МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

им. М.В.Ломоносова

Факультет вычислительной математики и кибернетики

НЕКОТОРЫЕ ВОПРОСЫ СЛОЖНОСТИ АЛГОРИТМОВ

А. А. Сапоженко

Издательство факультета ВМиК МГУ
2001 г.

Аннотация

Пособие является частью курса “Основы кибернетики” и посвящено некоторым вопросам сложности алгоритмов. Излагаются результаты по алгоритмичесим трудностям синтеза схем и построения минимальных ДНФ, понятия сводимости и NP-полноты, устанавливается связь между временно́й сложностью вычислений на машинах Тьюринга и сложностью схем.

1 Введение

Пособие является частью курса “Основы кибернетики” и посвящено некоторым вопросам сложности алгоритмов. Излагаются результаты по алгоритмичесим трудностям синтеза схем и построения минимальных ДНФ, понятия сводимости и NP-полноты, устанавливается связь между сложностью схем и временно́й сложностью вычислений на машинах Тьюринга.

Параграф 2 посвящен алгоритмическим трудностям синтеза минимальных схем. Излагается один из первых математических результатов по алгоритмической сложности дискретных задач, полученный С.В.Яблонским в 1959г. Содержательный смысл его состоит в том, что некоторые задачи синтеза схем не могут быть решены без перебора.

В параграфе 3 дается понятие локального алгоритма, введенное Ю.И.Журавлевым и излагаются некоторые результаты из теории локальных алгоритмов. В частности, доказывается неразрешимость в классе локальных алгоритмов произвольного конечного индекса задачи о вхождении конъюнкции из сокращенной ДНФ булевой функции в хотя бы в одну минимальную ДНФ этой функции.

Параграфы 4 – 6 посвящены подходу к вопросу о сложности комбинаторных задач, основанному на понятии полиномиальной сводимости и теореме Ст. Кука. Суть этого подхода в том, что существует большой класс так называемых NP-полных задач, ни для одной из которых пока (к 2001г.) не удалось найти полиномиального алгоритма. Все эти задачи эквивалентны между собой в том смысле, что либо каждая из них решается эффективно, либо ни одна из них такого решения не имеет. Таким образом принадлежность некоторой задачи этому классу является некоторым доводом в пользу того, что она не имеет эффективного решения.

В параграфе 4 формулируются понятия полиномиальной сводимости, детерминированных и недетерминированных вычислений, NP-полноты и доказывается теорема Ст. Кука о том, что всякая задача из класса NP полиномиально сводится к задаче о выполнимости конъюнктивных нормальных форм (КНФ).

Параграф 5 посвящен некоторым разновидностям задач о выполнимости КНФ. Доказывается принадлежность задачи ВЫПОЛНИМОСТЬ классу NP, полиномиальность задачи 2-ВЫПОЛНИМОСТЬ и NP-полнота задачи 3-ВЫПОЛНИМОСТЬ.

В параграфе 6 расширяется список NP-полных задач. Доказывается NP-полнота задач 0-1 ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ, КЛИКА, ВЕРШИННОЕ ПОКРЫТИЕ, ПОКРЫТИЕ МНОЖЕСТВ, РАСКРАСКА.

В параграфе 7 устанавливается соотношение между схемной сложностью и временно́й сложностью тьюринговых вычислений. Доказывается теорема Сэвиджа, о том, что всякое вычисления на машине Тьюринга, осуществляемое за T шагов, может быть смоделировано схемой из функциональных элементов, сложность которой по порядку не превышает T^2.

Автор выражает признательность А.И.Савельевой за помощь в подготовке текста к печати.

2 Алгоритмические трудности синтеза схем

Этот параграф посвящен алгоритмическим трудностям синтеза минимальных схем. Излагается один из первых математических результатов по алгоритмической сложности дискретных задач, полученный С.В.Яблонским в 1959г. [1]. Суть его состоит в том, что решение задачи построения бесконечной последовательности функций, имеющих сложную схемную реализацию, так называемыми "правильными" алгоритмами непременно связана с построением всех функций алгебры логики. Данное изложение является переработкой статьи [2]. С целью упрощения доказательство ведется на примере схем из функциональных элементов (сокращенно, СФЭ), а не контактных схем как в оригинале.

Определение 2.1 *Множество функций $Q \subseteq P_2$ называется* **инвариантным классом**, *если наряду с каждой функцией $f \in Q$ оно содержит все функции, получающиеся из f применением следующих трех операций:*

1) *добавление и изъятие фиктивных переменных,*
2) *переименование переменных (без отождествления),*
3) *подстановка констант на места некоторых переменных.*

Инвариантными являются, например, классы линейных, монотонных функций. С другой стороны, классы T_0 и T_1 функций, сохраняющих константы, а также класс самодвойственных функций не являются инвариантными, ибо они не замкнуты относительно подстановки констант. *Тривиальными* инвариантными классами называются множество всех функций P_2 и пустое множество. Обозначим через $Q(n)$ множество всех $f \in Q$, зависящих (не обязательно существенно) от переменных $x_1, x_2, \ldots, x_n$.

Теорема 2.1 *Последовательность $\sqrt[2^n]{\mid Q(n) \mid}$ не возрастает и при этом $1 \leq \lim\limits_{n\to\infty} \sqrt[2^n]{\mid Q(n) \mid} \leq 2$ для всякого непустого инвариантного класса Q.*

Доказательство.
Пусть $n \geq 0$ и $f(x_1, \ldots, x_n, x_{n+1})$ — произвольная функция из $Q(n+1)$. Из разложения

$$f(x_1, \ldots, x_n, x_{n+1}) = x_{n+1} f(x_1, \ldots, x_n, 1) \vee \bar{x}_{n+1} f(x_1, \ldots, x_n, 0)$$

следует, что функция f полностью определяется парой своих подфункций $f(x_1, \ldots, x_n, 1)$ и $f(x_1, \ldots, x_n, 0)$. Поскольку последние также принадлежат классу Q, то

$$\mid Q(n+1) \mid \leq \mid Q(n) \mid^2.$$

Отсюда

$$\sqrt[2^{n+1}]{\mid Q(n+1) \mid} \leq \sqrt[2^n]{\mid Q(n) \mid}. \tag{1}$$

Из (1) вытекает невозрастание последовательности $\sqrt[2^n]{\mid Q(n) \mid}$. Покажем, что она ограничена. Если Q не пусто, то

$$1 = \sqrt[2^n]{1} \leq \sqrt[2^n]{\mid Q(n) \mid} \leq \sqrt[2^n]{2^{2^n}} = 2. \tag{2}$$

Следовательно, предел последовательности $\sqrt[2^n]{\mid Q(n) \mid}$ существует и заключен в сегменте $[1, 2]$. □

Из теоремы 2.1 следует, что $\lim_{n\to\infty} \sqrt[2^n]{\mid Q(n) \mid}$ можно представить в виде 2^σ, где $0 \leq \sigma \leq 1$. Число σ называется *характеристикой* инвариантного класса Q. Иногда мы будем указывать его в качестве индекса при Q или $Q(n)$. Из сказанного выше следует, что

$$\mid Q_\sigma(n) \mid = 2^{\sigma 2^n (1+\epsilon_n)}, \text{ где } \epsilon_n \to 0 \text{ при } n \to \infty. \tag{3}$$

Следствие 2.1 *Если инвариантный класс Q_σ не совпадает с P_2, то $\sigma < 1$.*

В самом деле, при некотором фиксированном m существует функция $g(x_1, \ldots, x_m) \notin Q$. Так как последовательность $\sqrt[2^n]{\mid Q(n) \mid}$ не возрастает, то

$$\lim_{n\to\infty} \sqrt[2^n]{\mid Q(n) \mid} \leq \sqrt[2^m]{\mid Q(m) \mid} \leq (2^{2^m} - 1)^{2^{-m}} < 2.$$

Отсюда вытекает утверждение. □

Теорема 2.2 *Существует инвариантный класс Q с характеристикой $\sigma = \frac{1}{2}$.*

Доказательство.
Рассмотрим класс Q, состоящий из функций $f(x_1, \ldots, x_n)$, представимых в виде

$$f(x_1, \ldots, x_n) = l(x_1, \ldots, x_n) \& g(x_1, \ldots, x_n), \tag{4}$$

где l — линейная функция а g — произвольная функция из P_2, любая существенная переменная которой является существенной переменной функции l. Легко видеть, что Q является инвариантным классом.

Нижняя оценка $\mid Q(n) \mid$. Выберем линейную функцию l в (4) равной $x_1 \oplus \ldots \oplus x_n$. Пусть $B^{n,1}$ — множество двоичных наборов длины n с нечетным числом координат, равных 1. Ясно, что $N_l = B^{n,1}$ и $|B^{n,1}| = 2^{n-1}$. Среди

функций $g(x_1,\ldots,x_n)\in P_2$ найдется множество G из $2^{2^{n-1}}$ функций попарно отличающихся друг от друга на множестве $B^{n,1}$. Ясно, что число функций вида (4) с $l=x_1\oplus\ldots\oplus x_n$ и $g\in G$ равно $2^{2^{n-1}}$. Отсюда $2^{2^{n-1}}\leq\mid Q(n)\mid$.

