# Основы кибернетики

**Доказательство теоремы Кука – Левина. Примеры $NP$-полных проблем, связанных с графами**

Материал излагается в соответствии с брошюрой: Сапоженко А. А. Некоторые вопросы сложности алгоритмов : Учебное пособие. — М. : Издательский отдел факультета ВМиК МГУ, 2001. — 46 с.

## Полиномиальная сводимость. *NP*-полнота. Теорема Кука – Левина

**Машины Тьюринга.** Термин “машина Тьюринга” (сокращенно МТ) употребляется здесь для одноленточных *детерминированных*. машин, изучавшихся на первом курсе. Некоторое (непринципиальное) отличие состоит в том, что мы рассматриваем МТ с односторонней разделенной на ячейки лентой, бесконечной вправо. Алфавит ленты МТ обозначим через $A$, а множество состояний — через $Q$. Символом $q_1$ обозначается начальное состояние, символом $a_1$ — пустой символ, присутствующий по определению в алфавите $A$. Считается, что в начальный момент слово $w = b_1 b_2 \ldots b_n$, обрабатываемое МТ, записано в первых $n$ ячейках ленты, а все остальные ячейки ленты содержат символ $a_1$.

*Детерминированность* МТ означает, что для каждой пары вида $(a, q)$, где $a$ — символ входного алфавита, а $q$ — символ состояния, в программе МТ присутствует не более одной команды вида: $aq \to a'q'd$, начинающейся с $aq$. Содержательный смысл команды $aq \to a'q'd$ таков: если считывающее устройство в обозреваемой им ячейке обнаружило символ $a$ и при этом машина Тьюринга находится в соотоянии $q$, то следует заменить символ в обозреваемой ячейке на $a'$, машине Тьюринга перейти в состояние $q'$, а считывающему устройству совершить перемещение в соответствии с символом $d \in \{L, S, R\}$ ($L$ — сдвинуться на одну ячейку влево, $S$ — остаться на месте, $R$ — сдвинуться на одну ячейку вправо). Программа МТ состоит из конечного числа команд.

Пусть в процессе работы МТ на некотором такте $t$ оказалось, что на ленте записано слово $w = b_1 b_2 ... b_m$, МТ находится в состоянии $q_j$, головка МТ обозревает ячейку с номером $k$. *Конфигурацией (мгновенным описанием)*, соответствующей этому такту $t$, называется слово вида $C_t = b_1 b_2 ... b_{k-1} q_j b_k ... b_m$. Конфигурация, соответствующая первому такту, называется *начальной*, а последнему (если МТ останавливается) — *заключительной*. *Вычислением МТ $M$ на входе $w$* называется последовательность конфигураций $C_1, C_2, ..., C_t, ...$, возникающая при работе МТ $M$ на слове $w$. *Время работы $t_M(w)$ МТ $M$ на входе $w$* определяется как число конфигураций в вычислении МТ $M$ на входе $w$. Если вычисление бесконечно, полагаем $t_M(w) = \infty$. Пусть среди состояний МТ имеются выделенные заключительные состояния — принимающее и отвергающее. Тогда вычисление называется *принимающим* (*отвергающим*), если оно заканчивается в принимающем (отвергающем) состоянии.

**Недетерминированные Машины Тьюринга.** Отличие недетерминированной МТ (сокращенно, НМТ). состоит в том, что в программе НМТ для пары $(a, q)$ в ее программе может присутствовать несколько (но не более некоторого фиксированнного для заданной МТ числа) команд, начинающихся с $aq$. Без ограничения общности мы можем сейчас обойтись случаем, когда паре $aq$ соответствует не более двух команд с началом $aq$. Пусть в программе НМТ имеется пара команд $aq \to a'q'L$ и $aq \to a''q''R$. Тогда, находясь в состоянии $q$ и обозревая символ $a$ на ленте, НМТ может выбрать любую из двух возможностей: записать в обозреваемую ячейку символ $a'$, перейти в состояние $q'$ и сдвинуть головку влево, либо записать в обозреваемую ячейку символ $a''$, перейти в состояние $q''$ и сдвинуть головку вправо. При этом считается, что НМТ как бы создает две копии самой себя и прослеживает последовательность вычислений обоих способов действия. Понятие конфигурации для НМТ не отличается от того, что определено выше.

*Вычислением НМТ на входе* $w$ называется последовательность конфигураций $C_1, C_2, ..., C_t, ...$, с $C_1 = q_1 w$ и такая, что $C_{i+1}$ получается из $C_i$ с помощью одной из команд, соответствующих паре $a(i)q(i)$, где $q(i)$ — символ состояния, входящий в $C_i$, а $a(i)$ — буква из $C_i$, стоящая справа от $q(i)$. Всякое вычисление можно изобразить ориентированной цепью, вершинами которой являются конфигурации, а каждая дуга соединяют две последовательные вершины. В случае детерминированных МТ вычисление однозначно определяется входом. В случае НМТ объединение цепей, соответствующих вычислениям на входе $w$, представляет собой ориентирорванное (от корня) дерево с корнем $C_1 = q_1 w$.

