Іспит з теорії алгоритмів
студента групи ПМі-21
Суса Назарія

Білет № 25

① Головні оператори. Оператор примітивної рекурсії. Оператор мінімізації

При побудові частково-рекурсивних функцій використовують три головні оператори: оператор суперпозиції, оператор примітивної рекурсії; оператор мінімізації.

Оператор примітивної рекурсії.

Нехай задано часткові числові функції: $g^n: N^{(n)} \to N$ та $h^{n+2}: N^{(n+2)} \to N$, де $N = Z^+$ - розширена множина натуральних чисел, $N^{(n)} = \{ \langle x_1, \ldots, x_n \rangle \mid x_i \in N \}$ - множина усіх можливих n натуральних чисел.

Розглянемо часткову функцію $f^{n+1}: N^{(n+1)} \to N$ таку, що

$$(*) \begin{cases} f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n) \\ f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)) \end{cases}$$ для довільних натуральних чисел $x_1, \ldots, x_n, y$.

Оператор, який за допомогою системи рівнянь (*) з функцій g і h дає змогу побудувати функцію f, називають оператором примітивної рекурсії і позначають R: $f = R(g, h)$

Функції f, g, h - числові.

У випадку одномісної функції $f^1: N \to N$ маємо

$$\begin{cases} f(0) = a, \ a \in N \\ f(y+1) = h(y, f(y)) \end{cases}$$

Оператор мінімізації.

Нехай задана часткова функція $f^n: N^{(n)} \to N$. Припустимо, що існує механічна процедура для обчислення функції f^n, причому f^n не визначена лише в тому випадку, коли ця процедура працює нескінченно довго і не видає результату.

Зафіксуємо перші $n-1$ аргументів цієї функції і розглянемо рівняння $f(x_1, .., x_{n-1}, y) = x_n$.

Найменше значення $y=a$, для якого виконується рівність $f(x_1, .., x_{n-1}, a) = x_n$ позначимо $\mu_y(f(x_1, .., x_{n-1}, y) = x_n)$.

Процес знаходження такого числа буде безмежним, коли:

1. $f(x_1, .., x_{n-1}, 0)$ невизначене
2. значення $f(x_1, .., x_{n-1}, y)$ для $y = \overline{0,k}$ визначені, але не дорівнюють x_n, а для $y = k+1$ функція f невизначена
3. значення $f(x_1, .., x_{n-1}, y)$ визначені для $y = 0, 1, 2 ...$, але відмінні від x_n, тобто рівняння $f(x_1, .., x_{n-1}, a) = x_n$ не має натуральних розв'язків.

Оператор, за допомогою якого з функції $f(x_1, .., x_n)$ утворюється функція $\mu_y(f(x_1, .., x_{n-1}, y) = x_n)$, називають <u>оператором мінімізації</u> (оператором найменшого кореня). Його позначають Mf: $Mf = \mu_y(f(x_1, .., x_{n-1}, y) = x_n)$.

У загальному випадку функція Mf є частковою

② <u>Основні властивості алгоритмів. Способи запису алгоритмів</u>

<u>Різновиди алгоритмів. Композиція алгоритмів.</u>

Система правил П повинна мати властивості, які притаманні кожному алгоритму: дискретність, ефективність, скінченність, результативність, масовість:

<u>Дискретність алгоритму</u>. Алгоритм описує процес послідовної побудови величини, який відбувається за дискретний час.

Спочатку в момент t_0 задають скінченну систему σ_0 вхідних даних. Далі в кожний інтервал часу (t_i, t_{i+1}) із системи σ_i певних даних, які були в момент часу t_i, за певним правилом отримують систему σ_{i+1}. Такий перехід називають елементарним кроком алгоритму. Ці інтервали є різними для різних елементарних кроків.

Ефективність алгоритму. Елементарні кроки алгоритму повинні бути ефективними, тобто виконуватися точно і за короткий відрізок часу. Це ще називають зрозумілістю алгоритму. Різні виконавці, згідно з алгоритмом, повинні діяти однаково і отримати однаковий результат для однакового початкового набору даних.

Скінченність алгоритму. Алгоритм завжди повинен закінчуватися після скінченної кількості кроків.

Результативність. Алгоритм завжди забезпечує отримання певного результату. Якщо деякий елементарний крок не дає результату, то має бути зазначено, що саме треба вважати результатом роботи цього алгоритму.