Верхяя оценка $\mid Q(n)\mid$. При фиксированной функции $l=x_{i_1}\oplus\ldots\oplus x_{i_r}$ имеется не более $2^{2^{r-1}}\leq 2^{2^{n-1}}$ различных функций $f\in Q(n)$. Число линейных функций, зависящих от переменных $x_1,\ldots,x_n$, равно 2^{n+1}. Поэтому $\mid Q(n)\mid\leq 2^{n+1}2^{2^{n-1}}$. Таким образом

$$2^{2^{n-1}}\leq\mid Q(n)\mid\leq 2^{n+1}2^{2^{n-1}}.$$

Отсюда

$$\lim_{n\to\infty}\sqrt[2^n]{\mid Q(n)\mid}=2^{1/2}.$$

Тем самым инвариантный класс Q с характеристикой $\sigma=1/2$ построен. $\square$

Упражнение 1. Доказать, что класс линейных функций является инвариантным с характеристикой 0.

Упражнение 2. Доказать, что класс монотонных функций является инвариантным с характеристикой 0. (*Указание*: Воспользоваться тем, что число монотонных функций, зависящих от переменных $x_1,x_2,\ldots,x_n$, не превосходит $n^{\binom{n}{\lceil n/2\rceil}}$). Возникает следующий вопрос. Можно ли для любого σ такого, что $0\leq\sigma\leq 1$ построить инвариантный класс с характеристикой σ?

Ответ на этот вопрос заключается в следующем. При $\sigma=1$ таковым является множество P_2 всех булевых функций, а при $\sigma\in[0,1)$ ответом является теорема, формулировку которой мы здесь приведем:

Теорема 2.3 (С.В.Яблонский [2]) *Для любого $\sigma\in[0,1)$ существует континуум попарно различных инвариантных классов Q с характеристикой σ.*

Обозначим через $L(f)$ сложность минимальной схемы из функциональных элементов, реализующей функцию f, и пусть $L(n)=\max_{f\in P_2(n)}L(f)$, $L_Q(n)=\max_{f\in Q(n)}L(f)$. В [3] доказана следующая

Теорема 2.4 (О.Б.Лупанов (см.[3]))

$$L(n)=\frac{2^n}{n}(1+\delta_n),\tag{5}$$

где $\delta_n\to\infty$ при $n\to\infty$.

Следующая теорема также принадлежит О.Б.Лупанову

Теорема 2.5 *Если Q — инвариантный класс с характеристикой σ, то*

$$L_Q(n) \leq \sigma \frac{2^n}{n}(1 + \Delta_n), \tag{6}$$

где $\Delta_n \to 0$ при $n \to \infty$.

Доказательство.
Пусть k — целое число, $1 \leq k \leq n$. В силу (3) имеем

$$\mid Q_\sigma(k) \mid = 2^{\sigma 2^k (1+\epsilon_k)}, \tag{7}$$

где $\epsilon_k \to 0$ при $k \to \infty$. Можно считать, что функции $f(x_1, x_2, \ldots, x_k)$ из $Q(k)$ пронумерованы числами от 1 до m_k, где $m_k = |Q(k)|$. Функции f_i поставим в соответствие двоичный вектор $(t_1^i, t_2^i, \ldots, t_l^i)$ длины $l = \lceil \log m_k \rceil$, являющийся двоичным разложением числа $i-1$, $i = 1, \ldots, m_k$. Этот вектор назовем *кодом* функции $f(x_1, x_2, \ldots, x_k)$. В силу (7) имеем

$$l = \lceil \log m_k \rceil = \sigma 2^k (1 + \epsilon_k).$$

Покажем, как вычислить значение произвольной функции $f(x_1, x_2, \ldots, x_n)$ из $Q(n)$ на произвольном наборе $(\alpha_1, \alpha_2, \ldots, \alpha_n)$. Разложим $f(x_1, x_2, \ldots, x_n)$ по последним $n - k$ переменным:

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{(\alpha_{k+1}, \ldots, \alpha_n)} x_{k+1}^{\alpha_{k+1}}, \ldots, x_n^{\alpha_n} \& f(x_1, \ldots, x_k, \alpha_{k+1}, \ldots, \alpha_n). \tag{8}$$

Ясно, что функция f однозначно определяется набором из 2^{n-k} своих подфункций $f(x_1, \ldots, x_k, \alpha_{k+1}, \ldots, \alpha_n)$, каждая из которых принадлежит множеству $Q(k)$, а, значит, имеет свой код. Этот код однозначно определяется набором $(\alpha_{k+1}, \ldots, \alpha_n)$. Следовательно, для определения кода $(t_1^i, \ldots, t_l^i)$ достаточно вычислить l функций от $n - k$ переменных. По коду подфункции однозначно восстанавливается вектор ее значениий. Для этого достаточно вычислить 2^k булевых функций, зависящих от переменных $(t_1^i, \ldots, t_l^i)$ Далее, по вектору значений подфункции $f(x_1, \ldots, x_k, \alpha_{k+1}, \ldots, \alpha_n)$ и вектору $(\alpha_1, \ldots, \alpha_k)$ значений переменных $x_1, \ldots, x_k$, однозначно определяется значение функции $f(\alpha_1, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n)$.

В соответствии с вышесказанным можно следуюшим образом построить схему S_f, реализующую функцию f и состоящую из трех блоков (см. рис.1).

Блок A вычисляет код $(t_1^i, \ldots, t_l^i)$ подфункции $f(x_1, \ldots, x_k, \alpha_{k+1}, \ldots, \alpha_n)$ по значениям $\alpha_{k+1}, \ldots, \alpha_n$ переменных $x_{k+1}, \ldots, x_n$, т.е. вычисляет l функций этих переменных.

Положив $k = \lceil \frac{1}{2} \log_2 n \rceil$, в силу (5) получаем, что для сложности $L(A)$ блока A выполнено следующее:

$$L(A) \leq l \frac{2^{n-k}}{n-k}(1+\delta_{n-k}) \leq \sigma 2^k (1+\epsilon_k) \frac{2^{n-k}}{n-k}(1+\delta_{n-k}) \sim \sigma \frac{2^n}{n}. \qquad (9)$$

Рис.1

Блок B вычисляет по коду подфункции вектор ее значениий, т.е. вычисляет 2^k булевых функций, зависящих от l переменных. В силу (5) получаем, что сложность $L(B)$ блока B удовлетворяет неравенству

$$L(B) \leq 2^k \frac{2^l}{l}(1+\delta_l) \leq 2^k \frac{2^{\sigma(1+\epsilon_k)2^k}}{\sigma(1+\epsilon_k)2^k}(1+\delta_l) = O\Big(2^{\sigma(1+\epsilon_k)2^k}\Big). \qquad (10)$$

Блок C по столбцу значений подфункции $f(x_1, \ldots, x_k, \alpha_{k+1}, \ldots, \alpha_n)$ и вектору $(\alpha_1, \ldots, \alpha_k)$ вычисляет $f(\alpha_1, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n)$. Тем самым он вычисляет некоторую булеву, зависящую от $k + 2^k$ переменных. В силу (5) получаем, для сложности $L(C)$ блока C следующее неравенство

$$L(C) \leq \frac{2^{2^k+k}}{2^k+k}(1+\delta_{2^k+k}). \qquad (11)$$

Из (9) — (11) вытекает, что

$$L(S_f) \le L(A) + L(B) + L(C) \le \sigma\frac{2^n}{n} + 2^{O(\sqrt{n})}).$$

Отсюда следует утверждение. □

Определение 2.2 *Последовательность функций* $f_1(x_1)$, $f_2(x_1, x_2), \ldots,$ $f_n(x_1, \ldots, x_n), \ldots,$ *называется* **сложной**, *если* $L(f_n) = L(n)$.

Определение 2.3 *Алгоритм* A, *строящий бесконечную последовательность булевых функций* $\{f_i(x_1, \ldots, x_i)\}_{i=1}^{\infty}$ *из* P_2 *называется* **правильным**, *если он строит все функции минимального инвариантного класса, содержащего эту последовательность.*

Теорема 2.6 (С. В. Яблонский) *Любой правильный алгоритм, строящий сложную последовательность функций* $\{f_i(x_1, \ldots, x_i)\}_{i=1}^{\infty}$ *функций из* P_2, *строит все множество* P_2.

Доказательство.
Предположим противное. Тогда последовательность $\{f_i(x_1, \ldots, x_i)\}$ самых сложных функций содержится в некотором инвариантном классе $Q_\sigma \ne P_2$. При этом $\sigma < 1$ в силу следствия 2.1. По Теореме 6 для любой функции $g \in Q_\sigma(n)$ можно построить СФЭ S_g такую, что $L(S_g) \le \sigma\frac{2^n}{n}(1 + \Delta_n)$. Но класс Q_σ содержит некоторую самую сложную функцию $f(x_1, \ldots, x_n)$, для которой в силу (5) имеем $L(f) = L(n) > (1 - \delta_n)2^n/n$, где $\delta_n \to 0$ при $n \to \infty$. При достаточно больших n приходим к противоречию. □

Список литературы

[1] С.В.Яблонский, О невозможности элиминации перебора всех функций из P_2 при решении некоторых задач теории схем// ДАН СССР, 124, 1, 1959, С.44-47.

[2] С.В.Яблонский, Об алгоритмических трудностях синтеза минимальных контактных схем// В сб. Проблемы кибернетики, М: Наука, Вып. 2, С.75-121.

[3] О.Б.Лупанов, Асимптотические оценки сложности управляющих систем, Изд-во МГУ, 1984.

3 Локальные алгоритмы

В параграфе дается понятие локального алгоритма, введенное Ю.И.Журавлевым (см. [1] и [2]), и излагаются некоторые результаты из теории локальных алгоритмов. В частности, доказывается неразрешимость в классе локальных алгоритмов произвольного конечного индекса задачи о вхождении конъюнкции из сокращенной ДНФ булевой функции в хотя бы в одну минимальную ДНФ этой функции.