**Распознавание языков.** Пусть $A$ — конечный алфавит. Через $A^\omega$ обозначим множество всех *слов* (конечных последовательностей) в алфавите $A$. Через $||w||$ обозначим *длину* слова $w$, определяемую, как число букв в $w$. Произвольное подмножество $L \subseteq A^\omega$ называется *языком* в алфавите $A$. Говорят, что МТ (НМТ) $M$ с двумя заключительными состояниями (принимающим и отвергающим) *распознает* язык $L$, если для всякого слова $w \in A^\omega$ принимающее вычисление $M$ на входе $w$ существует тогда и только тогда, когда $w \in L$. В случае, когда $w \notin L$, все вычисления либо бесконечны, либо являются отвергающими. Говорят, что МТ (НМТ) $M$ *распознает язык* $L$ *за полиномиальное время*, если она распознает $L$ и существует полином $p$ такой, что для каждого слова $w \in L$ существует принимающее вычисление длины, не превышающей $p(||w||)$.

Через $\mathbf{P}$ обозначим класс языков, распознаваемых детермирнированными МТ за полиномиальное время. Через $\mathbf{\Pi}$ обозначим множество отображений вида $f : A^\omega \to A^\omega$, вычисляемых детерминированными МТ за полиномиальное время. Пусть $L$ и $K$ — языки. Говорят, что $L$ *(полиномиально) сводится к* $K$ (обозначение $L \prec K$), если существует функция $f \in \mathbf{\Pi}$ такая, что $f(w) \in K \Leftrightarrow w \in L$.

Языки $L$ и $K$ *(полиномиально) эквивалентны*, если $K \prec L$ и $L \prec K$. Класс языков, распознаваемых НМТ за полиномиальное время, обозначается через **NP**. Язык $L$ называется *(полиномиально) полным* или *NP-полным*, если
1) $L \in \mathbf{NP}$,
2) $K \in \mathbf{NP} \Rightarrow K \prec L$.

Справедливы следующие простые утверждения.

**Утверждение 1.** *Если* $L \prec K$ *и* $K \prec H$, *то* $L \prec H$.

**Утверждение 2.** *Если* $K \in \mathbf{P}$ *и* $L \prec K$, *то* $L \in \mathbf{P}$.

**Утверждение 3.** $\mathbf{P} \subseteq \mathbf{NP}$.

**Утверждение 4.** *Либо все NP-полные языки принадлежат* **P**, *либо ни один из них не принадлежит* **P**. *Первое имеет место тогда и только тогда, когда* $\mathbf{P} = \mathbf{NP}$.

## Теорема Кука – Левина

Язык ВЫПОЛНИМОСТЬ (сокращенно ВЫП) состоит из слов, представляющих собой выполнимые КНФ, т. е. КНФ, не равные тождественно 0, в некотором конечном алфавите $A$, содержащем символы $($, $)$, $\&$, $\vee$, $\neg$, а также символы для записи названий переменных.

**Теорема 1 (С. А. Кук – Л. А. Левин, независимо, 1971).** *Задача ВЫП является NP-полной.*

*Доказательство* состоит из двух частей. Первая часть: доказательство того, что если $L \in \mathbf{NP}$, то $L \prec$ ВЫП. Вторая часть: доказательство того, что задача ВЫП принадлежит классу $\mathbf{NP}$.

*Доказательство первой части.* Пусть $L \subseteq \Sigma^\omega$. Поскольку $L \in \mathbf{NP}$, то существует НМТ, распознающая язык $L$ за полиномиальное время. Пусть полином $p(x)$ и НМТ $M$ таковы, что $M$ распознает $L$ и $t_M(w) \le p(||w||)$ для любого слова $w \in L$. Мы укажем способ построения по произвольному слову $w$ КНФ $A(w) = A(w, M, p)$, выполнимой тогда и только тогда, когда $w \in L$. Тем самым будет указано отображение $f : A^\omega \to A^\omega$, удовлетворяющее условию $f(w) \in L \Leftrightarrow A(w) \in$ ВЫП. Принадлежность построенного отображения $f$ классу $\mathbf{\Pi}$ легко проверяется. Занумеруем ячейки односторонней ленты НМТ $M$ слева направо натуральными числами. Пусть $\Sigma = \{a_1, a_2, ..., a_l\}$ — алфавит ленты НМТ $M$, $\{q_1, q_2, ..., q_r\}$ — множество состояний НМТ, $w \in \Sigma^\omega$ — произвольное слово длины $n$. Положим $T = \lfloor p(n) \rfloor$. Заметим, что если МТ заканчивает работу не более чем за $p(n)$ тактов, то ячейки ленты с номерами большими, чем $T$, не посещаются считывающим устройством машины Тьюринга.

Введем логические (булевы) переменнные, от которых будет зависеть строящаяся КНФ $A(w)$.
$P^i_{s,t}$, где $1 \le i \le l$; $1 \le s, t \le T$. Переменная $P^i_{s,t}$ истинна тогда и только тогда, когда ячейка с номером $s$ на шаге $t$ содержит символ $a_i$
$Q^i_t$, где $1 \le i \le r$; $1 \le t \le T$. Переменная $Q^i_t$ истинна тогда и только тогда, когда на шаге $t$ НМТ находится в состоянии $q_i$.
$S_{s,t}$, где $1 \le s, t \le T$. Переменная $S_{s,t}$ истинна тогда и только тогда, когда на шаге $t$ ячейка с номером $s$ обозревается головкой.
КНФ $A(w)$ является конъюнкцией $B \,\&\, C \,\&\, D \,\&\, E \,\&\, F \,\&\, G$, образованной следующим образом.

