Міхневич Владислав

Білет № 19

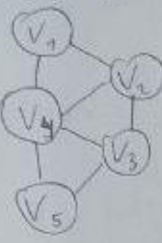
1. Класи P і NP є визначеними як класи мов. Це спрощує систему позначень. Задачі з різних галузей часто можна сформулювати як задачі розпізнавання мов. Наприклад задача яка для кожного значення вхідних даних потребує відповіді „так" чи „ні". Можна закодувати значення у вигляді ланцюжків і переформулювати її як задачу про розпізнавання мов, що складається з усіх ланцюжків вхідних даних, які приводять до відповіді „так". Для того щоб зв'язок задача – мова був зрозумілішим приймемо такі узгодження

1) цілі числа будемо зображати в десятковій системі
2) вузли графа будемо зображати цілими числами $1, \ldots, n$ закодованими у десятковій системі. Ребро будемо зображати ланцюжком (i_1, i_2) де i_1, i_2 – вузли
3) булеві вирази з n пропозиційними змінними будемо зображати ланцюжками у яких „*" позначає „і", „+" позначає „або", „¬" позначає „не", а цілі числа $1, 2, \ldots n$ – зображають пропозиційні змінні

Можна сказати, що задача належить P або NP, якщо її стандартне зображення належить до P або NP відповідно

<u>Приклад 1</u>: Задача про кліку для неорієнтованих графів. У графі G k-клікою наз. k-вузловий повний граф. Саму задачу формулюють так: чи містить заданий граф G k-кліку, де k – задане ціле число

Мову L, яка описує задачу про кліку, утворюють такі ланцюжки вигляду $k(i_1, j_1)(i_2, j_2)\ldots(i_m, j_m)$, що граф з ребрами (i_r, j_r), $1 \le r \le m$, містить k-кліку.
Задача про кліку належить до класу NP. У цьому разі для перевірки достатньо $O(n^3)$ кроків, де n - довжина коду задачі.



При $k = 3$ цей граф можна закодувати ланцюжком $3(1,2)(1,4)(2,4)(2,3)(3,4)(3,5)(4,5)$
Цей граф містить три клiки: $\{v_1, v_2, v_4\}$, $\{v_2, v_3, v_4\}$, $\{v_3, v_4, v_5\}$
Задача є NP повна.

<u>Приклад 2:</u> Задача про виконанну булеву формулу. Її формулюють так: чи виконанна задана булева формула?
Задача виконання булевої формули є NP-повною.
Алгоритм, який розпізнає чи формула є виконанною, починає з того, що вгадує виконаний набір 0-1 істинних і хибних значень, для яких формула повертає істину.
Отже, деяка недетермінована машина Тюрінга розпізнає виконання булевої ф-ли з послідовною часовою складністю, і тому ця задача належить до класу NP.

<u>Приклад 3:</u> Задача про комівояжера. В цій задачі задано множину міст і вартість подорожі між ними. Необхідно визначити порядок, у якому треба відвідати всі міста і повернутись назад у початкове місто. Задача буде розв'язана, якщо знайдено гамільтонів цикл з найменшою довжиною. Ця задача належить до класу NP-повних задач. В задачі з N містами необхідно розглянути $(N-1)!$ маршрутів.

У випадку великих чисел неможливо за розумний час отримати результат.

② Розмір задачі – обсяг вхідних даних, час затрачений алгоритмами, як ф-ію розміру задачі наз. часовою складністю. Поведінка цієї складності при збільшенні розміру задачі називають асимптотичною часовою складністю. Аналогічно визначають ємнісну складність, як затрачуваний обсяг пам'яті. Нехай A – алгоритм для розв'язування деякого класу задач, а n – розмір окремої задачі. $f_A(n)$ – робоча функція, яка дає оцінку алгоритму для задачі розміру. Алгоритм A наз. поліноміальним, якщо $f_A(n)$ зростає не швидше, ніж поліном, а в іншому разі – експоненціальним. Функція $f(n)$ наз. порядку O від ф-ії $g(n) \neq 0$, якщо виконується:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = const \neq 0$$

Функція $f(n)$ наз. порядку o від ф-ії $g(n) \neq 0$, якщо:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Алгоритм наз. поліноміальним якщо $f_A(n) = O(P_n(n))$, тобто алгоритм наз. поліноміальним, якщо $f_A(n)$ зростає не швидше ніж поліном від n, в іншому випадку експоненціальним. Звідси до експоненціальних належать і алгоритми, складність яких можна оцінити, наприклад функцією $n^{\log n}$, яка не є експонентною.

Алгоритм – це формально описана обчислювальна процедура що отримує вхідні дані, які наз. також входом алгоритму, або його аргументом і видає результат обчислень на вихід.

У загальній теорії алгоритмів виділяють дві сторони:

– дескриптивну, яка вивчає питання наявності чи відсутності алгоритмів і числень, які приводять до заданої мети і способи задання цих алгоритмів та числень.

– метричну, яка займається оцінюванням складності процесів обчислення.

3. Довести примітивну рекурсивність функції

$$f(x,y,z,v) = x\cdot y + z\cdot v$$

1) "+" $f_1(x,y) = x+y$ – примітивно рекурсивна

$f_1(x,0) = x+0 = I_1^1(x) = g(x)$

$f_1(x,y+1) = h(x,y,f(x,y)) = x+y+1 = S^1(f(x,y))$

2) "·" $\begin{cases} f_2(x,0) = g(v) = x\cdot 0 = 0 = O^1(x) \\ f_2(x,y+1) = h(x,y,f(x,y)) = x(y+1) = xy+x = S^3(f_1^?, f(x,y), I^1(x,y)) \end{cases}$

$f(x,y,z,v)$:

3) $\begin{cases} f(x,y,z,v) = x\cdot y + z\cdot 0 = x\cdot y = f_2(x,y) = g(x,y,z) \\ f(x,y,z,v+1) = xy+zv+z = S^3(f_1^?, f(x,y,z,v), I_3^4) = \end{cases}$

$= h(x,y,z,v, f(x,y,z,v))$