В первой части параграфа дается определение локального алгоритма и доказываются утверждения о распознавании свойств конъюнкций входить в тупиковые ДНФ. Показано, что свойства конъюнкции "входить в во все тупиковые" или "входить хотя бы в одну тупиковую ДНФ" произвольной функции можно осуществить с помощью локального алгоритма ограниченного индекса, т.е. на основе информации о строении некоторой окрестности ограниченного порядка рассматриваемой конъюнкции (см. следствия 3.1 и 3.2). Во второй части доказывается, что свойство конъюнкции входить хотя бы в одну минимальную ДНФ не распознается в общем случае алгоритмами любого конечного индекса (следствие 3.4). Изложение ведется с некоторыми упрощениями по сравнению с [2].

О вхождении конъюнкции в тупиковые ДНФ

Ниже речь идет о локальных алгоритмах, предназначенных для упрощения ДНФ. Входом алгоритма является сокращенная ДНФ. Алгоритм расставляет отметки над конъюнкциями. Отметки несут информацию о выполнении или невыполнении некоторых свойств. Вычисление отметок носит локальный характер в том смысле, что значение пометок является однозначной функцией окрестности рассматриваемой конъюнкции и пометок конъюнкций из этой окрестности. Понятие окрестности конъюнкции в ДНФ определяется по индукции.

Определение 3.1 Окрестность $S_0(K, D)$ нулевого порядка *конъюнкции K в ДНФ D определяется равенством*

$$S_0(K, D) = \{K\}.$$

Пусть окрестность $S_r(K, D)$ порядка $r \geq 0$ конъюнкции K в ДНФ D уже определена. Тогда **окрестность порядка** $r + 1$ *конъюнкции K в ДНФ D определяется равенством*

$$S_{r+1}(K, D) = \{L \in D : \exists K \in S_r(K, D) : N_K \cap N_L \neq \emptyset\}.$$

Пример. Пусть $D = D_f^{сокр} = K_1 \vee \vee K_2 \vee K_3 \vee K_4$, где $K_1 = \bar{x}_1\bar{x}_2$, $K_2 = \bar{x}_1x_3$, $K_3 = x_2x_3$, $K_4 = x_1x_2$. Тогда (см. рис. **??**) $S_0(K_2, D) = K_2$, $S_1(K_2, D) = \{K_1, K_2, K_3\}$ и $S_r(K_2, D) = \{K_1, K_2, K_3, K_4\}$, при $r \geq 2$.

Ниже рассматриваются следующие свойства конъюнкций K:

1^0 K входит во все тупиковые ДНФ функции f.

2^0 K входит в хотя бы в одну тупиковую ДНФ функции f.

3^0 K входит в хотя бы в одну минимальную ДНФ функции f.

Рис.2

Значения функции φ_i, вычисляющей пометку, относящуюся к свойству $i = 1, 2, 3$, выбираются следующим образом:

$$\varphi_i(K, S_r(K, D)) = \begin{cases} 1, & \text{если свойство выполнено,} \\ 0, & \text{если свойство не выполнено,} \\ -, & \text{если неизвестно, выполнено ли свойство.} \end{cases}$$

Функции φ_i обладают следующим *свойством локальности*: Для любых двух ДНФ D и D', содержащих слагаемое K, выполняется равенство

$$\varphi_i(K, S_r(K, D)) = \varphi_i(K, S_r(K, D')) \tag{12}$$

Равенство окрестностей подразумевает не только совпадение окрестностей как множеств, но и равенство пометок над соответствующими конъюнкциями.

Локальный алгоритм индекса r преобразует сокращенную ДНФ без пометок (или, что то же, с пометками вида " $-$ ") в ДНФ, состоящую из тех же конъюнкций, но с пометками, несущими информацию о вхождении их в ДНФ того или иного типа. При вычислении пометки над очередной конъюнкцией алгоритм использует информацию о ее окрестности порядка r с учетом уже вычисленных пометок над конъюнкциями из этой окрестности. Алгоритм работает следующим образом:

1. Нумеруются некоторым образом конъюнкции K из $D_f^{сокр}$. Все пометки над конъюнкциями в начальный момент равны " $-$ ".

2. Выбирается первая по номеру конъюнкция K. Вычисляются одна или несколько функций $\varphi_i, i = 1, 2, 3$, по паре $(K, S_r(K, D))$. В результате вычисления либо хотя бы одна пометка изменилась, либо нет. В первом случае

заново нумеруем конъюнкции и продолжаем процесс с учетом новых пометок. Во втором случае переходим к следующей по номеру конъюнкции. Если номер конъюнкции оказался последним, и ни одна отметка не изменилась после очередной перенумерации, алгоритм заканчивает работу. Результатом является ДНФ $D_f^{сокр}$ с теми пометками над конъюнкциями, которые удалось вычислить.

В дальнейшем используются следующие известные в теории ДНФ понятия и факты. Множество всех наборов $(\alpha_1, \ldots, \alpha_n)$ таких, что $\alpha_i \in \{0, 1\}$, $i = 1, \ldots, n$, называется *n-мерным единичным кубом* и обозначается через B^n. Положим $N_f = \{\tilde{\alpha} \in B^n : \ f(\tilde{\alpha}) = 1\}$. *Гранью* куба B^n называется множество g, для которого существует конъюнкция K, зависящая (не обязательно существенно) от переменных $x_1, \ldots, x_n$, такая, что $g = N_K$. *Размерностью* грани называется число $\log_2 |g|$. Грань размерности 1 называется *ребром*. Грань называется *интервалом* функции f, если $g \subseteq N_f$, и *максимальным интервалом* функции f, если g не содержится ни в каком другом ее интервале. Говорят, что вершина $\tilde{\alpha} \in B^n$ *покрывается* конъюнкцией K, если $\tilde{\alpha} \in N_K$. Конъюнкция K *поглощается* функцией f (*поглощается* ДНФ D), если $N_K \subseteq N_f$ (соответственно, если $N_K \subseteq N_D$). Через $D \setminus K$ обозначим ДНФ, полученную из D отбрасыванием слагаемого K. Другие используемые, но неопределяемые здесь понятия можно найти в [2] или [3].

Лемма 3.1 *Пусть f — произвольная булева функция. Тогда*

1. *ДНФ $D_f^{сокр}$ реализует функцию f.*

2. *Всякая тупиковая, а, значит, и всякая минимальная ДНФ функции f может быть получена из $D_f^{сокр}$ путем отбрасывания некоторых слагаемых.*

Конъюнкция $K \in D_f^{сокр}$ называется *ядровой*, если существует $\tilde{\alpha} \in N_K$ такая, что $L(\tilde{\alpha}) = 0$ для любой конъюнкции $L \in D_f^{сокр} \setminus K$. Множество всех ядровых конъюнкций функции f называется ее *ядром*.

Теорема 3.1 (W.V.Quine) *Конъюнкция $K \in D_f^{сокр}$ содержится во всех тупиковых ДНФ функции f тогда и только тогда, когда она входит в ядро этой функции.*

Доказательство.

Необходимость. Пусть конъюнкция K входит во все тупиковые ДНФ функции f, но не входит в ее ядро. Тогда для каждой вершины $\tilde{\alpha}_i \in N_K$ существует конъюнкция $K_i \in D_f^{сокр}$ такая, что $K_i(\tilde{\alpha}_i) = 1$. Поэтому удаление K из

$D_f^{сокр}$ приводит к эквивалентной ДНФ. Продолжив удаление слагаемых из $D_f^{сокр}$ с сохранением эквивалентности, придем к некоторой тупиковой ДНФ функции f, не содержащей K. Получаем противоречие.

Достаточность. Из определения ядра следует, что удаление ядровой конъюнкции K из $D_f^{сокр}$ приводит к ДНФ, которая не эквивалентна $D_f^{сокр}$, а, значит, не реализует f. Утверждение следует теперь из леммы 3.1. □

Следствие 3.1 *Свойство вхождения K во все тупиковые ДНФ функции f можно установить по $S_1(K, D_f^{сокр})$.*

Доказательство.
В силу теоремы Квайна достаточно доказать, что свойство конъюнкции K "входить в ядро" можно установить по ее окрестности $S_1(K, D_f^{сокр})$. В самом деле, достаточно выяснить, поглощается ли конъюнкция K дизъюнкцией конъюнкций, содержащихся в $S_1(K, D_f^{сокр})$ и отличных от K. □

Пучком с центром в $\tilde{\alpha}$ относительно ДНФ D называется множество

$$\Pi(\tilde{\alpha}, D) = \{K \in D : K(\tilde{\alpha}) = 1\}.$$

Пусть D — ДНФ, а K — слагаемое этой ДНФ. Точка $\tilde{\alpha} \in N_D$ называется *регулярной* относительно (K, D), если

1. $K(\tilde{\alpha}) = 1$,
2. Существует $\tilde{\beta} \in N_D$ такая, что $K(\tilde{\beta}) = 0$ и $\Pi(\tilde{\beta}, D) \subset \Pi(\tilde{\alpha}, D)$.

Пусть D — ДНФ, а K — слагаемое этой ДНФ. Конъюнкция K называется *регулярной* относительно D, если всякая точка $\tilde{\alpha} \in N_K$ является регулярной относительно (K, D).