Формула $B$ утверждает, что на каждом шаге $t$ обозревается одна и только одна ячейка. Формула $B$ является конъюнкцией $B_1 \,\&\, B_2 \,\&\, \ldots \,\&\, B_T$, где $B_t$ утверждает, что на шаге $t$ обозревается одна и только одна ячейка

$$B_t = (S_{1,t} \vee S_{2,t} \vee \ldots \vee S_{T,t}) \;\&\; \mathop{\&}_{1 \le i < j \le T} (\bar{S}_{i,t} \vee \bar{S}_{j,t}).$$

Для $1 \le s, t \le T$ формула $C_{s,t}$ утверждает, что на шаге $t$ в ячейке $s$ находится один и только один символ, а $C$ является конъюнкцией всех таких $C_{s,t}$.
Формула $D$ утверждает, что для каждого $t$ НМТ находится ровно в одном состоянии. Формулы $C$ и $D$ строятся аналогично $B$.

Формула $E$ утверждает, что выполнены начальные условия.

$$E = Q_1^1 \& S_{1,1} \& P_{1,1}^{i_1} \& P_{2,1}^{i_2} \& ... \& P_{n,1}^{i_n} \& P_{n+1,1}^{1} \& ... \& P_{T,1}^{1},$$

где $w = a_{i_1} a_{i_2} ... a_{i_n}$, $q_1$ — начальное состояние и $a_1$ — пустой символ.

Формула $F$ утверждает, что для каждого $t$ преобразование слова на ленте, сдвиг считывающего устройства и изменение состояния осуществляются в соответствии с программой НМТ. Если же ячейка не обозревается, то содержимое ее не изменяется. $F$ представляет собой конъюнкцию формул $F_{s,t}$ по всем $s, t$.

Формула $F_{s,t}$ утверждает:
1) если на шаге $t$ ячейка с номером $s$ не обозревается на шаге $t$, то символ, находящийся в ней, не изменяется;
2) если же $s$-я ячейка обозревается на шаге $t$, то изменения состояния и символа в обозреваемой ячейке, а также сдвиг головки производятся в соответствии с программой НМТ по символу, находящемуся в $s$-й ячейке, и состоянию НМТ.

Пусть $R_{s,t,i,j}$ означает следующее: при условии, что на шаге $t$ обозревается ячейка $s$, в обозреваемой ячейке записан символ $a_i$, а НМТ находится в состоянии $q_j$, следует, что НМТ действует в соответствии с одной из команд, начинающихся с пары $a_iq_j$. Пусть, например, в программе НМТ присутствуют ровно две команды с началом $a_iq_j$: $a_iq_j \to a_{i_1}q_{j_1}L$ и $a_iq_j \to a_{i_2}q_{j_2}R$. Тогда высказывание $R_{s,t,i,j}$ имеет следующий вид:

$$R_{s,t,i,j} = \bar{P}^i_{s,t} \vee \bar{Q}^j_t \vee P^{i_1}_{s,t+1}Q^{j_1}_{t+1}S_{s-1,t+1} \vee P^{i_2}_{s,t+1}Q^{j_2}_{t+1}S_{s+1,t+1}.$$

Высказывание $F_{s,t}$ имеет вид:

$$F_{s,t} = \bar{S}_{s,t} \,\&\, \left( \underset{1\le i\le l}{\&} (\bar{P}^i_{s,t} \vee P^i_{s,t+1}) \right) \vee S_{s,t} \,\&\, \left( \underset{1\le i\le l}{\&} \; \underset{1\le j\le r}{\&} R_{s,t,i,j} \right).$$

Заметим, что формулы для $R_{i,j}$ и $F_{s,t}$, вообще говоря, не являются КНФ. Однако каждую из них можно представить, например, совершенной КНФ. Важным является то, что при фиксированных $s$ и $t$ число переменных, от которых зависят эти формулы, ограничено константой, зависящей только от $l$ и $r$. Поэтому после перехода к КНФ получатся формулы, длина которых также ограничена некоторой константой, зависящей только от $l$ и $r$.

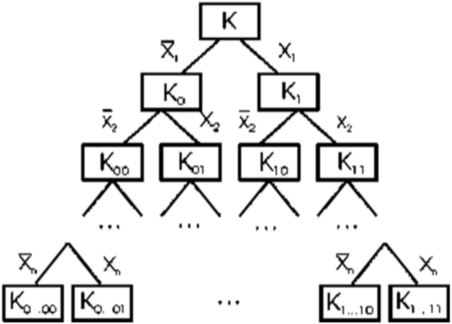
Наконец, формула $G$ утверждает, что на некотором шаге НМТ придет в принимающее заключительное состояние. Пусть таковым является $q_r$. Имеем

$$G = Q_1^r \vee Q_2^r \vee \cdots \vee Q_T^r.$$

Нетрудно проверить, что построенная таким образом формула $A$ обладает всеми требуемыми свойствами. Первая часть теоремы доказана.

*Доказательство второй части.* Пусть в КНФ $K$ встречаются переменные $x_1, ..., x_n$ (не обязательно существенные для реализуемой этой КНФ булевой функции). Кодом КНФ $K$ является слово $W(K)$ в алфавите $\{0, 1, \&, \vee, (,)\}$, полученное заменой в $K$ каждой буквы $x_i^\sigma$ двоичным словом вида $\sigma\alpha_1 \dots \alpha_m$, где $\alpha_1 \dots \alpha_m$ является двоичным разложением числа $i - 1$, причем $m = \lceil \log_2 n \rceil$.

Требуется доказать существование недетерминированной машины Тьюринга (сокращенно НМТ), распознающей язык ВЫП за полиномиальное время. Содержательно говоря, алгоритм распознавания выполнимости КНФ $K$ состоит в следующем. Имея на входе КНФ $K$, НМТ $M$ сначала строит две КНФ $K_0$ и $K_1$, получающиеся из $K$ путем подстановки вместо переменной $x_1$ соответственно констант 0 и 1. Затем НМТ $M$ повторяет процедуру подстановки констант 0 и 1 вместо переменной $x_2$ и т. д. Всякий раз НМТ $M$ прослеживает процесс подстановки параллельно.


