Московский государственный университет имени М. В. Ломоносова

Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

ДИПЛОМНАЯ РАБОТА СТУДЕНТА 517 ГРУППЫ

«Развитие методов синтеза дизъюнктивной нормальной формы для функций с ограниченным числом нулей и их применение к задачам распознавания»

Выполнила:

студентка 5 курса 517 группы

Морозова Дарья Юрьевна

Научный руководитель:

Заведующий кафедрой

Математических Методов

Прогнозирования, академик РАН

Журавлёв Юрий Иванович

Заведующий кафедрой

Математических Методов

Прогнозирования, академик РАН ____________________ Ю. И. Журавлёв

Москва, 2014

Содержание

1 Глава 1 **3**

2 Глава 2 **5**

3 Глава 3 **7**

3.1 Простейший алгоритм распознавания 8

3.2 Осторожный простейший алгоритм распознавания 8

3.3 Второй алгоритм распознавания 9

4 Глава 4 **9**

5 Глава 5 **11**

5.1 Простейший алгоритм разбиения множества нулей на полосы 11

5.2 Второй алгоритм разбиения множества нулей на полосы 14

6 Приложения **17**

Введение

Построение дизъюнктивных нормальных форм (д. н. ф.) для функций алгебры логики (булевых) функций, если при этом требуется построить д. н. ф. минимальной сложности (по числу входящих в неё букв - минимальной д. н. ф., или элементарных конъюнкций - кратчайшей д. н. ф.), при достаточно большом числе переменных является задачей практически неразрешимой даже для суперсовременных быстродействующих компьютеров.

Это при весьма слабых ограничениях на класс алгоритмов было доказано Ю. И. Журавлёвым [Оценки сложности алгоритмов построения минимальных дизъюнктивных форм для функций алгебры логики (булевых функций). Дискретный Анализ. Сб. Трудов Института математики Сиб. отд. АНСССР, вып. 3, 1964 г. [1]]. Поэтому в дальнейшем алгоритмы минимизации эффективно применялись, как правило, лишь для достаточно узких классов функций (монотонные, линейные функции и т. п.).

В дальнейшем мы будем использовать понятия обозначения из работы [4] - элементарная конъюнкция K, интервал элементарной конъюнкции N_K, сокращённая, минимальная, тупиковая д. н. ф. и т. п.

В настоящей работе мы рассмотрим весьма узкий подкласс булевых функций (функции с малым числом нулей). Эти функции после извесной работы Ю. И. Журавлёва и А. Ю. Когана [2] привлекли внимание различных авторов (А. Ю. Коган, А. Г. Дьяконов, Ю. В. Максимов) [10, 7, 8, 9, 5], причём А. Г. Дьяконов применил полученные результаты к решению задач распознавания с бинарной обучающей информацией.

Рассматривался случай задачи распознавания с двумя непересекающимися классами K_1, K_2; объекты распознавания задавались бинарными признаками (со значениями 0,1), число объектов обучения, как правило, было относительно невелико.

В данной работе рассматривается несколько другой подход к реализации в классе д. н. ф. булевых функций с малым числом нулей и способ применения полученных результатов к задачам распознавания бинарных объектов.

В работе используются общепринятые упрощения в теории д. н. ф., при этом существенно используются: $x_i^{\sigma} \cdot x_i^{\bar{\sigma}} = 0$, $K \vee K \cdot K' = K$ (правило поглощения), а также широко используемый в теории д. н. ф. метод построения сокращённой д. н.

ф., получивший в литературе название Блейка-Квайна (хотя значительно ранее он был описан русским учёным Порецким, а позднее использовался Квайном без ссылок на предыдущих авторов).