Теорема 3.2 (Ю.И.Журавлев). *Конъюнкция $K \in D_f^{сокр}$ не содержится ни в одной тупиковой ДНФ функции f тогда и только тогда, когда она регулярна относительно $D_f^{сокр}$.*

Доказательство.
Необходимость. Пусть существует точка $\tilde{\alpha} \in N_K$, не являющаяся регулярной относительно $(K, D_f^{сокр})$. Удалим из $D_f^{сокр}$ все конъюнкции пучка $\Pi(\tilde{\alpha}, D_f^{сокр})$ за исключением K. Полученная ДНФ D все еще реализует функцию f, ибо в противном случае нашлась бы точка $\tilde{\beta} \in N_f$ такая, что $K(\tilde{\beta}) = 0$ и $\Pi(\tilde{\beta}, D_f^{сокр}) \subset \Pi(\tilde{\alpha}, D_f^{сокр})$, что противоречило бы предположению о том, что точка $\tilde{\alpha} \in N_K$, не является регулярной. Отбрасывая, если понадобится, некоторые слагаемые из ДНФ D, получим тупиковую ДНФ. При этом конъюнкция K не может быть отброшена, ибо только она покрывает точку $\tilde{\alpha}$.

Следовательно, существует тупиковая ДНФ функции f, содержащая конъюнкцию K.

Достаточность. Пусть конъюнкция K регулярна относительно $D_f^{сокр}$. Удалим K из $D_f^{сокр}$. ДНФ $D = D_f^{сокр} \setminus K$ эквивалентна $D_f^{сокр}$, ибо, в силу регулярности K, для всякой точки $\tilde{\alpha} \in N_K$ существует $\tilde{\beta} \in N_D$ такая, что $\tilde{\beta} \notin N_K$ и $\Pi(\tilde{\beta}, D_f^{сокр}) \subset \Pi(\tilde{\alpha}, D_f^{сокр})$. Последнее означает, что всякая точка $\tilde{\alpha} \in N_K$ покрыта некоторой конъюнкцией из D. Продолжая отбрасывание слагаемых из ДНФ D с сохранением эквивалентности, придем к тупиковой ДНФ, не содержащей K. □

Следствие 3.2 *Свойство конъюнкции K "быть регулярной относительно ДНФ $D_f^{сокр}$", а тем самым, и свойства "входить хотя бы в одну тупиковую ДНФ функции f", можно установить по ее окрестности $S_2(K, D_f^{сокр})$.*

Доказательство.

В самом деле, требуется установить для каждой точки $\tilde{\alpha}$ из N_K, является ли эта точка регулярной. Для этого надо сравнить пучки с центрами в точках из N_K с пучками, центры которых находятся во множествах вида N_L, где L — конъюнкция из $S_1(K, D)_f^{сокр}$. Но $\Pi(\tilde{\beta}, D_f^{сокр}) \subseteq S_2(K, D)_f^{сокр}$ для любой точки $\beta \in N_L$, где L — конъюнкция из $S_1(K, D)_f^{сокр}$. В то же время, $\Pi(\tilde{\alpha}, D_f^{сокр}) \subseteq S_1(K, D)_f^{сокр}$ для любой точки $\alpha \in N_K$. Таким образом, все конъюнкции, содержащиеся в рассматриваемых пучках принадлежат $S_2(K, D)_f^{сокр}$. □

Вхождение конъюнкции хотя бы в одну минимальную ДНФ

Расстоянием между двоичными наборами $\widetilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ и $\widetilde{\beta} = (\beta_1, \ldots, \beta_n)$ называется число

$$\rho(\widetilde{\alpha}, \widetilde{\beta}) = \sum_{i=1}^{n} \mid \alpha_i - \beta_i \mid .$$

Последовательность вершин $\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_m \in B^n$ называется *цепью*, если выполнены условия:

1. $\rho(\widetilde{\alpha}_i, \widetilde{\alpha}_{i+1}) = 1$, для всех $i = 1, \ldots, m-1$,
2. $\rho(\widetilde{\alpha}_i, \widetilde{\alpha}_j) > 1$, для всех i, j таких, что $\mid i - j \mid > 1$.

Функция $f(\widetilde{x})$ называется *цепной*, если можно упорядочить наборы из N_f так, что образуется цепь.

Лемма 3.2 *Пусть f - цепная функция и $\mid N_f \mid = m > 1$. Тогда*

1. *Все максимальные грани f являются ребрами.*
2. *Пусть наборы $\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_m$ множества N_f в указанном порядке образуют цепь. Положим $N_i = \{\widetilde{\alpha}_i, \widetilde{\alpha}_{i+1}\}, i = 1, \ldots, m-1$, и пусть K_i —конъюнкция, соответствующая ребру N_i. Тогда $D_f^{сокр} = K_1 \vee K_2 \vee \ldots \vee K_{m-1}$.*
3. *Пусть m четно, тогда $D = K_1 \vee K_3 \ldots \vee K_{m-1}$ — единственная минимальная ДНФ функции f.*

Доказательство.

Утверждение 1 следует из того, что, с одной стороны, каждая вершина содержится хотя бы в одном ребре, а с другой стороны, в силу свойства 2 определения цепи, грани размерности 2 отсутствуют.

Утверждение 2 следует из утверждения 1.

Минимальность ДНФ D следует из того, что все максимальные грани цепной функции являются ребрами, каждое из которых содержит две вершины, а для покрытия $2m$ вершин цепи требуется не менее m ребер.

Единственность минимальной ДНФ D, состоящей из конъюнкций с нечетными номерами, докажем от противного. Если бы существовала еще одна минимальная ДНФ $D' \neq D$, то в ней нашлась бы конъюнкция вида K_{2i} такая, что $N_{K_{2i}} = \{\widetilde{\alpha}_{2i}, \widetilde{\alpha}_{2i+1}\}$. Пусть i — наименьшее целое, такое что конъюнкция K_{2i} является слагаемым ДНФ D'. Конъюнкция K_1 входит в ДНФ D' как ядровая. Объединение множества ребер $N_1, \ldots, N_{2i-1}, N_{2i}$ содержит $2i-1$ вершин из N_f. Для покрытия оставшихся $2m - 2i + 1$ вершин требуется еще по меньшей мере $m - i + 1$ ребер. Следовательно, ДНФ D' имеет не менее $m + 1$ слагаемых. Отсюда и из пункта 1 вытекает, что D' не является минимальной. □

Замечание 3.1 *В дальнейшем подразумевается, что вершины множества N_f, а также конъюнкции в $D_f^{сокр}$ цепной функции f пронумерованы в соответствии с пунктом 2 леммы 3.2.*

Следствие 3.3 *Если функция f цепная, то при "естественной" нумерации конъюнкций из $D_f^{сокр}$ ни одна конъюнкция с четным номером не входит в ее единственную минимальную ДНФ.*

Последовательность вершин $\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_{2m} \in B^n, m > 2$ называется *циклом*, если выполнены условия:

1. $\rho(\widetilde{\alpha}_i, \widetilde{\alpha}_{i+1}) = 1$, для всех $i = 1, \ldots, 2m-1$;
2. $\rho(\widetilde{\alpha}_i, \widetilde{\alpha}_j) > 1$, для всех i, j таких, что $\mid i - j \mid > 1 \ (mod\ 2m)$,
3. $\rho(\widetilde{\alpha}_{2m}, \widetilde{\alpha}_1) = 1$.

Лемма 3.3 *Для любого $n \geq 3$ в B^n существует цикл с $2n$ вершинами.*

Доказательство.
Примером такого цикла в B^n является последовательность

$$S_n = (00\ldots00),\ (00\ldots01), \ldots,\ (11\ldots11),\ (11\ldots10), \ldots, (10\ldots00),$$

в которой число единиц в очередном наборе сначала увеличивается на 1, а затем, достигнув n, последовательно уменьшается на 1. □

Функция $f(\widetilde{x})$ называется *циклической*, если можно упорядочить наборы из N_f так, что образуется цикл. Обозначим через $D_f^{\Sigma M}$ ДНФ, состоящую из всех конъюнкций, входящих хотя бы в одну минимальную ДНФ функции f.

Лемма 3.4 *Пусть f - циклическая. Тогда*

1. *Все максимальные грани f являются ребрами;*
2. *Пусть точки $\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_{2s}$ множества N_f в указанном порядке образуют цикл. Пусть $N_i = \{\widetilde{\alpha}_i, \widetilde{\alpha}_{i+1}\}$, $i = 1, \ldots, 2s-1$, и $N_{2s} = \{\widetilde{\alpha}_{2s}, \widetilde{\alpha}_1\}$, — ребра этого цикла. Обозначим через K_i конъюнкцию, соответствующую ребру N_i. Тогда $D_f^{сокр} = K_1 \vee K_2, \vee \ldots \vee K_{2s}$.*
3. *Существует ровно две минимальные ДНФ:*

$$D_1 = K_1 \vee K_3 \vee \ldots \vee K_{2s-1} \qquad и \qquad D_2 = K_2 \vee K_4 \vee \ldots \vee K_{2s}.$$

4. $D_f^{\cup M} = D_f^{сокр}$.

Доказательство.
Утверждения 1 и 2 доказываются аналогично соответствующим пунктам леммы 3.2.

Минимальность ДНФ D_1 и D_2 также вытекает по соображениям, аналогичным тем, что использовались в лемме 3.2. То, что других минимальных ДНФ нет, вытекает из того, что если в ДНФ входят конъюнкции с номерами разной четности, то, как и в лемме 3.2, такая ДНФ содержит не менее $m+1$ конъюнкций. Утверждение 4 вытекает из пунктов 2 и 3. □

Теорема 3.3 (Ю.И.Журавлев). *Для любого r существует n и $f(\widetilde{x}^n)$ такие, что*

1. $D_f^{\cup M} = D_f^{сокр}$;

2. *Для любой конъюнкции $K \in D_f^{сокр}$ существует функция h_K такая, что*

$$S_r(K, D_f^{сокр}) = S_r(K, D_{h_K}^{сокр}) \tag{13}$$

и $K \notin D_{h_K}^{\cup M}$.

