# теория сложности

> **literature:**
>
> - Arora Barak "Complexity Modern Approach" (1st part)
> - Garry Johnson "Трудно разрешенные задачи"
> - site: compendium of NP-complete problems
>
> **outline:**
>
> - NP-полнота
>     - Концепция недетрминированных вычислений
> - Сведения
>     - Теорема Кука-Левина
> - язык CNFSAT
>     - Теорема $CNFSAT \in NPC$
>     - Теорема $CNFSAT \to 3SAT$
> - Теорема $IND \in NPC$
> - диагональный метод
>     - теоремы об иерархии
>         - Теорема о ёмкости иерархии
>         - Теорема о временной иерархии
>     - Теорема Бэйкера-Гилла-Соловэя (BGS)

## 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: $R(n)$

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

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

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

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

$\Sigma_1$ - 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, $\Sigma_1$ - определение на языке сертификатов

---

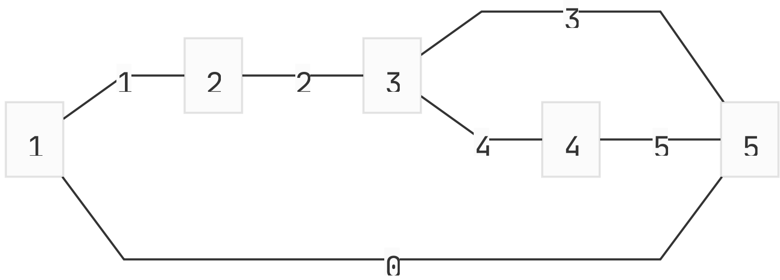
## сведения

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

> **ex** $SUBSETSUM = \{\langle [x_1, x_2, \ldots, x_n], s\rangle \mid \exists I \subset \{1, 2, \ldots, n\}, \sum_{i \in I} = s, x_i \in \mathbb{N}\}$
> `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 |
> | $x_5$ | 1 | 1 | 0 | 1 | 0 | 0 | 1 |

| | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| s | 3 | 2 | 2 | 2 | 2 | 2 | 2 |

$x_6 = 1$
$x_7 = 10$
$x_8 = 100$
$x_9 = 1000$
$x_{10} = 10000$
$x_{11} = 100000$
$f(\langle G, k\rangle)$, 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)
$BH_{1N} = \{\angle m, x, 1^t\rangle$ | m – недетрминированння машина тьюринга, x – вход, t – ограничение времени: $\exists$ последовательность недетерминированння 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
   $m_A$ - недетерминированння машина Тьюринга, решающая A за полином
   $p(n) = cn^k$
   $f(x) = \langle m_A, x, q^{p(|x|)}\rangle$
   $x \in A \iff \exists$ последовательность выборов $m_A(x) = 1$ (за $p(|x|)$)
2. $BH_{1N} \in NP$

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

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

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

### Th *(Кук, Левин)* SAT in NPC

$SAT \in NPC$

$BH_{1N} \leqslant SAT$
$\langle m, x, 1^t\rangle \overset{f}{\mapsto} \phi$

$\phi$ 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 $\#_{acc}$

$\langle m, x, 1^t \rangle \in BH_{1N} \iff \exists$ 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'

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

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

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

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

$N = (\wedge_{i,j} \wedge_{c_1c_2c_3c_3 \notin Q} X_{i-1,j-1,c_1} \wedge X_{i-1,j,c_2} \wedge X_{i,j+1,c_3} \wedge X_{i,j,c_4} \to c_1 = c_4) \wedge_{ijx} \wedge_{c_1 \ldots c_6 \ldots}$
допустимы

*qed* $\square$

---

## язык CNFSAT

**def** $CNFSAT = \{\phi \mid \phi$ в КНФ$, \phi \in SAT\}$
$(x_i \vee \neg x_j \ldots) \wedge (\vee \vee \vee) \wedge (\vee)$
*clause* (клоз)
**ex** 2-SAT (ровно две) HornSAT (не более одной без отрицания)

### Th CNFSAT in NPC

1. $CNFSAT \in NP$
2. $CNFSAT \in NPH$
   $SAT \leqslant CNFSAT$
   $\phi \xrightarrow{f \;\; (polynomial\ time)} \psi$
   $\phi \in SAT \iff \psi = f(\psi) \in CNFSAT$