K
$\bar{x}_1$
$x_1$
$K_0$
$K_1$
$\bar{x}_2$
$x_2$
$\bar{x}_2$
$x_2$
$K_{00}$
$K_{01}$
$K_{10}$
$K_{11}$
$\bar{x}_n$
$x_n$
$\bar{x}_n$
$x_n$
$K_{0\ldots00}$
$K_{0\ldots01}$
$K_{1\ldots10}$
$K_{1\ldots11}$

В результате после осуществления $n$ подстановок всякий раз будет получена некоторая КНФ с константами вместо переменных. Проверка равенства каждой из таких КНФ единице, очевидно, может быть осуществлена детерминированной машиной Тьюринга (а, значит, и НМТ) за время (число шагов), не превосходящее $O(L)$, где $L$ — длина кода $W(K)$. Если соответствующая КНФ равна единице, НМТ останавливается в принимающем состоянии, если КНФ равна нулю, то — в отвергающем. КНФ $K$ выполнима, если хотя бы при одной подстановке констант НМТ останавливается в принимающем состоянии.

Нетрудно составить программу НМТ, которая, имея на входе код $W(K)$ КНФ $K$, реализует описанные выше преобразования. НМТ будет иметь счетчик индексов переменных для того, чтобы знать, вместо какой переменной в данный момент подставляются константы. Выработать код индекса переменной на очередном шаге можно, прибавив 1 к двоичному счетчику индексов. Для этого требуется не более $O(m) = O(\log_2 n)$ шагов.

Подстановку константы вместо буквы $x_i^\sigma$ можно представлять себе как подстановку специальных символов $0^*$ и $1^*$ вместо первой координаты кода буквы $x_i^\sigma$. При этом мы полагаем, что символы $0^*$ и $1^*$ входят в алфавит НМТ и отличны от символов 0 и 1, используемых для кодирования индексов переменных. Осуществляя преобразования, связанные с подстановкой констант вместо переменной $x_i^\sigma$, НМТ сначала делает выбор, какую из констант подставлять (недетерминированная часть $i$-го этапа) а затем подстановка осуществляется детерминированным алгоритмом. Именно, следует отыскать в слове код очередной буквы вида $x_i^\sigma$ и заменить его первый разряд одним из двух символов $0^*$ и $1^*$.

Для распознавания того, что заданное подслово длины $m = \lceil \log_2 n \rceil$ слова $W$ длины $L$ совпадает с двоичной записью числа $i-1$, хранящейся на ленте, скажем, непосредственно перед самим словом, достаточно $O(mL)$ шагов. (Это легко пояснить, реализовав идею «протаскивания» слова длины $m$ через слово длины $L$ так, чтобы в пределах границ общей части этих слов их буквы чередовались; при этом буквы слова длины $m$ лучше вначале заменить их дублями $\hat{0}$, $\hat{1}$. Отметим, что указанного числа шагов $O(mL)$ достаточно, чтобы найти в слове длины $L$ все фрагменты, совпадающие со словом длины $m$). Число таких замен не превосходит $L$.

Таким образом, при заданном коде индекса переменной замену последней на константу можно осуществить не более чем за $O(L^2 \log_2 n)$ шагов. Ясно, что число шагов в каждом вычислении не превосходит $O(nL^2 \log_2 n) = O(L^3)$, где $L$ — длина входа. За это время НМТ придет в некоторое заключительное состояние. Если среди заключительных состояний окажется хотя бы одно принимающее, то КНФ $K$ выполнима. В противном случае она не является выполнимой. Таким образом, построенная НМТ распознает язык ВЫП за время $O(L^3)$. Теорема доказана.

## О задачах выполнимости КНФ

В этом разделе рассматриваются некоторые разновидности задач о выполнимости КНФ. Определим $k$-КНФ как конъюнкцию скобок, каждая из которых является дизъюнкцией не более $k$ букв, каждая из которых является переменной или ее отрицанием. Задача $k$-ВЫП состоит в распознавании выполнимости произвольной $k$-КНФ. Здесь доказывается полиномиальность задачи 2-ВЫП и $NP$-полнота задачи 3-ВЫП.

Определим задачу $k$-ВЫП (где $k \in \mathbb{N}$) ее входом и свойством.
**ВХОД:** $k$-КНФ $K(x_1, ..., x_n)$.
**СВОЙСТВО:** выполнимость.

**Теорема 2.** *2-ВЫП* $\in \mathbf{P}$.

*Доказательство.* Мы представим полиномиальный алгоритм распознавания 2-выполнимости. Идея алгоритма состоит в переходе от КНФ $K$, выполнимость которой требуется установить, к новой КНФ $K'$, выполнимой тогда и только тогда, когда $K$ выполнима, и содержащей на одну переменную меньше. Мы оценим число операций, необходимых для такого перехода. Ясно, что число самих переходов не превосходит числа $n$ переменных, от которых зависит исходная КНФ. После осуществления не более чем $n-1$ переходов такого типа мы получим КНФ $K_w$, реализующую функцию одной переменной. Ее выполнимость устанавливается за число шагов, ограниченное константой. Таким образом, доказательство будет состоять в описании перехода от $K$ к $K_w$ и оценки числа шагов, достаточных для его осуществления.