1 Глава 1

Пусть булева функция задана перечислением наборов

$\hat{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{in}), i = 1, 2, \dots, k,$

на которых она обращается в 0. Известно, что f может быть задана конъюнктивной совершенной нормальной формой (1):

$f(x_1, \dots, x_n) = \prod\limits_{i=1}^{k} (x_1^{\bar{\alpha}_{i1}} \vee \dots \vee x_n^{\bar{\alpha}_{in}}).$

В теории нетрудно провести умножение K членов, провести стандартные упрощения типа $x_t^\sigma \cdot x_t^\sigma = x_t^\sigma$, $K \vee K \cdot K' = K$ и т. п. и преобразовать (1) в дизъюнктивную нормальную форму $\mathcal{D}$, которую в дальнейшем можно перевести в сокращённую д. н. ф. $\mathcal{D}_f$, а затем, используя, например, критерий поглощения [4], построить реализующую тупиковую д. н. ф. В действительности, даже при относительно небольших K такой перевод требует большого машинного времени, что не под силу даже современным сверхбыстродействующим компьютерам. Поэтому задача построения д. н. ф. даже при небольших значениях K и больших n является весьма непростой.

В 1956 году С. В. Яблонский на одном из семинаров по дискретной математике обратил внимание на следующее: если рассмотреть функцию $f(x_1, \dots, x_n)$, которая обращается в 0 только на наборах $\hat{\alpha}_1 = (1, 1, \dots, 1)$ и $\hat{\alpha}_0 = (0, 0, \dots, 0)$, то совершенная к. н. ф. такой функции

(2) $f = (x_1 \vee \dots \vee x_n) \cdot (\bar{x}_1 \dots \bar{x}_n)$

преобразуется в минимальную д. н. ф.

$\mathcal{D}_f = x_1 \cdot \bar{x}_2 \vee x_2 \cdot \bar{x}_3 \vee \dots \vee x_{n-1} \cdot \bar{x}_n \vee x_n \cdot \bar{x}_1.$

Д. н. ф. состоит из n конъюнкций, является минимальной д. н. ф. и имеет сложность $2n$. Такая же сложность и у конъюнктивной д. н. ф. этой функции. Пример Яблонского (как отметил Ю. И. Журавлёв) никогда не публиковался, но послужил отправной точкой для построения относительно простых д. н. ф. для функций, имеющих небольшое число нулей.

Опишем подробнее алгоритм Ю. И. Журавлёва и А. Ю. Когана [3], так как в силу простоты реализации он будет в дальнейшем являться отправной точкой при построении алгоритмов распознавания для задач распознавания с бинарной информацией с двумя непересекающимися классами.

Пусть задана таблица нулей (3) функции $f(x_1, \ldots, x_n)$:

$\hat{\alpha}_1 = (\alpha_{11}, \alpha_{12}, \ldots, \alpha_{1n})$

$\hat{\alpha}_2 = (\alpha_{21}, \alpha_{22}, \ldots, \alpha_{2n})$

$\ldots\ldots\ldots\ldots\ldots\ldots\ldots$

$\hat{\alpha}_k = (\alpha_{k1}, \alpha_{k2}, \ldots, \alpha_{kn})$

Если в таблице имеются нулевые и единичные столбцы α_{tr}, $r = 1, \ldots, n$, то их можно удалить из таблицы (ввести в формулу, которая получится при перемножении слагаемых к. н. ф., так как они войдут в результирующую д. н. ф. как одно из слагаемых x_r или $\bar{x}_r$).

Выполним над таблицей (3) следующие преобразования:

1. $x_r \to x_r^*$, если $\alpha_{1r} = 0$, $x_r \to \bar{x}_r^*$, если $\alpha_{1r} = 1$. Получим таблицу, в которой нет нулевых столбцов, а строка $\bar{\alpha}_1$ переходит в нулевую строку (0, 0, ..., 0);

2. переставим одинаковые столбцы так, чтобы они имели последовательные номера $x_r^* \to y_i$, $\bar{x}_r^* \to \bar{y}_i$, $i = 1, \ldots, n$, т. е. одинаковые столбцы образуют блоки.

