теория сложности

literature:

- Arora Barak "Complexity Modern Approach" (1st part)
- Garry Johnson "Трудно разрешенные задачи"
- site: compendium of NP-complete problems

outline:

NP-полнота

Характеристики сложности вычисления.
Есть распознователи ($\Sigma^* \to B$) и преобразователи ($\Sigma^* \to \Sigma^*$)

- время: $T(n) = O(f(n))$
- память: $S(n)$
- random: $R(n)$

$DTIME(f) = \{L \mid \exists\, program\ p:$
$1.\ x \in L \implies p(X) = 1, x \notin L \implies p(x) = 0$
$2.\ n = |x| \implies T(p, x) = O(f(n))\}$

$h = (01)^* \in DTIME(n)$

$\widetilde{DTIME}(f) = \{h \mid \ldots\}$

палинромы: $Pal \in DTIME_{RAM}(n)$
$Pal \notin DTIME_{TM}(n)$

$P = \cup_{f-polynom} DTIME(f) = \cup_{i=0}^{\infty} DTIME(n^i)$

$p(n)q(n) : p + q, p * q, p(q(n))$
$L_1 L_2 \in P : L_1 \cup L_2 \in P, L_1 \cap L_2 \in P, \overline{L_1} \in P, L_1 L_2 \in P, L_1^* \in P$

концепция недетрминированных вычислений

Допускается $\iff$ $\exists$ последовательность переходов, которая приводит к допуску
недетерминированння программа p(x) допускает $\iff$ $\exists$ последовательность недетерминированных выборов, приводящая к допуску
p(x) не допускает $\iff$ $\forall$ последовательности выборов не допуск

def *NTIME(f)* = $\{L \mid \exists$ недетерминированная программа p
$1)\ p(x) - acc \iff x \in L;\ 2)\ T(p,x) = O(f(n))\}$

> **ex** задача о гамильтоновом цикле
>
> ```
> p(G)
>  vis[1..n]: arr of bool
>  s = 1
>  for i = 1..n
>      u = ?{1..n}
>      if (vis[u]) return false
>      if (su not in EG) return false
>      vis[u] = true
>      s = u
>  if (s ≠ 1) return false
>  return true
> ```

> **ex** `isComposite(z)`, $n = \lceil \log_B z \rceil$, где B - это основание системы счисления
>
> ```
> a = ?{2..z-1} // T = logn
> if z % a = 0  // poly(logn)
>  return true
> return false
> ```

Нельзя свопнуть бранчи и сделать проверку на простоту, потому что это `true` и `false` не симметричны в недетерминированных вычислениях (нельзя даже `isPrime(n): return !isComposite(n)`)

---

def *NP* = $\cup_{f-polynome}\ NTIME(f)$, nondeterministic polynomial
stat $P \subset NP$

? $P = NP$

неформально: класс P - класс задач, которые можно решить за полином, класс NP - класс задач, решение которых можно проверить за полином

Σ_1 - класс языков, в которых можно формализовать класс решения, которое можно проверить за полином
$\Sigma_1 = \{L \mid \exists$ полином p, работающая за полином программа R(x, y) - детерминированная
$x \in L \iff \exists\, y$ (называют *сертификат*): $|y| \le p(|x|)\ and\ R(x,y) = 1$
$x \notin L \implies \forall\, y\ (|y| \le p(|x|))\ R(x,y) = 0\}$

> **ex** гамильтонов цикл $Ham \in \Sigma_1$
>
> ```
> R(G, y):
>  y as arr[1..n] of int
>  // we can add: y = ?arr[i..n] of {1..n} // O(n)
>  vis = arr[1..n] of bool
>  for i = 1..n
>      if (y[i] y[i mod n+1] not in EG) return false
>      if vis[y[i]] return false
>      vis[y[i]] = true
>  return true
> ```

Th $NP = \Sigma_1$
$L \in NP, L \in \Sigma_1$
неформально: NP - определение на языке недетерминированных формат, Σ_1 - определение на языке сертификатов