Доказательство.
Зафиксируем r. Пусть $n > r+3$. Пусть $f = f(x_1, \ldots, x_n)$ — циклическая функция такая, что множество N_f состоит из вершин цикла S_n, определенного в лемме 3.3. Тогда $D_f^{\cup M} = D_f^{сокр}$ в силу леммы 3.4.

Пусть $K \in D_f^{сокр}$ и m есть нечетное число из пары $(r, r+1)$. Определим функцию h_K следующим образом. Положим

$$N_{h_K} = \cup_{L \in S_m(K, D_f^{\text{сокр}})} N_L.$$

Ясно, что h_K — цепная функция и $|N_{h_K}| = 4m$. Кроме того, $K \in D_{h_K}^{сокр}$ в естественной нумерации слагаемых в $D_{h_K}^{сокр}$ конъюнкция K имеет номер $2m$, а, значит, в силу следствия 3.3, она не входит в единственную минимальную ДНФ функции h_K. □

Следствие 3.4 *Для любого r существуют число n и функция $f(\widetilde{x}^n)$ такие, что никакой локальный алгоритм A индекса r с произвольной функцией вычисления пометок φ, удовлетворяющей условию локальности* (12) *и имеющей вид:*

$$\varphi(K, S_r(K, D_f^{сокр})) = \begin{cases} 1, & \text{если K} \in \text{D}_\text{f}^{\cup\text{M}}, \\ 0, & \text{если K} \notin \text{D}_\text{f}^{\cup\text{M}}, \\ -, & \text{если значение } \varphi \text{ неизвестно}, \end{cases}$$

не может изменить отметку ни над одной конъюнкцией из $D_f^{сокр}$.

Доказательство.
В самом деле, алгоритм A индекса r принимает решение о вхождении и или невхождении конъюнкции K в ДНФ $D_f^{\cup M}$ только по окрестности конъюнкции K порядка r. На первом шаге алгоритма все пометки над конъюнкциями равны"-". Поэтому в силу равенства (13) свойство локальности (12) выполнено для определенной выше функции φ, любой конъюнкции K из $D_f^{сокр}$ и пары ДНФ (D, D'), где $D = D_f^{сокр}$ и $D' = D_{h_K}^{сокр}$. Поэтому значение функции φ не может быть определенным. В самом деле, значение пометки "0" противоречит тому, что $K \in D_f^{\cup M}$, а значение "1" — тому, что $K \notin D_{h_K}^{\cup M}$. Отсюда вытекает утверждение. □

Список литературы

[1] Ю.И.Журавлев, Теоретико-множественные методы в алгебре логики// В сб. Проблемы кибернетики, М.: Наука, Вып. 8, 1962г., С.5-44.

[2] Ю.И.Журавлев, Избранные научные труды, М. 1998г., 417 с.

[3] А.А.Сапоженко, Дизюнктивные нормальные формы, Издательство МГУ, 1975, 90 с.

4 Теорема Кука

Введение. В параграфе доказывается теорема Ст. Кука. Центральными понятиями являются (полиномиальная) сводимость языков, детерминированные и недетерминированные машины Тьюринга, классы P и NP, NP-полнота. При изложении использовались источники [1],[2] и [3].

Существует большой класс вычислительных задач, заключающийся в распознавании тех или иных свойств графов, целых чисел, массивов целых чисел (векторов, матриц и т.п.), конечных множеств, булевых формул и др. Посредством кодирования таких объектов множествами слов эти задачи могут быть превращены в задачи распознавания языков. Тем самым вопрос о сложности самых разнообразных вычислительных задач сводится к вопросу о сложности распознавания языков.

Принято считать, что задача решается эффективно, если существует алгоритм ее решения со временем работы, которое ограничено полиномом от размера входных данных. Впервые эту рабочую гипотезу выдвинул и стал защищиать Джек Эдмондс [4]. Теорема Ст. Кука и понятие полиномиальной сводимости позволяют доказать, что большой класс (т.н. NP-полных) задач, ни для одной из которых пока (к 2001г.) не удалось найти полиномиального алгоритма, эквивалентны между собой в том смысле, что либо каждая из них решается эффективно, либо ни одна из них такого решения не имеет.

Определения. Термин "машина Тьюринга" (сокращенно МТ) употребляется здесь для одноленточных *детерминированных* машин, (см., например, [5]). Некоторое (непринципиальное) отличие состоит в том, что мы рассматриваем МТ с односторонней 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, $w = b_1, b_2, ..., b_n$, 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.

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*. *Вы-*

числением МТ M на входе w называется последовательность конфигураций $C_1, C_2, ..., C_t, ...$, возникающая при работе над словом w. Подразумевается, что конфигурация C_{t+1} однозначно определяется конфигурацией C_t и командой МТ M, начинающейся с пары (b_k, q_j), где b_k — символ, обозреваемый МТ в момент t, а q_j — состояние МТ в момент t. *Время работы* или *число шагов $t_M(w)$ 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) от детерминированной состоит в том, что в программе НМТ для пары (a, q), где a — символ из алфавита МТ, а q — символ состояния, в ее программе может присутствовать несколько (но не более некоторого фиксированнного для заданной МТ числа) команд, начинающихся с aq. Без потери общности можно ограничиться случаем, когда паре aq может соответствовать 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_{t+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 сучае НМТ объединение цепей, соответствующих вычислениям на входе w, представляет собой ориентированное (от корня) дерево с корнем $C_1 = q_1 w$.

Распознавание языков. Пусть A — конечный алфавит. Через A^ω 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 всех слов $w \in L$ существует принимающее вычисление длины, не превышающей $p(||w||)$.

Через $\mathbf{P}$ обозначим класс языков, распознаваемых МТ за полиномиальное время. Через $\mathbf{\Pi}$ обозначим множество отображений вида $f : A^\omega \to A^\omega$, вычисляемых 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 (*полиномиально*) *эквивалентны* K, если $K \prec L$ и $L \prec K$. Класс языков, распознаваемых НМТ за полиномиальное время, обозначается через $\mathbf{NP}$. Язык L называется *NP-полным*, если

1) $L \in \mathbf{NP}$.

2) $K \in \mathbf{NP} \Rightarrow K \prec L$.

Справедливы следующие простые утверждения.

Утверждение 1. Если $K \in \mathbf{P}$ и $L \prec K$, то $L \in \mathbf{P}$.

Утверждение 2. $\mathbf{P} \subseteq \mathbf{NP}$.

Утверждение 3. Либо все NP-полные языки принадлежат $\mathbf{P}$, либо ни один из них не принадлежит $\mathbf{P}$. Первое имеет место тогда и только тогда, когда $\mathbf{P}$=$\mathbf{NP}$.

Утверждение 4. Если $L \prec K$ и $K \prec H$, то $L \prec H$.

Язык ВЫПОЛНИМОСТЬ (короче, ВЫП) состоит из слов в алфавите $A = \{(,), \&, \vee, \neg, x_i,\ i = 1, 2, ...\}$, представляющих собой выполнимые КНФ, т.е. КНФ, не равные тождественно 0.

Теорема.(Ст.Кук) Если $L \in \mathbf{NP}$, то $L \prec \mathit{ВЫП}$.

Доказательство. 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 $t_M(w) \leq p(||w||)$ 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 $f : L \to \mathit{ВЫП}$, удовлетворяющее условию $f(w) \in L \Leftrightarrow A(w) \in \mathit{ВЫП}$. 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, $w \in \Sigma$ — произвольное слово

длины n. Положим $T = p(n)$. Заметим, что если МТ заканчивает работу не более чем за $p(n)$ тактов, то ячейки ленты с номерами большими, чем T не посещаются головкой.

Введем 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^j_t, где $1 \le j \le r$; $1 \le t \le T$. Переменная Q^j_t истинна тогда и только тогда, когда на шаге t НМТ находится в состоянии q_j.

$S_{s,t}$, 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.

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

$$B_t = \left(S_{1,t} \bigvee S_{2,t} \bigvee ... \bigvee S_{T,t}\right) \wedge \left[\bigwedge_{1 \le i < j \le T} (\overline{S}_{i,t} \bigvee \overline{S}_{j,t})\right].$$

Для $1 \le t \le T$ 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.

$$E = Q^1_1\&S_{1,1}\&P^{i_1}_{1,1}\&P^{i_2}_{2,1}\&...\&P^{i_n}_{n,1}\&P^1_{n+1,1}\&...\&P^1_{T,1},$$

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

Формула F утверждает, что для каждого t преобразование слова на ленте, сдвиг головки 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, из того, что в обозреваемой ячейке ленты записан символ a_i и НМТ находится в состоянии q_j, следует, что НМТ действует в соответствии хотя бы с одной из команд, начинающихся с пары $a_i q_j$. Пусть, например, в программе НМТ присутствуют две команды с началом a_i, q_j: $a_i\, q_j \to a_{i_1} q_{j_1} L$ и $a_i\, q_j \to a_{i_2} q_{j_2} R$. Тогда высказывание $R_{s,t,i,j}$ имеет следующий вид:

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

Высказывание $F_{s,t}$ имеет вид

$$F_{s,t} = \overline{S}_{s,t} \& \left[\bigwedge_{1 \le i \le l} (\overline{P^i_{s,t}} \bigvee P^i_{s,t+1}) \right] \bigvee S_{s,t} \& \left[\bigvee_{1 \le i \le l} \bigvee_{1 \le j \le r} R_{s,t,i,j} \right]$$

Заметим, что формулы для $R_{s,t,i,j}$ и $F_{s,t}$ 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 НМТ. Поэтому после перехода к КНФ получатся формулы ограниченной длины.

