теория сложности

> **literature:** Arora Barak "Complexity Modern Approach" (1st part)
>
> **outline:**
>
> - NP-полнота
>   - Концепция недетрминированных вычислений
> - Сведения
>   - Теорема Кука-Левина

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: $R(n)$

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

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

$DT\widetilde{IME}(f) = \{h \mid \dots\}$

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

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

$p(n)q(n) : p + q, p * q, p(q(n))$
$L_1L_2 \in P : L_1 \cup L_2 \in P, L_1 \cap L_2 \in P, \overline{L_1} \in P, L_1L_2 \in P, L_1^* \in P$

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

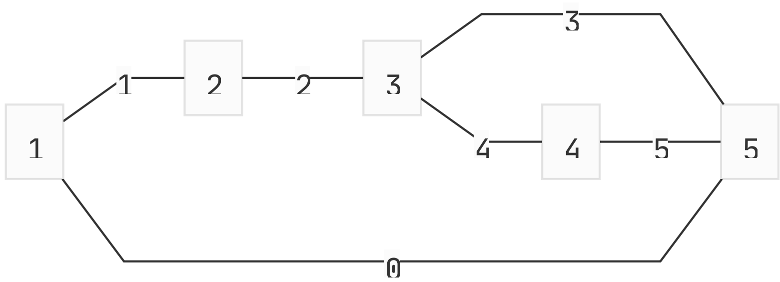
def сводим B к A *по Тьюрингу*: A, B - языки, C - сложностный класс, $B \in C^A$ (C *с оракулом* A). не считая вызова функции `isInA(x): Bool`, остальные ограничения класса C учитываются.

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

def *сведене по Карпу (m-сведение)*: язык B сводится к A ($B \le A$), 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$\}$
> $IND \le CLIQUE$
> $f(< G, k >) =< \overline{G}, k >$ // 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 $\}$
> $IND \le VCOVER$
> $f(< G, k >) =< G, n - k >$, 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 - $2^{|s|}$
> $VCOVER \le SUBSETSUM$
> 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

$\Leftarrow$: $[x_1 ..., x_{n+n}], s$ $\exists$ 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 – недетрминированя машина тьюринга, x – вход, t – ограничение времени: $\exists$ последовательность недетерминированя выборов машины Тьюринга m, что она допускается за t шагов: `m(x) = 1`}

Th $BH_{1N} \leq NPC$

1. $BH_{1N} \in NPH$
   $A \in NP$
   // def по Карпу
   m_A - недетерминированя машина Тьюринга, решающая A за полином
   $p(n) = cn^k$
   $f(x) =< m_A, x, q^{p(|x|)} >$
   $x \in A \iff \exists$ последовательность выборов $m_A(x) = 1$ (за $p(|x|)$)
2. $BH_{1N} \in NP$

L $A \leq^k B, B \leq^k C \implies A \leq^k C$
$x \xrightarrow{t} f(x) \xrightarrow{t} g(f(x))$
con $A \in NPH, A \leq B \implies B \in NPH$

stat если $B \leq A$, $A \in NPH$
$NP \xrightarrow{t} BH_{1N} \xrightarrow{t} SAT$

def $SAT = \{\phi(x_q ... x_n) \mid \exists x_1 ... x_n\ \phi(x_1 ... x_n) = 1, \phi -$ б ф }

Th (Кук, Левин)

$SAT \in NPC$

$BH_{1N} \leq SAT$
$< m, x, 1^t > \overset{f}{\mapsto} \phi$

ϕ удовлетворяет $\iff$ $\exists$ последовательность недетерминированных выборов $m(x) = 1$, за время t

больше t шагов не будет, есть мгновенные описания машины $\alpha\#_q\beta$
дополним описания до длины t + 1
$q_0 \vdash q_1 \vdash \ldots \vdash q_t$

табло вычислений: первая строка - стартовое состояние, $i \to i+1, q_i \vdash q_{i+1}$, допуск: последовательность до $\#_{acc}$

$< m, x, 1^t > \ \in BH_{1N} \iff \exists$ допускающее табло вычислений

количество состояний $|Q| = z$, множество ленточного алфавита $|PT| = y$, $z + y = k$
заведём $(t+1)^2 k$ переменных, x_{ijc} – верно ли, что в табло в i-й j-й ячейке записан символ 'c'

$\phi(x_{ijc}) = C \wedge S \wedge T \wedge N$

$C = \wedge i, j = 0..t \vee_C ((\wedge \neg X_{ij\alpha}) \wedge X_{ijc})$

$S = X_{00\#_s} \wedge X_{01x_1} \wedge X_{02x_2} \wedge \ldots \wedge X_{0nx_n} \wedge X_{0(n+1)B} \wedge \ldots$

$T = X_{t0\#x} \vee X_{t1\#_y} \vee \ldots \vee X_{tt\#_y}$

$N = (\wedge_{i,j} \wedge_{c_1 c_2 c_3 c_3 \notin Q} X_{i-1,j-1,c_1} \wedge X_{i-1,j,c_2} \wedge X_{i,j+1,c_3} \wedge X_{i,j,c_4} \to c_1 = c_4) \wedge_{ijx} \wedge_{c_1 \ldots c_6 \ldots}$ допустимы

$qed \, \square$