базис: $\wedge, \vee, \neg$

строим дерево разбора нашей формулы $\phi$:

- если у neg сын neg, то можем удалить
- neg -> and/or => neg <- and/or -> neg neg

каждому поддереву соответствует преобразованная подформула $\phi_i(x_{i_1} \ldots x_{i_k})$ ,
хотим построить следующее: $\psi_i(x_{i_1} \ldots x_{i_k}, y_1 \ldots y_{i_t})$
$\phi(\overline{X}) = 1 \implies \exists \overline{y} \psi(\overline{x}, \overline{y}) = 1$
$\phi(\overline{X}) = 0 \implies \forall \overline{y} \psi(\overline{x}, \overline{y}) = 0$

| вершина | brand new $\psi$ |
|---|---|
| X | $\phi = X, \psi = X$ |
| neg X | $\phi = \neg X, \psi = \neg X$ |
| and | $\phi_1 \wedge \phi_2, \psi_1 \wedge \psi_2$ |

| вершина | brand new $\psi$ |
|---|---|
| or | $\psi_1 \vee \psi_2$ не можем написать, потому что это не будет в КНФ<br>новая переменная z:<br>$(\psi_1 \vee z) \wedge (\psi_2 \vee \neg z)$ |

> получается, что число клозов равно числу листьев
> внутри каждого клоза число вхождений равно число переменных + или

#clauses = #leaves
#entries = #vars + #or
poly

$\square$ *qed*

### Th CNFSAT to 3SAT

> $3SAT = CNFSAT \wedge 3CNF$

1. $3SAT \in NP$
2. $3SAT \in NPH$
   $CNFSAT \leqslant 3SAT$

| $\psi$ | $X$ |
|---|---|
| $(x \vee y \vee u) \wedge (x \vee y \vee \neg u)$ | $x \vee y$ |
| ok | $x \vee y \vee z$ |
| вспомогательные переменные<br>k - 3 новые перменные:<br>$(x_1 \vee x_2 \vee t_1) \wedge (\neg t_1 \vee x_3 \vee t_2) \wedge (\neg t_2 \vee x_2 \vee t_3) \wedge \ldots \wedge (\neg t_{k-3} \vee x_{k-1} \vee x_k)$ | $x_1 \vee x_2 \vee \ldots \vee x_{k}$ |

$\square$ *qed*

3SAT - superstar

## Th IND in NPC

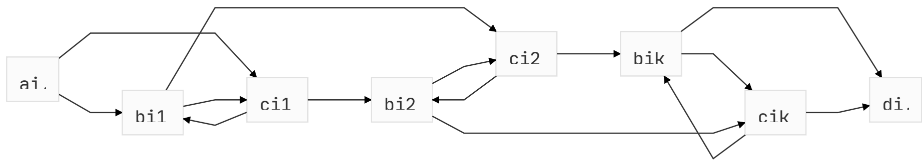
дана формула $\phi$ в 3КНФ, мы хотим вывести граф G и число k, такие что $\phi$ удовлетворима тогда и только тогда, когда в графе есть независимое множество размера k
$\phi \in 3SAT \iff \langle G, k \rangle \in IND$

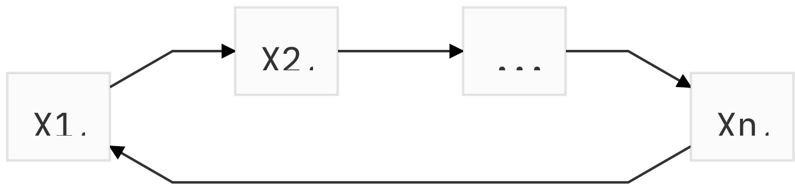
в $\phi$ k clauses, граф построим из k triangles
в вершинах переменные, соответствующие claus'ам
соединим переменные с их отрицанием

---

$HAM = \{G \mid G -$ ориентированный граф, содержит Гамильтонов цикл$\}$
$HAM \in NP$
$HAM \in NPH$

$\phi(x_1 x_2 \ldots x_n)$ k clauses
$x_i \to 2k + 2$ вершины




где X - это компонента предыдущего вида

---

## диагональный метод

### теоремы об иерахии

$DSPACE(f) = \{L \mid \exists$ программа p: $x \in L \implies p(x) = 1\ S(p, x) = O(f(n))\}$
$x \notin L \implies p(x) = 0$