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

$$G = Q^r_1 \bigvee Q^r_2 \bigvee \ldots \bigvee Q^r_T.$$

Нетрудно проверить, что построенная таким образом формула A обладает всеми требуемыми свойствами. □

Список литературы

[1] Кибернетический Сборник No 12 (Нов. серия), М. МИР, 1975, С. 5–10.

[2] А. Ахо, Д. Хопкрофт, Д. Ульман// Построение и анализ вычислительных алгоритмов, М., Мир, — 1979. — 536 С.

[3] М.Гэри, Д.Джонсон, Вычислительные машины и труднорешаемые задачи, М., Мир, — 1982. — 416 С.

[4] J. Edmonds, Paths, trees and flowers,// Canad. J. Math., VIII, C. 449-467.

[5] С. В. Яблонский “Введение в дискретную математику” М.: Наука, 1986.

5 О задачах выполнимости КНФ

Параграф посвящен некоторым разновидностям задач о выполнимости КНФ. Определим k-КНФ как конъюнкцию скобок, каждая из которых является дизъюнкцией не более k букв. Задача k-ВЫП состоит в распознавании выполнимости произвольной k-КНФ. Здесь доказывается принадлежность задачи ВЫП классу NP, полиномиальность задачи 2-ВЫП и NP-полнота задачи 3-ВЫП.

Пусть КНФ K зависит от переменных $x_1, ..., x_n$ (не обязательно существенно). Кодом КНФ K является слово $W(K)$ в алфавите $\{0, 1, \&, \vee\}$, полученное вычеркиванием символов скобок и заменой в K каждой буквы x_i^σ двоичным вектором вида $(\sigma, \alpha_1, ..., \alpha_m)$, где вектор $(\alpha_1, ..., \alpha_m)$ является двоичным разложением числа i, причем $m = \lceil \log n \rceil$.

Теорема 5.1 *Задача ВЫП*$\in NP$

Доказательство. Требуется доказать существование недетерминированной машины Тьюринга (сокращенно, НМТ), распознающей язык ВЫП. Определение НМТ дано выше (см. параграф 4). Содержательно говоря, алгоритм распознавания выполнимости КНФ K состоит в следующем. Имея на входе КНФ K, НМТ μ строит две КНФ K_0 и K_1, получающиеся из K путем подстановки вместо переменной x_1 соответственно констант 0 и 1. Затем НМТ μ повторяет процедуру подстановки констант 0 и 1 вместо переменной x_2 и т.д. Всякий раз НМТ μ прослеживает процесс подстановки параллельно (см. рис. 3).

Рис.3

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 $O(m) = O(\log n)$ шагов.

Подстановку константы вместо буквы x_i^σ можно представлять себе как подстановку специальных символов $0*$ и 1^* вместо первой координаты кода буквы x_i^σ. 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, 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^σ и заменить его первый разряд одним из двух символов $0*$ и 1^*. Для распознавания того, что заданное подслово длины $m = \lceil \log n \rceil$ слова W длины L совпадает с двоичной записью числа i, хранящейся на ленте, скажем, непосредственно перед самим словом, достаточно $O(mL)$ шагов. Число таких замен не превосходит L. Таким образом, при заданном коде индекса переменной замену последней на константу можно осуществить не более чем за $O(L^2 \log n)$ шагов. Ясно, что число шагов в каждом вычислении не превосходит $O(nL^2 \log n) = O(L^3)$, 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 $O(L^3)$. □

Следствие 5.1 *Задача ВЫП является NP-полной.*

Утверждение следует из теоремы Кука и теоремы 5.1.

Определим задачу k-ВЫП ее входом и свойством.
ВХОД: $F(x_1, \ldots, x_n)$ — 3-КНФ.
СВОЙСТВО: выполнимость.

Теорема 5.2 *2-ВЫП* $\in P$.

Доказательство. Мы представим полиномиальный алгоритм распознавания 2-выполнимости. Идея алгоритма сотоит 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 $n-2$ переходов такого типа мы получим КНФ K_ω, реализующую функцию одной переменной. Ее выполнимость устанавливается за число шагов, ограниченное константой. Таким образом доказательство будет состоять в описании перехода от K к K_ω и оценки числа шагов, достаточных для его осуществления.

Кодирование КНФ. Буквы (т.е. переменные и их отрицания) кодируются двоичныим векторами длины $\lceil \log n \rceil + 1$, 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 индекса переменной. Символы & и $\vee$ включаются в кодирующий алфавит. Скобки отбрасываются. Таким образом, например, КНФ $K = x_1 \& (x_2 \vee \bar{x}_3)$ кодируется словом $W(K) = 101 \& 110 \vee 011$.

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 $(x \vee \bar{x})$.

Если исходная КНФ K содержит l_1 однобуквенных сомножителей и l_2 двубуквенных скобок, а наибольший индекс переменной в ней равен n, то длина $L(W(K))$ ее кода равна, очевидно, $(l_1 + 2l_2)(\lceil \log n \rceil + 1) + l_1 + 2l_2 - 1 = (l_1 + 2l_2)(\lceil \log n \rceil + 2) - 1$. Заметим, что число связок & и $\vee$ на единицу меньше числа букв.

Выявление однобуквенных сомножителей. Если в КНФ K имеется однобуквенный сомножитель, то в подслове, состоящем только из связок & и $\vee$, встретятся два символа & подряд. Для обнаружения этой ситуации требуется порядка L шагов, где L — длина слова $W(K)$, кодирующего K.

Случай, когда K содержит однобуквенные сомножители. В случае, когда буква x является однобуквенным сомножителем КНФ K, переход

к КНФ K' осуществляется с помощью следующих преобразований:

$$(a)\ x\&x = x;\ \ (b)\ x\&\bar{x}\&F = x\&\bar{x};\ \ (c)\ \ x\&(x \vee y) = x;\ \ (d)\ x\&(\bar{x} \vee y) = x\&y.$$

После выполнения этих преобразований (а также преобразований, сводящихся к ним с использованием коммутативности 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 $A = x$, либо $A = x\&K'$, 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, чем исходная.

Остается оценить число шагов, достаточное для осуществления соответствующих преобразований. Каждое из них сводится к нахождению кода буквы x или $\bar{x}$, т.е. подслова длины $\lceil \log n \rceil + 1$, 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 $O(L \log n)$ шагов на обычной (детерминированной) машине Тьюринга, а код КНФ K в код КНФ K' можно преобразовать не более чем за $O(L^2 \log n) \leq O(L^3)$ шагов.

Случай, когда однобуквенные сомножители отсутствуют. Пусть КНФ K не содержит однобуквенных сомножителей. Тогда переход к 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 — некоторая буква, отличная от x и $\bar{x}$, входящих в K, K_1 представляет собой конъюнкцию всех дизъюнкций вида $(x \vee y)$, входящих в K, а K_2 представляет собой конъюнкцию всех остальных дизъюнкций, входящих в K. Таким образом, КНФ K_0 представима в виде $K_0 = \bigwedge_{1 \leq i \leq k}(\bar{x} \vee y_i)$, а K_1 представима в виде $K_1 = \bigwedge_{1 \leq j \leq m}(\bar{x} \vee z_j)$. Заметим, что

$$K_0 K_1 K_2 = (\bar{x} \vee y_1 ... y_k)(x \vee z_1 ... z_m) K_2. \tag{14}$$

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 правая часть (14) выполнима тогда и только тогда, когда выполнима формула

$$(y_1 ... y_k \vee z_1 ... z_m) K_2 = \bigwedge_{1 \leq i \leq k} \bigwedge_{1 \leq j \leq m} (y_i \vee z_j) K_2 = K'. \tag{15}$$

КНФ K' не содержит x и $\bar{x}$, а число букв в ней не больше L^2, где L — число букв в $W(K)$. Нетрудно построить машину Тьюринга, преобразующую

$W(K)$ в $W(K')$ за $O(L^3)$ шагов. КНФ K', возможно, содержит скобки вида $(y \vee y)$ и $(y \vee \bar{y})$, а также скобки, отличающиеся лишь порядком слагаемых. Удаление этих вхождений можно осуществить за число шагов, не превосходящее $O(L_1^2)$, где L_1 длина $W(K')$, а, значит, не более чем за $O(L^4)$ шагов, где L — число букв в $W(K)$.

Поскольку число переходов не превосходит $n - 2 \leq L$, то для преобразования исходной КНФ в формулу, зависящую не более, чем от одной переменной, достаточно $O(L^5)$ шагов.

В случае, когда КНФ K зависит не более, чем от одной переменной, возможны следующие случаи: $K = x$, $K = \bar{x}$, $K = x \vee \bar{x}$, $K = x\&\bar{x}$. 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 5.3 *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 $C = y_1 \vee ... \vee y_m$ — 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 $D = (y_1 \vee y_2 \vee u)(y_3 \vee ... \vee y_m \vee \bar{u})$. Покажем, что КНФ K выполнима тогда и только тогда, когда КНФ $D\&K_1$ выполнима.

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

Пусть теперь $\widetilde{\beta} = (\alpha_1, ..., \alpha_n, \beta)$ — набор, обращающий КНФ $D\&K_1$ в единицу. Пусть сначала $\beta = 0$. Тогда $g(\widetilde{\alpha}) = 1$ и $D\&K_1(\widetilde{\alpha}) = 1$, а, значит, $K(\widetilde{\alpha}) = 1$. Если же $\beta = 1$, то $h(\widetilde{\alpha}) = 1$ и $D\&K_1(\widetilde{\alpha}) = 1$.

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 $m_1, ..., m_k$ — числа букв в скобках КНФ K и $m_i > 3$. Тогда достаточно добавить аналогичным способом не более $2(m_1 + ... + m_k - 3k)$ букв с тем, чтобы получить 3-КНФ K^*, выполнимую тогда и только тогда, когда КНФ K выполнима.