Выполняемые преобразования не меняют структуру функции - сокращённая д. н. ф. переходит в сокращённую, тупиковые и минимальные д. н. ф. - в соответствующие тупиковые и минимальные; поэтому обратными преобразованиеми получается исходная таблица (3).

В результате проделанных преобразований таблица (3) приобретёт следующий вид (4):

y_1	$\ldots$	y_l	$\ldots$	y_p	$\ldots$	y_q	$\ldots$	y_s	$\ldots$	y_n
0	$\ldots$	0	$\ldots$	0	$\ldots$	0	$\ldots$	0	$\ldots$	0
α'_{11}	$\ldots$	α'_{11}	$\ldots$	α'_{p1}	$\ldots$	α'_{p1}	$\ldots$	α'_{s1}	$\ldots$	α'_{s1}
α'_{12}	$\ldots$	α'_{12}	$\ldots$	α'_{p2}	$\ldots$	α'_{p2}	$\ldots$	α'_{s2}	$\ldots$	α'_{s2}
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
α'_{1k-1}	$\ldots$	α'_{1k-1}	$\ldots$	α'_{pk-1}	$\ldots$	α'_{pk-1}	$\ldots$	α'_{sk-1}	$\ldots$	α'_{sk-1}

Построим функцию φ: из каждого блока отбираем первый столбец. Получаем (5):

y_1	$\dots$	y_p	$\dots$	y_s
0	$\dots$	0	$\dots$	0
α'_{11}	$\dots$	α'_{p1}	$\dots$	α'_{s1}
α'_{12}	$\dots$	α'_{p2}	$\dots$	α'_{s2}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
α'_{1k-1}	$\dots$	α'_{pk-1}	$\dots$	α'_{sk-1}

На всех k наборах таблицы (5) функция $\varphi = 0$, на остальных 1. Пусть $\tilde{\mathcal{D}}_f$ - некоторая д. н. ф., реализующая функцию φ.

Тогда исходная функция f представима в виде д. н. ф. $\mathcal{D}_f$:

$\mathcal{D}_f = y_1 \cdot \bar{y}_2 \vee y_2 \cdot \bar{y}_3 \vee \dots \vee y_{l-1} \cdot \bar{y}_l \vee y_l \cdot \bar{y}_1 \vee \dots \vee y_p \cdot \bar{y}_{p+1} \vee \dots \vee y_{q-1} \cdot \bar{y}_q \vee y_q \cdot \bar{y}_p \vee \dots \vee y_s \cdot \bar{y}_{s+1} \vee \dots \vee y_n \cdot \bar{y}_s \vee \tilde{\mathcal{D}}_f$.

Число конъюнкций в $\mathcal{D}_f$ равна $n + |\tilde{\mathcal{D}}_f|$, где $|\tilde{\mathcal{D}}_f|$ - число конъюнкций в $\tilde{\mathcal{D}}_f$. Заметим, что число нулевых значений φ не превосходит k, а число переменных не превосходит числа блоков в (4). Последнее, очевидно, не превосходит $2^{k-1} - 2$. Так, для исходной функции с четырьмя нулями - не превосходит 7, а с пятью нулями - 15. Построение д. н. ф. для функций с 4, 5, 6 нулями легко реализуется на современных компьютерах.

Полные функции подробно исследовались в работе А. Г. Дьяконова [7] и дипломной работе А. Н. Кириллова (Москва, факультет ВМК, 2013).

2 Глава 2

Рассматривается задача распознавания с двумя непересекающимися классами с бинарной информацией.

В данном случае объекты задаются n бинарными признаками, принимающими значения $0, 1$. Классы, как обычно, обозначаем через K_1, K_2. Очевидно, $K_1 \cap K_2 = \varnothing$.

Всевозможные 2^n объектов запишем в виде таблицы, в которой первые q объектов принадлежат классу K_1, последние l - классу K_2, для остальных «классовая» принадлежность неизвестна.