*Кодирование КНФ.* Буквы (т. е. переменные и их отрицания) кодируются двоичными векторами длины $\lceil \log_2 n \rceil + 1$, где $n$ — наибольший индекс переменной в исходной КНФ. Первая координата вектора, являющегося кодом некоторой буквы, равна 1, если буква является переменной, и равна 0, если буква является отрицанием переменной. Остальные координаты вектора представляют собой двоичную запись уменьшенного на 1 индекса переменной. Символы (, ), & и $\vee$ включаются в кодирующий алфавит. Таким образом, например, КНФ $K = x_1 \,\&\, (x_2 \vee \bar{x}_3)$ кодируется словом $W(K) = 100 \,\&\, (101 \vee 010)$.
Скобки считаются *одинаковыми*, если они совпадают или отличаются лишь порядком букв. Предполагается, что исходная КНФ не содержит одинаковых скобок и скобок вида $(x \vee x)$ и $(x \vee \bar{x})$.

За $O(l^2 \log_2^2 n)$ шагов запись произвольной КНФ $K$, содержащей $l$ букв и зависящей от $n$ переменных, можно преобразовать в запись эквивалентной ей КНФ $\tilde{K}$ *канонического вида*, т. е. либо такой КНФ, в которой нет одинаковых скобок и скобок вида $(x \vee x)$ и $(x \vee \bar{x})$, и при этом все однобуквенные сомножители (если они есть) зависят от попарно различных переменных, либо КНФ вида $x \,\&\, \bar{x}$. Действительно, достаточно $O(l)$ раз провести процедуру избавления от нежелательных повторов множителя, попросту затирая эти повторы (каждая такая процедура потребует, как мы видели, $O(\log_2 n \cdot l \log_2 n) = O(l \log_2^2 n)$ шагов), а затем переписать КНФ «набело» уже без пропусков за $O((l \log_2 n)^2)$ шагов. В случае же наличия противоположных однобуквенных сомножителей следует сделать вывод о невыполнимости КНФ.

Ясно, что общее количество букв $l(\tilde{K})$ в КНФ $\tilde{K}$ канонического вида не превосходит $l_n = \max(4 \cdot \binom{n}{2} + n, 2) \leq 2n^2$, так что $n \leq l(\tilde{K}) \leq 2n^2$; для длины $L$ кода $W(\tilde{K})$ верны неравенства: $n \log_2 n \leq L \leq 2n^2(4 + \log_2 n)$. В дальнейшем будем считать, что исходная КНФ — канонического вида.

*Случай, когда $K$ содержит однобуквенные сомножители.* Заметим, что для обнаружения однобуквенного сомножителя в КНФ $K$ по ее записи длины $L$ достаточно $O(L)$ шагов. В случае, когда буква $x$ является однобуквенным сомножителем КНФ $K$, переход к КНФ $K'$, не содержащей букв $x$ и $\bar{x}$, осуществляется с помощью следующих основных преобразований:

$$x\&x = x, \quad x\&\bar{x}\&F = x\&\bar{x}, \quad x\&(x\vee y) = x, \quad x\&(\bar{x}\vee y) = x\&y.$$

После выполнения этих преобразований (а также преобразований, сводящихся к использованию коммутативности и ассоциативности операций & и $\vee$) полученная формула $A$ содержит не более одного вхождения каждой из букв $x$ и $\bar{x}$. При этом либо $A = x\bar{x}$, либо $A = x^{\sigma}$, либо $A = x^{\sigma}K'$, где $K'$ — КНФ, не зависящая от $x$. Ясно, что в первом случае исходная КНФ не является выполнимой, во втором она выполнима, а в третьем она выполнима тогда и только тогда, когда выполнима КНФ $K'$, не содержащая ни $x$, ни $\bar{x}$. В первых двух случаях алгоритм заканчивает работу. В третьем получаем КНФ $K'$ с меньшим числом переменных и с не большей длиной кода, чем исходная КНФ.

Оценим число шагов, достаточное для осуществления соответствующих преобразований. Каждое из них сводится к нахождению кода буквы $x$ или $\bar{x}$, т. е. подслова длины $\lceil \log_2 n \rceil + 1$, в коде КНФ $K$ длины $L$ и последующем вычеркивании его или замене всего слова на $x\bar{x}$. Нетрудно убедиться в том, что каждое из этих преобразований требует не более $O(L \log_2 n)$ шагов на обычной (детерминированной) машине Тьюринга, а код КНФ $K$ в код КНФ $K'$ можно преобразовать не более чем за $O(lL \log_2 n) = O(L^2)$ шагов. КНФ $K'$ далее следует привести к каноническому виду, на что потребуется еще $O(L^2)$ шагов.

*Случай, когда однобуквенные сомножители отсутствуют.* Пусть КНФ $K$ не содержит однобуквенных сомножителей и имеет канонический вид. Пусть $l$ — число букв в $K$, $n$ — число переменных, $L$ — длина записи. Тогда переход к КНФ $K'$ осуществляется следующим образом. Выбираем некоторую букву $x$ в КНФ $K$. Пусть $K_0$ представляет собой конъюнкцию всех дизъюнкций вида $(\bar{x} \vee y)$ (где $y$ — буква, отличная от $x$ и $\bar{x}$), входящих в $K$, $K_1$ представляет собой конъюнкцию всех дизъюнкций вида $(x \vee z)$ (где $z$ — буква, отличная от $x$ и $\bar{x}$), входящих в $K$, а $K_2$ представляет собой конъюнкцию всех остальных дизъюнкций, входящих в $K$. Таким образом, КНФ $K_0$ представима в виде $K_0 = \mathop{\&}\limits_{1 \leq i \leq k} (\bar{x} \vee y_i)$, а $K_1$ представима в виде $K_1 = \mathop{\&}\limits_{1 \leq j \leq m} (x \vee z_j)$.