Полиномиальность преобразования очевидна. □

Упражнение. Доказать NP-полноту задачи 4-ВЫП.

Задача ТАВТОЛОГИЯ определяется следующим образом.
ВХОД: ДНФ $D = D(x_1, \ldots, x_n)$.
СВОЙСТВО: $D(\alpha_1, \ldots, \alpha_n) = 1$ для всякого набора $(\alpha_1, \ldots, \alpha_n)$.

Упражнение. Доказать NP-полноту задачи ТАВТОЛОГИЯ.

Список литературы

[1] Кибернетический Сборник No 12 (Нов. серия), М. МИР, 1975, С. 5–38.

[2] А. Ахо, Д. Хопкрофт, Д. Ульман// Построение и анализ вычислительных алгоритмов, М., Мир, — 1979. — С. 420–428.

6 Некоторые NP-полные задачи

В этом параграфе расширяется список NP-полных задач. Доказывается NP-полнота задач 0-1 ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ, КЛИКА, ВЕРШИННОЕ ПОКРЫТИЕ, ПОКРЫТИЕ МНОЖЕСТВ, РАСКРАСКА. Доказательство NP-полноты очередной задачи проводится путем сведения к ней одной из уже известных NP-полных задач. Сведение 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. Ниже (см. рис. 4) дана схема сведения задач.

Рис.4

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

ВХОД: Матрица $A = (a_{i,j})$, размера $p \times n$, и целочисленный вектор $\mathbf{b} = (b_1, ..., b_p)$.

СВОЙСТВО: Существует 0-1-вектор $\mathbf{x} = (x_1, ..., x_n)$ такой, что

$$A\mathbf{x}^{\mathbf{T}} \geq \mathbf{b}^{\mathbf{T}}. \tag{16}$$

Теорема 6.1 *ВЫП* $\prec$ *0-1 ЦЛП.*

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

$$a_{ij} = \begin{cases} 1, & \text{если} \quad x_i \in C_j, \\ -1, & \text{если} \quad \bar{x}_i \in C_j, \\ 0, & \text{иначе} \end{cases}$$

и

$$b_i = 1-\text{ число отрицаний переменных в } C_i.$$

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 $\mathbf{x} = (x_1, ..., x_n)$, удовлетворяющий (16), существует тогда и только тогда, когда КНФ K выполнима.

Достаточность. Пусть K выполнима. Тогда существует набор $\tilde{\alpha} = (\alpha_1, ..., \alpha_n)$ значений переменных такой, что $K(\tilde{\alpha}) = 1$. Обозначим через A^i строку $(a_{i1}, \ldots, a_{in})$ матрицы A. Покажем, что (16) выполнено при $\mathbf{x} = \tilde{\alpha}$. Для этого убедимся, что для всякого $i = 1, \ldots, p$ выполнено

$$(A^i, \mathbf{x}) \geq b_i = 1 - \text{ число отрицаний переменных в } C_i. \qquad (17)$$

В самом деле, скобка C_i обращается на наборе $\tilde{\alpha}$ в 1. Это означает, что либо существует переменная из C_i, обращающаяся в 1, либо в C_i существует отрицание некоторой переменной, обращающейся в 0. В первом случае минимальное значение скалярного произведения $(A^i, \mathbf{x})$ достигается в случае, когда все координаты x_j при коэффициентах a_{ij}, равных 1, за исключением одного, обращаются в 0, а все координаты x_j при коэффициентах a_{ij}, равных -1, обращаются в 1. Во втором случае минимальное значение скалярного произведения $(A^i, \mathbf{x})$ достигается в случае, когда все координаты x_j при коэффициентах a_{ij}, равных 1, обращаются в 0, а все координаты x_j при коэффициентах a_{ij}, равных -1, за исключением одного, обращаются в 1. В обоих случаях (17) удовлетворяется.

Необходимость. Пусть существует двоичный вектор $\mathbf{x} = (x_1, ..., x_n)$, удовлетворяющий (16). Покажем, что $K(\mathbf{x}) = 1$, а, значит, КНФ K выполнима. Из (16) следует, что (17) выполнено для всякого $i = 1, \ldots, p$. Отсюда вытекает, что все скобки КНФ K содержат хотя бы одну букву, обращающуюся в 1 на наборе $\mathbf{x} = (x_1, ..., x_n)$. □

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 $\mathbf{b} = (1, -1)$. Решением неравенства (16) являются векторы $(0, 1, \alpha)$, $(1, 0, \beta)$ и $(1, 1, 1)$, где $\alpha, \beta \in \{0, 1\}$. 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-*клику*).

Теорема 6.2 *ВЫП* $\prec$ *КЛИКА.*

Доказательство.
Пусть КНФ $K = C_1 \& ... \& C_q$ является конъюнкцией некоторых q скобок, где $C_i = (y_{i1} \vee ... \vee y_{ik_i})$, а y_{ij} — некоторая буква. Положим

$$V = \{< y, i >: \ y \text{ есть буква из } C_i, 1 \leq i \leq q\};$$

$$E = \{\{< y, i >, < z, j >\} : \ i \neq j, \ y \neq \bar{z}\};$$

$$k = q.$$

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

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

Достаточность. Пусть K выполнима. Тогда существует набор $\tilde{\alpha} = (\alpha_1, ..., \alpha_n)$ значений переменных такой, что $K(\tilde{\alpha}) = 1$. Каждая скобка обращается на этом наборе в 1. Следовательно, всякая скобка C_i содержит хотя бы одну букву, принимающую значение 1. Пусть для C_i такой буквой будет y_i. Убедимся в том, что множество вершин $\{< y_i, i >\}$, $i = 1, ..., q$, порождает полный подграф в G. Если не так, то найдутся такие i и j, что $i \neq j$ и вершины $< y_i, i >$ и $< y_j, j >$ не смежны в графе G. Тогда $y_i = \bar{y}_j$. Но это невозможно в силу выбора букв y_i.

Необходимость. Пусть G содержит q-клику. Вторые компоненты вершин, образующих клику, попарно различны, ибо вершины с равными вторыми компонентами не смежны в графе G. Следовательно, вершины клики взаимно однозначно соответствуют скобкам КНФ K. Пусть вершины клики имеют вид $< y_i, i >$, $i = 1, ..., q$. Обозначим через S (через $\bar{S}$) множество тех букв y_i, которые являются переменными (соответственно, отрицаниями переменных). Ясно, что $S \cap \bar{S} = \emptyset$, ибо в противном случае некоторые вершины вида $< y_i >$ и $< y_j, j >$, такие, что $y_i = \bar{y}_j$, были смежны в графе G. Если положить все переменные из S равными 1, а переменные из $\bar{S}$ равными 0, то каждая скобка C_i обратится в 1. Значит, КНФ K выполнима. $\square$

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 рис. 5, и число 2.

Рис.5

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 6.3 *КЛИКА* $\prec$ *ВЕРШИННОЕ ПОКРЫТИЕ.*

Доказательство.
Отображение входов имеет вид:

$$G' = (V', E') \text{ есть дополнение графа } G = (V, E).$$

$$l = |V| - k.$$

Заметим, что множество $A \subseteq V$ 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. □

Рис.6

Пример. Граф G с множеством вершин $\{1, 2, 3, 4\}$ (см. рис. 6) содержит клику $\{1, 2, 3\}$. В графе $\overline{G}$ дополнение этого множества покрывает все ребра.

Задача ПОКРЫТИЕ МНОЖЕСТВ:

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

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

Теорема 6.4 *ВЕРШИННОЕ ПОКРЫТИЕ $\prec$ ПОКРЫТИЕ МНОЖЕСТВ.*

Доказательство.
Пусть задан вход задачи ВЕРШИННОЕ ПОКРЫТИЕ: граф $G' = (V', E')$ и число l. Положим

$S = E', \quad S_j = \{< u, v_j > \in E' : u \in V'\}, \quad$ и $\quad h = l.$

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

Задача РАСКРАСКА:

ВХОД: Граф $G = (V, E)$, и число k.

СВОЙСТВО: Существует функция $\varphi : V \to Z_k$ такая, что $\varphi(u) \neq \varphi(v)$ для всех $(u, v) \in E$.

Теорема 6.5 3-*ВЫПОЛНИМОСТЬ $\prec$ РАСКРАСКА.*

Доказательство.
Пусть формула K представляет собой 3-КНФ с n переменными и t сомножителями (скобками). Покажем как построить за время, ограниченное полиномом от $\max(n, t)$, граф $G = (V, E)$ с $3n + t$ вершинами, который можно раскрасить в $n + 1$ цветов тогда и только тогда, когда КНФ K выполнима.

Пусть $x_1, x_2, \ldots, x_n$ и $C_1, C_2, \ldots, C_t$ — соответственно переменные и сомножители КНФ K. Пусть $v_1, v_2, \ldots, v_n$ — новые символы. Без потери общности будем считать, что $n \geq 4$, поскольку любую КНФ, число различных переменных которой не превосходит 3, можно проверить на выполнимость за время, линейно зависящее от ее длины, не прибегая к раскраске.

Вершины графа G таковы:

1. $x_i, \bar{x}_i, v_i$ для $1 \leq i \leq n$,
2. C_i для $1 \leq i \leq t$.

Ребрами графа G являются
1. все (v_i, v_j), для которых $i \neq j$,
2. все (v_i, x_j) и $(v_i, \bar{x}_j)$, для которых $i \neq j$,
3. $(x_i, \bar{x}_i)$ для $1 \leq i \leq n$,
4. (x_i, C_j), если x_i не входит в C_j, и $(\bar{x}_i, C_j)$, если $\bar{x}_i$ не входит в C_j.

