Теорія алгоритмів.

Екзамен.

Банко Марківна. ПМО-21

Білет № 4

1) Різновиди алгоритмів:

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

• Алгоритм $A = \langle \varphi, \Pi \rangle$ назив. детермінованим (визначеним), якщо система Π є такою, що система величин σ_{i+1} однозначно визначена системою σ_i і алгоритм задовільняє цю властивість.

В іншому випадку він є недетермінованим.

• Алгоритм $A = \langle \varphi, \Pi \rangle$ назив. самозмінним, якщо в процесі переробки алгоритмом вхідних слів система Π змінюється і несамозмінним в іншому випадку

Частковим випадком самозмінних алгоритмів є алгоритми, які самонавчаються.

• Нехай система правил алгоритму $A = <Ч, П>$ закодована певним способом у вхідному алфавіті алгоритму A. Позначимо цей код P^{cod}. Алгоритм $A=<Ч,П>$ наз. самозастосовним, якщо слово P^{cod} входить в область визначення алгоритму, і несамозастосовним в протилежному випадку.

• Алгоритм наз. універсальним, якщо він еквівалентний довільному наперед заданому алгоритму $A = <Ч, П>$.

Композиції алгоритмів.

Є чотири способи композиції алгоритмів: суперпозиція, об'єднання, розгалуження та ітерація.

• При суперпозиції двох алгоритмів A та B вихідне слово одного з них розглядають як вхідне слово іншого

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

- <u>Об'єднання алгоритмів</u>

$P \in D(A) \cap D(B)$ $\qquad C(P) = A(P)B(P)$

На всіх інших словах алгоритм C вважають невизначеним.

- <u>Розгалуженням</u> алгоритмів наз. композицію трьох 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 фіксоване слово, переважно порожнє. Областю визначення функціонального алгоритму буде переріз областей визначення всіх трьох алгоритмів: $D(F) = D(A) \cap D(B) \cap D(C)$

- <u>Ітерацією</u> (повторенням) двох алгоритмів A і B наз. алгоритм C, який визначають так: для довільного вхідного слова P вихідне слово C(P) отримують як результати послідовного багаторазового застосування алгоритму A доти, доки не буде отримано слово, перетворюване алгоритмом B в деяке фіксоване слово R.

3) Функцію наз. примітивно-рекурсивною, якщо вона утворена з найпростіших функцій за допомогою скінченного числа застосувань операторів суперпозиції S^n_m та примітивної рекурсії R^n.

$f(x_1, x_2) = x_1 \cdot x_2$

$y(x_1) = f(x, 0) = x \cdot 0 = 0 = O^1(x_1)$

$h(x_1, x_2, f(x_1, x_2)) = (x_1, x_2 + 1) = (x_1, (x_2+1)) =$

$= S^3(+, f(x_1, x_2), x_1) = S^3(R(I_1^1(x_1) S^1(x_1, x_2, y))),$

$I_3^3(x_1, x_2, y), I_1^3(x_1, x_2, y))).$

Ми довели, що функція $f(x_1, x_2) = x_1 + x_2$

є примітивно-рекурсивною.

2) Формальне визначення МТ. Мова Тьюрінга.

З математичного погляду МТ – це певний алгоритм для переробки вхідних слів. Оскільки спосіб переробки машинних слів відомий, якщо відома машинна програма, то МТ вважають заданою, якщо задані її зовн. і внутр. алфавіти, програма та визначено, які з символів відповідають початковій конфігурації й заключному стану. Кожна МТ реалізує таку функцію

$\Pi : S \times \{Q \setminus q_F\} \to S \times Q \times Z$ або інакше

$\Pi = \{ \langle s_i, q_i \rangle \to \langle s_j, q_r, Z \rangle \}, q_r \neq q_F$.

Тоді формально машиною Тьюрінга наз. сімку $(S, Q, q_0, q_F, I, \Lambda, \Pi)$.

Якщо для часткової словникової функції $\varphi(P)$ існує МТ, яка її обчислює, то функцію $\varphi(P)$ наз. обчислюваною за Тьюрінгом.

· <u>Теза Тьюрінга</u>: Для довільного алгоритму $A = \langle \varphi, \Pi \rangle$ у довільному скінченному алфавіті X існує функція $\varphi(P) : \{P\}_X \to \{Q\}_X$, обчислювана за Тьюрінгом.

Теза Тьюрінга визначає відповідність між інтуїтивним ~~поняттям~~ поняттям алгоритму і точним мат. поняттям функції, обчислюваної на МТ.