Заметим, что

$$K_0 K_1 K_2 = (\bar{x} \vee y_1 \cdots y_k)(x \vee z_1 \cdots z_m) K_2.$$

Легко проверить, что формула $(\bar{x} \vee Y)(x \vee Z)$, в которой $Y$ и $Z$ не зависят от $x$, выполнима тогда и только тогда, когда выполнима формула $Y \vee Z$. С учетом того, что $K_2$ не зависит от $x$, аналогично получаем, что формула $K_0 K_1 K_2$ выполнима тогда и только тогда, когда выполнима формула

$$(y_1 \cdots y_k \vee z_1 \cdots z_m) K_2 = \mathop{\&}_{1 \leq i \leq k} \mathop{\&}_{1 \leq j \leq m} (y_i \vee z_j) K_2 = \hat{K}.$$

КНФ $\hat{K}$ не содержит ни $x$, ни $\bar{x}$, а число букв в ней не больше $l + km \leq l + 4n^2 = O(n^2)$.

Нетрудно построить машину Тьюринга, преобразующую код $W(K)$ в код $W(\hat{K})$ длины $O(n^2 \log_2 n)$ за $O(n^2 \cdot \log_2 n \cdot n^2 \log_2 n) = O(n^4 \log_2^2 n)$ шагов. КНФ $\hat{K}$, возможно, содержит скобки вида $(y \vee \bar{y})$ или $(y \vee y)$, а также скобки, отличающиеся лишь порядком слагаемых. Удаление этих вхождений можно осуществить за число шагов, не превосходящее $O(\hat{l}^2 \log_2^2 n)$, где $\hat{l}$ — число букв в $\hat{K}$, а, значит, не более чем за $O(n^4 \log_2^2 n)$ шагов. В результате удаления лишних скобок получим КНФ $K'$ канонического вида с числом букв, не превосходящим $l_{n-1} \leq 2(n-1)^2$. Таким образом, переход от КНФ $K$ к КНФ $K'$ осуществим за $O(n^4 \log_2^2 n)$ шагов.

Поскольку число переходов не превосходит $n-1 \le L$, то для преобразования исходной КНФ в формулу, зависящую не более, чем от одной переменной, достаточно $O(n^5 \log_2^2 n) = O(L^5)$ шагов.
КНФ $K_w$ зависит не более, чем от одной переменной, и имеет длину, ограниченную некоторой константой.
Возможны следующие случаи: $K_w = x\bar{x}$, $K_w = (x \vee \bar{x})$, $K_w = x$, $K_w = \bar{x}$. Ясно, что ответ на вопрос о выполнимости в этом случае получается за $O(1)$ шагов. Таким образом, распознавание выполнимости 2-КНФ осуществимо за $O(L^5)$ шагов на детерминированной машине Тьюринга. Теорема доказана.

**Теорема 3.** *Задача 3-ВЫП является NP-полной.*

*Доказательство.* Покажем, что задача ВЫП полиномиально сводится к задаче 3-ВЫП. Для этого укажем, как преобразовать произвольную КНФ $K$ в 3-КНФ $K'$, выполнимую тогда и только тогда, когда КНФ $K$ выполнима.

Пусть $C = (y_1 \vee \dots \vee y_m)$ — скобка, являющаяся сомножителем КНФ $K$, и $m > 3$. Обозначим через $K_1$ КНФ, полученную из $K$ вычеркиванием скобки $C$. Пусть $u$ — переменная, не входящая в $K$. Положим $D = (y_1 \vee y_2 \vee u)(y_3 \vee \dots \vee y_m \vee \bar{u})$. Покажем, что КНФ $K$ выполнима тогда и только тогда, когда КНФ $D \,\&\, K_1$ выполнима.

Пусть $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ — набор, обращающий КНФ $K$ в единицу. Положим $g(x_1, \ldots, x_n) = y_1 \vee y_2$ и $h(x_1, \ldots, x_n) = y_3 \vee \cdots \vee y_2$. Тогда $g(\tilde{\alpha}) \vee h(\tilde{\alpha}) = 1$. Если $g(\tilde{\alpha}) = 1$, то набор $\tilde{\beta} = (\alpha_1, \ldots, \alpha_n, 0)$ обращает КНФ $D \,\&\, K_1$ в единицу (последняя координата набора $\tilde{\beta}$ есть значение переменной $u$). Если $h(\tilde{\alpha}) = 1$, то набор $\tilde{\beta} = (\alpha_1, \ldots, \alpha_n, 1)$ обращает КНФ $D \,\&\, K_1$ в единицу.
Пусть теперь $\tilde{\beta} = (\alpha_1, \ldots, \alpha_n, \beta)$ — набор, обращающий КНФ $D \,\&\, K_1$ в единицу. Пусть сначала $\beta = 0$. Тогда $g(\tilde{\alpha}) = 1$ и, значит, $K(\tilde{\alpha}) = 1$. Если же $\beta = 1$, то $h(\tilde{\alpha}) = 1$ и потому $K(\tilde{\alpha}) = 1$.

Указанное выше преобразование КНФ $K$ в КНФ $D \,\&\, K_1$ уменьшает на единицу число букв в скобке $C$ и увеличивает общее число букв на 2. Пусть $m_1, ..., m_k$ — все бо́льшие трех количества букв в скобках КНФ $K$. Тогда достаточно добавить аналогичным способом не более $2(m_1 + \cdots + m_k - 3k)$ букв с тем, чтобы получить 3-КНФ $K^*$, выполнимую тогда и только тогда, когда КНФ $K$ выполнима.
Полиномиальность преобразования очевидна. Теорема доказана.

