Екзаменаційна робота
з теорії алгоритмів
ПМі-23
Гнень Крістіни

Білет № 4

(1) Різновиди алгоритмів. Композиція алгоритмів

Наявність чи відсутність у системі правил тієї чи іншої властивості розбиває множину всіх алгоритмів на класи щодо конкретної властивості.

Розглянемо 4 властивості:

1) Нехай система П є такою, що система величин $б_{i+1}$ однозначно визначена системою $б_i$. Алгоритм $A = \langle p, П \rangle$ назив. детермінованим (визначеним), якщо П задовольняє цю властивість, і недетермінованим в іншому випадку

(В детермінованому ал. алфавітний оператор є однозначний, у недетермінованому - багатозначний

2) Алгоритм $A = \langle p, П \rangle$ назив. самозмінним, якщо в процесі переробки алгоритмом вхідних слів система П змінюється, і незамозмінним - в іншому випадку.

Самозмінний алгоритм можна розглядати як систему двох алгоритмів, перший з яких виконує переробку вхідних слів, а другий вносить зміни в робочий алгоритм.

* Частковим випадком самозмінних алгоритмів є алгоритми, які самонавчаються.

3) Нехай система правил алгоритму $A = \langle \varphi, \Pi \rangle$ закодована певним способом у вхідному алфавіті алгоритму A. Позначимо цей код P^{cod}. Алгоритм $A = \langle \varphi, \Pi \rangle$ назив. самозастосовним, якщо слово P^{cod} входить в область визначення алгоритму і несамозастосовним в протилежному випадку.

4) алгоритми називають універсальними, якщо вони еквівалентні довільному наперед заданому алгоритму $A = \langle \varphi, \Pi \rangle$ (можна реалізовувати будь-які алгоритм. процеси)

Композиція алгоритмів

Нові алгоритми можуть бути побудовані з уже відомих шляхом застосування різних способів композиції алгоритмів

Є 4 способи:

1) Суперпозиція алгоритмів

При суперпозиції двох алгоритмів A і B вихідне слово одного з них розглядають як вхідне слово іншого.

*Може виконуватись для довільної скінченної кількості алгоритмів

$A(P) = xP \quad B(P) = Py \implies C(P) = xPy$

2) об'єднання алгоритмів

Нехай A і B 2 алгоритми в одному й тому ж алфавіті V.; D(A) D(B) – області визначення цих алгоритмів

Об'єднанням алгоритмів A і B є такий алгоритм C в цьому ж алфавіті, який перетворює довільне вхідне слово $P \in D(A) \cap D(B)$ в конкатенацію слів A(P) і B(P):

$C(P) = A(P)\,B(P)$ /на всіх інших словах алг. C вважають невизначеним).

$A(P) = xP \quad B(P) = Py \quad C(P) = xPPy$

3) розгалуження алгоритмів – це композиція 3 алгоритмів A, B, C задана 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 фіксоване слово, наперед задане. Область визначення отриманого алг. буде перетин областей визнач. всіх трьох алгоритмів

4) Ітерація (повторення) двох алгоритмів A і B назив. ал. C, який виконується так: для довільного вх. ~~с~~ слова P вих. слово C(P) отримують як результат багаторазового застосування алгоритму A доти, доки не буде отримане слово, перетворюване алгоритмом B в деяке фіксоване слово R

② Формальне визначення машини Тьюрінга.

Теза Тьюрінга

З математичного погляду — це певний алгоритм для переробки вхідних слів.

Оскільки спосіб переробки машиною слів визначений, як-що визначена програма машини, то МТ вважають заданою, якщо задані: → зовнішній алфавіт

- → внутрішній алфавіт
- → програма
- → символи, що відповідають порожній комірці й заключному стану.

Можна МТ реалізує таку функцію $П: S \times \{Q \setminus q_F\} \to S \times Q \times Z$

або $П = \{ <s_i, q_l> \to <s_j, q_r, Z> \}, q_l \neq q_F$.

Тоді формально машиною Тьюрінга називають 7-у

$(S, Q, q_0, q_F, I, \Lambda, П)$, де $I \subseteq S$ множина вхідних символів, Λ - порожній алфавіт

Якщо визначити строго формально МТ як математичний об'єкт, то можна на формальному рівні визначити

поняття → конфігурації
- → кроку обчислення
- → і всі інші поняття

Вважають, що слово P задане в стандартному положенні, якщо головка фіксує перший зліва символ вхідного слова. Якщо ж початкову конфігурацію K_0 взяти стандартне положення слова P, то процес МТ(K_0), в процесі переробки вхідного слова P, і його позначають МТ(P).

МТ обчислює деяку словникову функцію $\varphi(P)$, яка відображає слово з деякої множини слів у значення слова, якщо для довільного слова P процес МТ(P) доходить заключної стандартної конфігурації, асоційованої зі словом $\varphi(P)$ - результатом цієї функції.

Теза Тьюрінга: Для довільного алгоритму $A<\varphi, П>$ у довільному скінченному алфавіті X існує функція $\varphi(P): \{P\}_+ \to \{Q\}_x$, обчислювана за Тьюрінгом.

=> Будь-який алгоритм, заданий у довільній формі, можна замінити еквівалентною йому МТ.

=> Питання про можливість алгоритмізації того чи іншого процесу зводиться питанню про можливість реалізації цього процесу на МТ.

3. Довести, що функція $f(x_1, x_2) = x_1 : x_2$ є примітивно рекурсивною.

$$f(x,y) = \frac{x}{y}$$

$$f(0,y) = 0 \cdot y = 0 = O^1(y) = g(x)$$

$$f(x+1,y) = \frac{x+1}{y} = f(x,y) + \frac{1}{y} = h(x,y,f(x,y))$$

$$h(x,y,f(x,y)) = I^3_3(x,y,z) + \frac{1}{I^3_2(x,y,z)}$$

$$f = R(g,h) = R\left(O^1(y), S\left(+;\ I^3_3(x,y,z), \frac{1}{I^3_2(x,y,z)}\right)\right)$$