Вершины $v_1, v_2, \ldots, v_n$ образуют полный граф с n вершинами, так что для их раскраски требуется n различных цветов. Каждая из вершин x_j и $\bar{x}_j$ соединена с каждой вершиной v_j, $i \neq j$ и, значит, x_j и $\bar{x}_j$ не могут быть того же цвета, что и v_i, если $i \neq j$. Так как вершины x_j и $\bar{x}_j$ смежны, то они не могут быть одинакогого цвета, и потому граф G можно раскрасить в $n+1$ цветов только тогда, когда одна из вершин x_j и $\bar{x}_j$ имеет тот же цвет, что и v_j , а другая имет новый цвет, который мы назовем *специальным.*

Пусть той из вершин x_j и $\bar{x}_j$, которая раскрашена в специальный цвет. Рассмотрим цвет, приписанный вершинам C_j. Вершина C_J смежна по крайней мере с $2n-3$ из $2n$ вершин $x_1, x_2, \ldots, x_n, \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$. Так как мы предположили, что $n \geq 4$, то для каждого j найдется такое i, что вершина C_j смежна как с x_i, так и с $\bar{x}_i$. Поскольку одна из вершин x_i или $\bar{x}_i$ раскрашена в специальный цвет, то C_j не может быть раскрашена в специальный цвет. Если скобка C_j содержит такой символ y, что вершине $\bar{y}$ приписан специальный цвет, то вершина C_j не смежна ни с какой вершиной, раскрашенной также, как y, и, значит, ей можно приписать тот же цвет, что и у вершины y. В противном случае нужен новый цвет.

Таким образом, все вершины C_i можно раскрасить без дополнительных цветов тогда и только тогда, когда символам можно так приписать специальный цвет, чтобы каждый сомножитель содержал такой символ y, что символу $\bar{y}$ приписан специальный цвет, т.е. тогда и только тогда, когда переменным можно так присвоить значения, что бы в каждом сомножителе оказался y со значением 1 ($\bar{y}$ со значением 0), т.е. тогда и только тогда, когда КНФ C выполнима. □

Пример. Пусть $K = (x_1 \vee x_2)(\bar{x}_1 \vee x_3)$. Граф, соответствующий данному входу задачи 3-ВЫП, показан на рис. 7. Указанная на нем раскраска в цвета A, B, C и дополнительный цвет S соответствует набору (010), обращающему КНФ K в единицу.

Рис.7

Список литературы

[1] Кибернетический Сборник No 12 (Нов. серия), М. МИР, 1975, С. 5–10.

[2] А. Ахо, Д. Хопкрофт, Д. Ульман// Построение и анализ вычислительных алгоритмов, М., Мир, — 1979. — С. 420–428.

7 Теорема Сэвиджа

В этом параграфе устанавливается связь между сложностью булевых функций и временно́й сложностью машинных вычислений. Наличие такой связи может показаться неожиданным, поскольку схемная сложность ассоциируется скорее с описательной сложностью, нежели со сложностью вычислений. Тем не менее, как мы увидим, сложность схемы из функциональных элементов хорошо мажорирует время машинных вычислений. Это и утверждает теорема Дж. Сэвиджа [1]. Доступное для отечественного читателя изложение (которому мы здесь в основном следуем) можно найти также в [2].

Рассматриваются обычные (детерминированные) машины Тьюринга с односторонней бесконечной вправо лентой, алфавитом ленты $A = \{a_1, \ldots, a_m\}$ и алфавитом состояний $Q = \{q_0, \ldots q_k\}$. *Начальное* состояние обозначается через q_0, а *заключительное* — через q_k. Один из символов ленты называется *пустым* и обозначается через Λ. Он обозначает отсутствие значащего символа в ячейке ленты. Другие понятия, касающиеся машин Тьюринга, даны в параграфе 4. В начальный момент на ленте записано исходное слово $x_1, x_2, \ldots, x_n$ и головка обозревает самый левый символ этого слова в состоянии q_0. Все остальные ячейки заполнены символом Λ.

Прежде чем перейти к непосредственному моделированию машины Тьюринга схемами, отметим ряд обстоятельств, затрудняющих сравнение. Прежде всего, машины Тьюринга допускают входные слова произвольной длины, в то время как схемы — только слова фиксированной длины. Далее, машины Тьюринга к некоторым входным словам не применимы, тогда как схема определена на каждом входном слове. Время работы машины Тьюринга, вообще говоря, не ограничено никакой общерекурсивной функцией, в то время как минимальные схемы имеют ограниченную сложность.

Мы будем применять для моделирования обобщенные схемы из функциональных элементов, у которых на входах и выходах элементов — символы алфавитов A и $\widetilde{Q}$, где A — ленточный алфавит, $\widetilde{Q} = Q \cup \{\tilde{q}\}$, Q — алфавит состояний, $\tilde{q}$ — новый символ, который называется *холостым* состоянием.

Ясно, что каждый такой элемент можно моделировать обычной (двоичной) схемой константной сложности после предварительного кодирования букв алфавитов A и $\widetilde{Q}$ конечными двоичными последовательностями. Поэтому переход от обобщенных схем к обычным связан с увеличением сложности лишь в константу раз. Итак, пусть машина Тьюринга M работает на словах длины n не более T тактов.

Рис.8

Построим обобщенную схему, которая моделирует работу M на словах длины n. Схема строится из преобразующих элементов U и фильтрующих элементов Φ. Элемент U имеет два входа и четыре выхода (см. рис. 8).

На левый вход подаются символы $a_i \in A$, на правый $q_j \in \widetilde{Q}$. Если $q_j = \tilde{q}$, то элемент U производит тождественное преобразование, т.е.

$$a'_i = a'_l, \quad q_j^R = q_j^L = q_j^S = \tilde{q}.$$

Если $q_j \neq \tilde{q}$, то в системе команд машины M отыскиваем команду с левой частью $a_i q_j$. Пусть ее правая часть есть $a'_i q'_j L$. Тогда на выходе элемента U

$$a'_i = a'_l, \quad q_j^L = q'_j, \quad q_j^R = q_j^S = \tilde{q}.$$

Если символ движения головки L заменить на S или R, то, соответственно, будет $q_j^S = q'_j$ или $q_j^R = q'_j$, а на других q-выходах $\tilde{q}$.

На входах и выходе элемента Φ возникают только символы алфавита $\widetilde{Q}$. Если на одном из входов q^1, q^2, q^3 появляется символ, отличный от $\tilde{q}$, то он проходит на выход (случай, когда несколько входов отличны от $\tilde{q}$, невомозжен). Если на всех входах появляется $\tilde{q}$, то выход равен $\tilde{q}$.

Заметим, что если время работы МТ над словом $x_1 \dots x_n$ не превосходит T, то головка может уйти вправо от начального положения не далее, чем на T ячеек. Поэтому достаточно держать в поле зрения зону ленты из T ячеек, которые мы будем нумеровать числами от 1 до T. Схема, которую мы построим, имеет прямоугольный вид. У нее T (двухярусных) строк и T столбцов. При этом i-ая строка ($i = 1, 2, \dots, T$) выходами своих T элементов представляет i-ую конфигурацию машины M, а именно, элемент U j-го

$(1 \leq j \leq T)$ столбца — символ, содержащийся в j-ой ячейке, а элемент Φ — состояние машины, обозревающей j-ю ячейку. При всяком i в точности для одного j состояние отлично от $\tilde{q}$ (а именно, для той ячейки, которая действительно обозревается головкой в i-й конфигурации). Для всех остальных ячеек элемент Φ выдает значение $\tilde{q}$ (холостое состояние). На пересечении i-й стороки и j-ого столбца в схеме один элемент U и один элемент Φ. Мы будем изображать их один под другим, тем самым каждая строка будет двухярусной. Порядок соединения элементов показан на рис.9.

Рис.9

Для наглядности на том же рисунке сверху показан фрагмент текущей конфигурации, а на входах элементов — соответствующие значения схемы в предположении, что команда машины M имеет вид $a_i q_j \to a'_i q'_j L$.

Теорема 7.1 (Дж.Сэвидж) *Пусть машина Тьюринга M работает на словах длины n не более $T_M(n)$ тактов. Тогда ее можно моделировать схемой из функциональных элементов сложности $O(T_M^2(n))$.*

Доказательство. Построенная нами схема из обобщенных элементов моделирует работу машины M. Фактически она воспроизводит последовательность конфигураций при работе машины над входным словом. Из построения ясно, что схема содержит $2T^2$ обобщенных элементов. После замены каждого такого элемента схемой из двоичных элементов сложность возрастает только в константу раз. □

Примечание. Легко видеть, что в построенной схеме много "лишних" элементов. Это, например, элементы правого верхнего угла схемы, где длительное время состояние остается холостым. И в оставшейся части схемы существенны только элементы в окрестности нехолостого значения состояния. Это предоставляет большие возможности для упрощения схемы. К.П.Шнорр [3] понизил верхнюю оценку до $O(T_M(n)\log S_M(n)+\|M\|)$, где $T_M(n)$ и $S_M(n)$ соответственно число тактов и число ячеек, достаточное для обработки слова длины n машиной Тьюринга, а $\|M\|$ — число команд машины M.

Список литературы

[1] J.E.Savage, Computational work and time on finite machines, J. Ass. Comp. Mach., 1972, v. 19, p. 660-674.

[2] Р.Г.Нигматуллин, Сложность булевых функций, Издательство Казанского университета, 1983, 208 с.

[3] C.P.Schnorr, The network complexity and the Turing machine complexity of finite functions. Acta Informatica, 1976, v. 7, p. 95-107.

Содержание