## Некоторые *NP*-полные задачи

В этом разделе расширяется список $NP$-полных задач. Доказывается $NP$-полнота задач 0-1 ЦЕЛОЧИСЛЕННОЕ [ЛИНЕЙНОЕ] ПРОГРАММИРОВАНИЕ, КЛИКА, ВЕРШИННОЕ ПОКРЫТИЕ, ПОКРЫТИЕ МНОЖЕСТВА. Доказательство $NP$-полноты очередной задачи проводится путем свед́ения к ней одной из уже известных $NP$-полных задач. Свед́ение состоит в преобразовании входов некоторой задачи во вход исследуемой задачи с условием, что соответствующие свойства одновременно выполняются или не выполняются для рассматриваемых задач. Принадлежность задач классу **NP**, как правило, является очевидной и не доказывается. Полиномиальность преобразования входов также легко усматривается.

## Задача 0-1 ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ (0-1 ЦЛП)

**ВХОД:** Матрица $A = (a_{ij})$ размера $p \times n$ и целочисленный вектор $b = (b_1, ..., b_p)$.
**СВОЙСТВО:** Существует 0-1-вектор $\alpha = (\alpha_1, ..., \alpha_n)$ такой, что $A\alpha^T \geq b^T$.

**Теорема 4.** *ВЫП* $\prec$ *0-1 ЦЛП.*

*Доказательство.* Пусть $K = C_1 \,\&\, \ldots \,\&\, C_p$ — произвольная КНФ с $p$ скобками, зависящая от переменных $x_1, \ldots, x_n$. Для $i = 1, ..., p$, $j = 1, ..., n$ положим

$$a_{ij} = \begin{cases} 1, & \text{если буква } x_j \text{ встречается в скобке } C_i, \\ -1, & \text{если буква } \bar{x}_j \text{ встречается в скобке } C_i, \\ 0, & \text{иначе.} \end{cases}$$

а $b_i$ приравняем к разности числа 1 и числа отрицаний переменных в скобке $C_i$.

Очевидно, что вход задачи 0-1 ЦЛП можно задать словом длины, не превосходящей $O(np)$, а преобразование записи КНФ $K$ в запись входа задачи 0-1 ЦЛП можно осуществить на детерминированной машине Тьюринга за полиномиальное время от длины записи КНФ $K$.
Покажем, что 0-1-вектор $\alpha = (\alpha_1, ..., \alpha_n)$ такой, что $A\alpha^T \geq b^T$, существует тогда и только тогда, когда КНФ $K$ выполнима.

Заметим: скалярное произведение всякого вектора $(a_{i1}, \dots, a_{in})$ (образующего $i$-ю строку в матрице $A$, соответствующую скобке $C_i$) на 0-1-вектор $\alpha = (\alpha_1, ..., \alpha_n)$ является целым числом и достигает своего минимума (равного взятому со знаком «минус» числу букв с отрицаниями в скобке $C_i$) — по всем 0-1-векторам длины $n$ — на всяком таком и только таком векторе $\alpha$, в котором на местах тех переменных, чьи буквы встречаются в скобке $C_i$ без отрицаний, находятся нули, а на местах тех переменных, чьи буквы встречаются в скобке $C_i$ с отрицаниями, находятся единицы. Но в точности на всех таких векторах $\alpha$ и только на них скобка $C_i$ обращается в нуль.

Выполнимость КНФ $K$ равносильна существованию набора $\alpha = (\alpha_1, ..., \alpha_n)$ значений переменных такого, что $K(\alpha) = 1$. Это, в свою очередь, равносильно выполнению для всех $i = 1, \ldots, p$ равенств $C_i(\alpha) = 1$, что, в силу приведенного выше рассуждения, эквивалентно тому, что всякое скалярное произведение строки матрицы $A$ на вектор $\alpha$ не достигает своего минимума и потому (будучи целочисленным) не меньше чем $b_i$. А это и означает, что выполнено неравенство $A\alpha^T \geq b^T$. Теорема доказана.

**Пример.** Для КНФ $K = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2 \vee x_3)$ входом соответствующей задачи 0-1 ЦЛП являются матрица

$$A = \begin{pmatrix} 1 & 1 & 0 \\ -1 & -1 & 1 \end{pmatrix}$$

и вектор $b = (1, -1)$. Решениями неравенства $A\alpha^T \geq b^T$ являются векторы $(0,1,0)$, $(0,1,1)$, $(1,0,0)$, $(1,0,1)$ и $(1,1,1)$. Они же обращают КНФ $K$ в единицу.

## Задача КЛИКА

**ВХОД:** Граф $G = (V, E)$, число $k$.
**СВОЙСТВО:** В $G$ существует полный подграф с $k$ вершинами ($k$-*клика*).

*Теорема 5.* ВЫП $\prec$ КЛИКА.