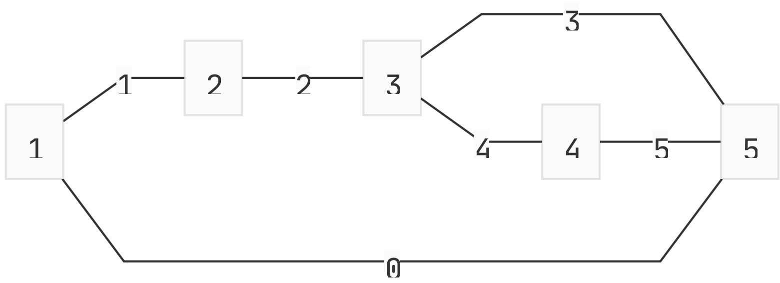
сведения

def сводим B к A *по Тьюрингу*: A, B - языки, C - сложностный класс, $B \in C^A$ (C с *оракулом* A). не считая вызова функции `isInA(x): Bool`, остальные ограничения класса C учитываются.

def сведение по *Куку-Левину (Тьюрингу за полином)* $B \in P^A$

def *сведене по Карпу (m-сведение)*: язык B сводится к A ($B \leq A$), если $\exists$ вычислимая за полином функция f такая, что $x \in B \iff f(x) \in A$

> **ex** $IND = \{< G, k > | $ в $G\ d$ независимое множество размера k $\}$
> $CLIQUE = \{< G, k > |$ в $G \exists$ клика размера k$\}$
> $IND \leq CLIQUE$
> $f(< G, k >) =< \overline{G}, k >$ // за полином
> в G и множестве размера k $\iff$ в $\overline{G}$ $\exists$ клика размера k
> $VCOVER = \{< G, k > |$ в G $\exists$ вершинное покрытие размера k $\}$
> $IND \leq VCOVER$
> $f(< G, k >) =< G, n - k >$, где n - число вершин G

> **ex**
> $SUBSETSUM = \{< [x_1, x_2, \dots, x_n], s > \ | \exists I \subset \{1, 2, \dots, n\}, \sum_{i \in I} = s, x_i \in \mathbb{N}\}$
> `dp[i][w]` - можно ли первые i $\Sigma = w$ // w - $2^{|s|}$
> $VCOVER \leq SUBSETSUM$
> пронумеруем вершины с единицы, рёбра – с нуля, битовыми масками каждой вершине сопоставляем рёбра
>
> 
>
>
> | | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
> |---|---|---|---|---|---|---|---|
> | x_1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
> | x_2 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
> | x_3 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
> | x_4 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

	6	5	4	3	2	1	0
x_5	1	1	0	1	0	0	1
s	3	2	2	2	2	2	2

$x_6 = 1$
$x_7 = 10$
$x_8 = 100$
$x_9 = 1000$
$x_{10} = 10000$
$x_{11} = 100000$
$f(< G, k >)$, n - число вершин, m - число рёбер, $s = k22...2$, m двоек

f сводит VCOVER к SUBSETSUM

$\Rightarrow$: в G $\exists$ вершинное погрытие размера k

$\Leftarrow$: $[x_1 \dots, x_{n+n}], s$ $\exists$ решение $\Rightarrow$ в G $\exists$ вершинное покрытие размера k

def язык называется *NP-hard* (*NP-трудный*), если выполнены следующие условия:

$\forall B \in NP : B \leq A$

def A называется *NP-complete* (*NP-полный*), если:

1) $A \in NPH$

2) $A \in NP$

// $NPC = NPH \cap NP$

ex BH_{1N} (bounded halting unary nondeterministic)

$BH_{1N} = \{< m, x, 1^t > \ |$ m – недетрминировання машина тьюринга, x – вход, t – ограничение времени: $\exists$ последоватеьность недетерминировання выборов машины Тьюринга m, что она допускается за t шагов: `m(x) = 1`$\}$

Th $BH_{1N} \leq NP$