$PSACE = \cup_{p-polynom} DSPACE(p)$

**Th NP subset PS subset EXP**

**thesis** если p запускает q, q использует $O(f)$ памяти, то p может тоже для этого использовать$O(f)$ памяти

#### Th о ёмкости иерархии

$\frac{f}{g} \to 0$ тогда $\exists L : L \in DSPACE(g) \backslash DSPACE(f)$

$h = \sqrt{fg},\ \ \frac{h}{g} \to 0,\ \ \frac{f}{h} \to 0$

$n = |\langle p, x\rangle|$

$L = \{\langle p, x\rangle \mid$ неверно, что $(p(\langle p, x\rangle) = 1,$ использовав $h(n)$ памяти $)\}$

$L \in DSPACE(g)$

Пусть $L \notin DSPACE(f)$, q - разрешает L, используя $\leqslant cf(n)$, рассмотрим $n_0 : h(n_0) > cf(n_0)$, $n_0 > |q|$

рассмотрим $x : |\langle q, x\rangle| = n_0$

$q(\langle q, x\rangle) =\ ?$

$q(\langle q, x\rangle) = q \implies \langle q, x\rangle \in L \implies !(q(\langle q, x\rangle) = 1\ and\ S(q, \langle q, x\rangle) \leqslant cf(n)\langle h(n_0)) \implies q(\langle q, x\rangle) = 0$

$q(\langle q, x\rangle) = 0 \implies \langle q, x\rangle \notin L \implies q(\langle q, x\rangle) = 1$

#### Th о временной иерархии

DSPACE -> DTIME, память -> время

ломается немного первая часть, так что новое условие:

$\frac{f}{g} \to 0, \exists h : \frac{f}{h} \to 0, \frac{sim(h)}{g} \to 0.$ $(sim(h) = O(g))$ (где $sim(f)$ - за сколько можно просимулировать программу, работающую за f) тогда
$\exists L : L \in DTIME(g) \backslash DTIME(f)$

$h = \sqrt{fg}, \ \ \frac{h}{g} \to 0, \ \ \frac{f}{h} \to 0$

$n = |\angle p, x\rangle|$

$L = \{\angle p, x\rangle \mid$ неверно, что $(p(\angle p, x\rangle) = 1,$ использовав $h(n)$ времени $)\}$

$L \in DTIME(g)$

Пусть $L \notin DTIME(f)$, q - разрешает L, используя $\leqslant cf(n)$, рассмотрим
$n_0 : h(n_0) > cf(n_0)$, $n_0 > |q|$

рассмотрим $x : |\langle q, x\rangle| = n_0$

> **Implies** $P \neq EXP$
>
> $f = n^{\log_2 n} = 2^{(\log_2 n)^2}$
> $g = 2^n$
> $\frac{f}{g} \to 0 \implies \exists L \in DTIME(g) \backslash DTIME(f)$ (первая часть $\implies L \in EXP$, вторая – $\implies L \notin P$)

### Th (Бейкер, Гилл, Соловэй) BGS

$u = \{\langle p, x\rangle \mid \ p(x) = 1\}$
$uni(p, x) \to$ останавливается ли p на x

Вычисления с оракулом $p^A$ - p  с оракулом A

$\exists$ оракул $A : p^A = NP^A$
$\exists$ оракул $B : p^B \neq NP^B$

// **релятивизуется**, если доказательство остаётся верным, если всему фиксированному в программе добавить оракул

рассмотрим $A \in PSC$

$p^A \overset{1}{\subset} NP^A \overset{2}{\subset} PS^A \overset{3}{\subset} PS \overset{4}{\subset} P^A$:

1. любая недетерминированння программа частный случай детерминированной
2. релятивизуется
3. можем заменить вызов оракула на процедуру проверки
4. потому что взяли PSpace полный, любой сводится за полином и спросим у оракула

B $\ \ U_B = \{x \mid \exists y \in B \ \ \ |x| = |y|\}$

**L** $\forall B \ \ U_b \in NP^B$

Придумаем $B : U_B \notin P^B$