Масовість. Алгоритм повинен бути застосовним до цілого класу задач, а не до одиниці. Це означає, що систему величин σ_0 можна вибрати з деякої потенційно нескінченної множини.

Якщо система Π з пари $<\varphi, \Pi>$ задовільняє всі перераховані властивості, крім скінченності, то таку пару називають обчислювальним методом.

Способи запису алгоритмів:

- словесний запис (псевдокод)
- графічний запис (блок-схеми)
- запис мовою програмування (алгоритмічною мовою — мовою, що використовують для формального запису алгоритмів)

Різновиди алгоритмів.

Наявність чи відсутність у системі Π тієї чи іншої властивості розбиває множину всіх алгоритмів на два класи.

1. Нехай система правил Π є такою, що система величин σ_{i+1} однозначно визначена системою σ_i. Алгоритм $A = <\varphi, \Pi>$ називають детермінованим (визначеним), якщо Π задовільняє

що властивість. В іншому випадку алгоритм недетермінований, при цьому оператор φ – багатозначний.

2. Алгоритм $A = <\varphi, \Pi>$ називають самозупинним, якщо в процесі виконання алгоритму вхідні слова системи Π [illegible] і несамозупинним в іншому випадку.

3. Нехай система правил Π алгоритму $A = <\varphi, \Pi>$ закодована певним способом у вхідному алфавіті алгоритму. Позначимо її Π^{cod}.

Алгоритм $A = <\varphi, \Pi>$ називають самозастосовним, якщо слово Π^{cod} входить в область визначення A, і несамозастосовним в іншому випадку.

4. Алгоритм називають універсальним, якщо він еквівалентний довільному наперед заданому алгоритму $A = <\varphi, \Pi>$.

Композиція алгоритмів.

Чотири способи композиції алгоритмів:

1. Суперпозиція алгоритмів. $\xrightarrow{P} \boxed{A} \xrightarrow{A(P)} \boxed{B} \xrightarrow{Q}$

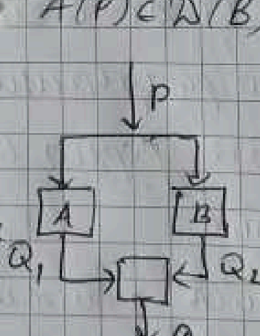
У випадку суперпозиції двох алгоритмів A та B вихідне слово одного розглядають як вхідне іншого.

$D(A)$ – область визначення A

$D(B)$ – область визначення B

$C(P) = B(A(P))$ $P \in D(A)$, $A(P) \in D(B)$

2. Об'єднання алгоритмів



Нехай алгоритми A і B визначені в одному алфавіті. Об'єднанням алгоритмів A та B є такий алгоритм C, який перетворює довільне вхідне слово $P \in D(A) \cap D(B)$ в конкатенацію слів: $C(P) = A(P)B(P)$

3. Розгалуження алгоритмів

Нехай задано 3 алгоритми A, B, C і їх області визн. D(A), D(B), D(C). Розгалуженням називають таку композицію алгоритмів A, B, C, задану співвідношенням:

$$F(P) = \begin{cases} A(P), & \text{якщо } C(P) = R \\ B(P), & \text{якщо } C(P) \neq R \end{cases}$$, де R – деяке слово

$D(F) = D(A) \cap D(B) \cap D(C)$

4. Ітерація алгоритмів

Ітерацією (повторенням) алгоритмів A та B називають алгоритм C, який визначають так: для довільного P вихідне слово C(P) отримують як результат послідовного багаторазового застосування A доти, доки не буде отримано слово, перетворюване алгоритмом B в деяке фіксоване слово R.

③ Задано $f(x,y) = x \cdot y$. Знайти $d(x,y) = M f(x,y)$, де M – оператор мінімізації

$f(x,z) = y = x \cdot z$, де z – найменше значення таке, що для деяких x і y, маємо $y = x \cdot z$.

Отже, $d(x,y) = \mu_z(x \cdot z = y) = \frac{y}{x}$

Приклад: $d(2,6)$

$z = 0$: $2 \cdot 0 = 0 \neq 6$ $\qquad d(2,6) = 3$

$z = 1$: $2 \cdot 1 = 2 \neq 6$

$z = 2$: $2 \cdot 2 = 4 \neq 6$

$z = 3$: $2 \cdot 3 = 6 = 6$

Застосування оператора мінізації над операцією множення реалізує операцію ділення.