*Доказательство.* Пусть $K = C_1 \mathbin{\&} \dots \mathbin{\&} C_p$ — произвольная КНФ с $p$ скобками, зависящая от переменных $x_1, \dots, x_n$. Для $i = 1, ..., p$ пусть $C_i = (y_{i1} \vee \dots \vee y_{ik_i})$, где $y_{ij}$ — некоторая буква, т. е. переменная или отрицание переменной. Положим

$$V = \{\langle y, i\rangle \mid y \text{ есть буква из скобки } C_i,\ 1 \leq i \leq p\},$$

$$E = \{(\langle y, i\rangle, \langle z, j\rangle) \mid i \neq j, y \neq \bar{z}\},$$

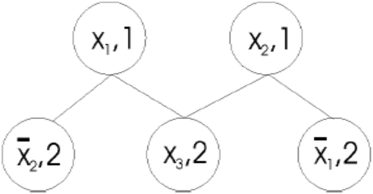
$$k = p.$$

Число вершин графа $G$ не превосходит $np$, а число ребер не превосходит $(np)^2$. Поэтому вход задачи КЛИКА можно закодировать словом, длина которого ограничена полиномом от длины записи КНФ К. Ясно также, что существует машина Тьюринга, преобразующая запись КНФ $K$ в запись графа $G$ и число $k$ за полиномиальное от длины записи КНФ $K$ время.

Покажем, что определенный выше граф $G$ содержит $k$-клику тогда и только тогда, когда КНФ $K$ выполнима.

Выполнимость КНФ $K$ равносильна существованию набора $\alpha = (\alpha_1, ..., \alpha_n)$ значений переменных такого, что $K(\alpha) = 1$. Это, в свою очередь, равносильно выполнению для всех $i = 1, \ldots, p$ равенств $C_i(\alpha) = 1$, т. е. в каждой скобке $C_i$ найдется буква $y_{ij_i}$, равная единице на наборе $\alpha$. Это, в силу определения графа $G$, эквивалентно тому, что все вершины $\langle y_{ij_i}, i\rangle$ $(i = \overline{1, p})$ попарно соединены ребрами (т. е. вершины соответствуют попарно различным скобкам и среди их букв нет противоположных букв одной переменной) и образуют клику порядка $p$, т. е. $k$-клику $(k = p)$. Теорема доказана.

**Пример.** Для КНФ $K = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2 \vee x_3)$ входом соответствующей задачи КЛИКА являются граф $G$, показанный на рисунке, и число 2.



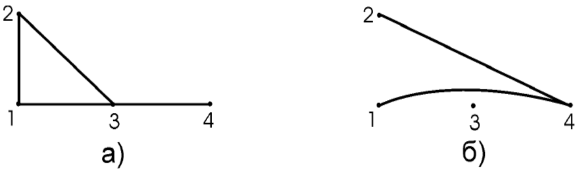
## Задача ВЕРШИННОЕ ПОКРЫТИЕ

**ВХОД:** Граф $G' = (V', E')$, число $l$.
**СВОЙСТВО:** Существует множество вершин $R$ такое, что $|R| \leq l$ и при этом каждое ребро графа $G'$ инцидентно некоторой вершине из $R$.

**Теорема 6.** *КЛИКА* $\prec$ *ВЕРШИННОЕ ПОКРЫТИЕ.*

*Доказательство.* Отображение входов имеет вид: $G' = (V', E')$ есть дополнение графа $G = (V, E)$, $l = |V| - k$. Заметим: множество $A$, $A \subseteq V$, является кликой в $G$ тогда и только тогда, когда $V \setminus A$ является вершинным покрытием дополнения $\bar{G}$ этого графа. Действительно, если $A$ — клика в $G$, то никакое ребро в $\bar{G}$ не соединяет никакие две вершины в $A$. Поэтому всякое ребро из $\bar{G}$ инцидентно хотя бы одной вершине из $V \setminus A$. Аналогично, если $V \setminus A$ является вершинным покрытием графа $\bar{G}$, то каждое ребро из $\bar{G}$ инцидентно хотя бы одной вершине из $V \setminus A$. Поэтому никакое ребро не соединяет две вершины из $A$, а значит, $A$ — клика в $G$. Теорема доказана.

**Пример.** Граф $G$ с множеством вершин $\{1, 2, 3, 4\}$ (см. рисунок (а)) содержит клику $\{1, 2, 3\}$. В графе $\bar{G}$ (см. рисунок (б)) дополнение этого множества покрывает все ребра.



## Задача ПОКРЫТИЕ МНОЖЕСТВА

**ВХОД:** Семейство $F = \{S_1, ..., S_m\}$ подмножеств множества $S$ такое, что $\bigcup\limits_{S_j \in F} S_j = S$, и число $h$.

**СВОЙСТВО:** Существует подсемейство $T \subseteq F$ такое, что $|T| \leq h$ и при этом $\bigcup\limits_{S_j \in T} S_j = S$.

**Теорема 7.** *ВЕРШИННОЕ ПОКРЫТИЕ* $\prec$ *ПОКРЫТИЕ МНОЖЕСТВА.*

*Доказательство.* Пусть задан вход задачи ВЕРШИННОЕ ПОКРЫТИЕ: граф $G' = (V', E')$ $(V' = \{v_1, \ldots, v_p\})$ и число $l$. Положим

$$S = E', \qquad S_j = \{(u, v_j) \mid (u, v_j) \in E', u \in V'\} \; (j = \overline{1, p}), \qquad h = l.$$

Очевидно, подсемейство $T = \{S_{i_1}, ..., S_{i_h}\}$ является покрытием множества $S$ (т. е. $\bigcup\limits_{S_j \in T} S_j = S$) тогда и только тогда, когда соответствующее подмножество вершин $\{v_{i_1}, ..., v_{i_h}\}$ графа $G' = (V', E')$ покрывает все ребра. Отсюда следует, что свойства задач ВЕРШИННОЕ ПОКРЫТИЕ и ПОКРЫТИЕ МНОЖЕСТВА выполняются или не выполняются одновременно. Теорема доказана.

**Спасибо за внимание!**