Виконання однієї команди МНР назвемо ***кроком*** МНР.

Роботу програми МНР починає, перебуваючи в деякій ***початковій конфігурації***, з виконання першої за списком команди. Наступну команду програми для виконання визначено так, як описано вище. Виконання програми завершується (програма зупиняється), якщо наступна для виконання команда відсутня (тобто номер наступної команди перевищує номер останньої команди програми).

Конфігурацію МНР у момент завершення виконання програми називають ***фінальною*** (***заключною***), вона визначає результат роботи МНР-програми над даною початковою конфігурацією.

Якщо МНР-програма P при роботі над початковою конфігурацією $K_0=\left(a_0,a_1,\ldots\right)$ ніколи не зупиняється, цей факт позначатимемо $P\left(a_0,a_1,\ldots\right)\uparrow$, якщо ж зупиниться після скінченної кількості кроків, цей факт будемо позначати $P\left(a_0,a_1,\ldots\right)\downarrow$.

Якщо МНР-програма P у роботі над початковою конфігурацією $\left(a_0,a_1,\ldots\right)$ зупиняється із фінальною конфігурацією $\left(b_0,b_1,\ldots\right)$, цей факт позначатимемо так: $P\left(a_0,a_1,\ldots\right)\downarrow\left(b_0,b_1,\ldots\right)$.

Конфігурацію виду $\left(a_0,a_1,\ldots,a_n,0,0,\ldots\right)$, у якій $'R_m=0$ для всіх $m>n$, будемо називати ***скінченною***. Таку конфігурацію будемо позначати $\left(a_0,a_1,\ldots,a_n\right)$. Якщо МНР-програма P починає роботу над скінченною початковою конфігурацією, то в процесі виконання програми P МНР перебуватиме тільки в скінченних конфігураціях.

МНР-програми P та Q назвемо ***еквівалентними***, якщо вони визначають однакові відображення послідовностей натуральних чисел. Це означає, що у процесі роботи над однаковими початковими конфігураціями вони або обидві зупиняються з однаковими фінальними конфігураціями, або обидвом не зупиняються.

МНР-програма P ***обчислює*** часткову n-арну функцію $f:N^{n+1}\to N$, якщо:

– за умови $\left(a_0,a_1,\ldots,a_n\right)\in D_f$ та $f\left(a_0,a_1,\ldots,a_n\right)=b$ маємо $P\left(a_0,a_1,\ldots,a_n\right)\downarrow b$;

– за умови $\left(a_0,a_1,\ldots,a_n\right)\notin D_f$ маємо $P\left(a_0,a_1,\ldots,a_n\right)\uparrow$.

Це означає, що значення аргументів функції послідовно розміщуються у регістрах, починаючи з R_0, а значення функції знімається з регістру R_0.

Неважко переконатись, що наведене визначення МНР-обчислюваності еквівалентне такому:

МНР-програма P обчислює часткову n-арну функцію $f:N^n\to N$, якщо

$$f\left(a_1,a_2,\ldots,a_n\right)=b\Leftrightarrow P\left(a_1,a_2,\ldots,a_n\right)\downarrow b.$$

Функцію називають ***МНР-обчислюваною***, якщо існує МНР-програма, яка обчислює цю функцію.

Розглянемо приклади МНР-програм для деяких функцій та предикатів.

Приклад 1. МНР-програма для функції $f(x,y)=x-y$. Зауважимо, що ця функція є частковою на множині визначення $\mathbb{N}_0=\{0,1,2,3,\ldots\}$, тобто

$$f(x,y)=\begin{cases} x-y, x\geq y \\ невизначена, x<y \end{cases}.$$

1) $J(0,1,5)$
2) $S(1)$
3) $S(2)$
4) $J(0,0,1)$
5) $T(2,0)$

Наприклад: $x=5$, $y=1$, $f(5,1)=4$.

R_0	R_1	R_2
5	1	0
5	2	1
5	3	2
5	4	3
5	5	4
4	5	4

Наприклад: $x=2$, $y=6$, $f(2,6)-невизначена$.

R_0	R_1	R_2
2	6	0
2	7	1
2	8	2
2	9	3
2	…	…
2	…	…

МНР зациклюється.

Приклад 2. МНР-програма для функції $f(x,y)=x\dot{-}y$. Ця функція називається ***усічена різниця***., тобто

$$f(x,y)=x\dot{-}y=\begin{cases}x-y, & x\geq y\\ 0, & x<y\end{cases}.$$

Зауважимо, що ця функція є всюди визначеною на множині $\mathbb{N}_0=\{0,1,2,3,\ldots\}$.

1) $J(0,1,7)$
2) $J(0,2,6)$
3) $S(1)$
4) $S(2)$
5) $J(0,0,1)$
6) $Z(2)$
7) $T(2,0)$

Наприклад: $x=4$, $y=2$, $f(4,2)=2$.

R_0	R_1	R_2
4	2	0
4	3	1
4	4	2
2	4	2

Наприклад: $x=2$, $y=6$, $f(2,6)=0$.

R_0	R_1	R_2
2	6	0
2	7	1
2	8	2
2	8	0
0	8	8

Зауважимо, що кожна МНР-програма обчислює множину функцій натуральних аргументів та значень, але, зафіксовуючи наперед арність функцій, маємо, що кожна МНР-програма обчислює єдину функцію заданої арності.

МНР-програму P назвемо ***стандартною***, якщо в P для кожної команди виду $J(m,n,q)$ виконано умову $q \le |P| + 1$.

Приклад 3. МНР-програма для функції $f(x,y,z) = \max\{x,y,z\}$.

1) $J(0,3,5)$
2) $J(1,3,6)$
3) $S(3)$
4) $J(0,0,1)$
5) $T(1,0)$
6) $Z(3)$
7) $J(0,3,11)$
8) $J(2,3,12)$
9) $S(3)$
10) $J(0,0,7)$
11) $T(2,0)$

Наприклад:

$x = 4$, $y = 2$, $z = 5$, $f(4,2,5) = 5$.

R_0	R_1	R_2	R_3	R_4
4	2	5	0	0
4	2	5	1	0
4	2	5	2	0
4	2	5	0	0
4	2	5	1	0
4	2	5	2	0
4	2	5	3	0
4	2	5	4	0
5	2	5	4	0

$x = 4$, $y = 1$, $z = 2$, $f(4,1,2) = 4$.

R_0	R_1	R_2	R_3	R_4
4	1	2	0	0
4	1	2	1	0
4	1	2	0	0
4	1	2	0	0
4	1	2	1	0
4	1	2	2	0