S_1	a_{11}	$\dots$	a_{1j}	$\dots$	a_{1n}	
S_i	a_{i1}	$\dots$	a_{ij}	$\dots$	a_{in}	K_1
S_q	a_{q1}	$\dots$	a_{qj}	$\dots$	a_{qn}	
S'	c_1	$\dots$	c_j	$\dots$	c_n	K_1 или K_2?
$\hat{S}_1$	b_{11}	$\dots$	b_{1j}	$\dots$	b_{1n}	
$\hat{S}_u$	b_{u1}	$\dots$	b_{uj}	$\dots$	b_{un}	K_2
$\hat{S}_l$	b_{l1}	$\dots$	b_{lj}	$\dots$	b_{ln}	

Объект S', как и любой из не вошедших в множество $K_1 \cap K_2$ объектов, может быть предъявлен к распознаванию.

Предположим, что нам удалось построить д. н. ф. $\mathcal{D}_1$, равную 0 только на наборах $S_1, \dots, S_q$. Обозначим множество всех 2^n наборов через M. Тогда на всех наборах из $M \backslash \{S_1, \dots, S_q\}$, очевидно, $\mathcal{D}_1 = 1$. Следовательно, для каждого S' найдётся хотя бы одна элементарная конъюнкция K, такая, что $K(S') = 1$. Возможно, что найдутся несколько таких конъюнкций. Предположим, что построены несколько д. н. ф. типа $\mathcal{D}_1$. Возможно также, что в этих д. н. ф. найдётся несколько конъюнкций, равных 1 на наборе S'.

Проведя аналогичное построение, только теперь рассматривая д. н. ф., равную 0 на множестве K_2, мы получим некоторое множество конъюнкций, равных 1 на S'. Возможно также построение совокупности д. н. ф.

Обозначим построенные совокупности через $\mathcal{D}_1$, $\mathcal{D}_2$.

Анализируя эти совокупности, можно отнести объект S' к одному из классов K_1 или K_2.

Последующая задача состоит в том, чтобы построить множества $\mathcal{D}_1$, $\mathcal{D}_2$ и найти алгоритм (или семейство алгоритмов), определяющий к какому из классов принадлежит объект S'.

3 Глава 3

Если число эталонных объектов в классах невелико, то можно построить д. н. ф. $\mathcal{D}(K_1)$, $\mathcal{D}(K_2)$, принимающие соответственно на наборах из классов K_1, K_2 значения 0, а на остальных: $2^n - |K_2|, 2^n - |K_1|$ - значения 1. Здесь через $|M|$ обозначено число объектов в множестве M. Так, если рассматривать метод, предложенный Ю. И. Журавлёвым и А. Ю. Коганом [3], число переменных функции φ не превосходит, соответственно, при $|K_1| = 3$, $|K_1| = 4$, $|K_1| = 5$, $|K_1| = 6$ значения 3, 7, 15, 31. Очевидно, либо вручную, либо с использованием современных компьютеров, нетрудно построить д. н. ф. $\mathcal{D}(K_1)$, $\mathcal{D}(K_2)$. При соответствующих построениях можно использовать также алгоритмы, предложенные А. Ю. Коганом [10] и А. Г. Дьяконовым [7, 8, 9].

Пример 1. Пусть эталоны класса K_2 в таблице обучения после проведения описанных выше преобразований $x_i \to x_j^{\sigma_{ij}}$ и образования блоков из одинаковых столбцов примут вид:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
1	1	1	0	0	0	1	1	1	0	0	0	1	1	0	0	1	1	1

В этом случае функция φ зависит от 7 переменных, и таблица её нулевых значений выглядит следующим образом:

z_1	z_2	z_3	z_4	z_5	z_6	z_7
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	1	1	0	0	1	1
1	0	1	0	1	0	1

Совершенная к. н. ф.:

$(z_1 \vee z_2 \vee z_3 \vee z_4 \vee z_5 \vee z_6 \vee z_7) \cdot (z_1 \vee z_2 \vee z_3 \vee \bar{z}_4 \vee \bar{z}_5 \vee \bar{z}_6 \vee \bar{z}_7) \cdot (z_1 \vee \bar{z}_2 \vee \bar{z}_3 \vee z_4 \vee z_5 \vee \bar{z}_6 \vee \bar{z}_7) \cdot (\bar{z}_1 \vee z_2 \vee \bar{z}_3 \vee z_4 \vee \bar{z}_5 \vee z_6 \vee \bar{z}_7)$

Её совершенная к. н. ф. легко переводится в д. н. ф. прямым умножением, после чего методом Блейка легко строится сокращённая д. н. ф., а затем из неё последовательным удалением элементарных конъюнкций - тупиковые д. н. ф.

Аналогично можно построить тупиковые д. н. ф. для класса K_2. Пусть в результате таких вычислений построены тупиковые д. н. ф. $\mathcal{D}_{11}, \mathcal{D}_{12}, \ldots, \mathcal{D}_{1k}$ принимающие значения 0 на K_2 и тупиковые д. н. ф. $\mathcal{D}_{21}, \mathcal{D}_{22}, \ldots, \mathcal{D}_{2r}$ с нулевыми значениями на K_1.

Приведённый пример позволяет построить семейство распознающих алгоритмов, если число эталонных объектов в таблице обучения невелико.

Д. н. ф. $\mathcal{D}_{1j}$ называется выделенной для S' по классу K_1, если $\mathcal{D}_{1j}(S') = 1$.

Аналогично выделяются д. н. ф. по классу K_2. Анализ выделенных д. н. ф. позволяет отнести S' к одному из классов K_1, K_2 или отказаться от распознавания.

В каждой из д. н. ф. выделяются элементарные конъюнкции K такие, что $K(S') = 1$ (напомним, что S' - распознаваемый объект). Совокупность таких конъюнкций из $\mathcal{D}_{1j}$, $j = 1, \ldots, k$, и $\mathcal{D}_{2j'}$, $j' = 1, \ldots, r$, обозначим соответственно через Q_1, Q_2.

3.1 Простейший алгоритм распознавания

$S' \in K_1$ если $|Q_1| > |Q_2|$, $S' \in K_2$, если $|Q_2| > |Q_1|$.

Здесь, как и далее, через $|M|$ обозначается число элементов в M.

3.2 Осторожный простейший алгоритм распознавания

Вводится целочисленный положительный параметр r^*.

$S' \in K_1$ если $|Q_1| > |Q_2| + r^*$, $S' \in K_2$, если $|Q_2| > |Q_1| + r^*$.

3.3 Второй алгоритм распознавания

Для каждой конъюнкции из Q_i, $i = 1, 2$, вычисляется число элементов $\hat{\alpha}$ в K_i, $i = 1, 2$, таких, что $K_i(\hat{\alpha}) = 1$.

Полученная величина называется весом $W(Q)$.

Подсчитываются суммы весов:

$W_1 = \sum\limits_{Q \in Q_1} W(Q)$, $W_2 = \sum\limits_{Q \in Q_2} W(Q)$.

Очевидно, что $W_1 = W_1(S')$, $W_2 = W_2(S')$.

Окончательное решение принимается по сравнению весов:

$S' \in K_1$, если $W_1 > W_2$, $S' \in K_2$, если $W_2 > W_1$.

Или с учётом параметра «осторожности» $r^* > 0$:

$S' \in K_1$, если $W_1 > W_2 + r^*$, $S' \in K_2$, если $W_2 > W_1 + r^*$.

Заметим, что в предложенном алгоритме основное решение принимается по числу элементарных конъюнкций K, для которых $K(S') = 1$. При этом необязательно рассматривать все тупиковые д. н. ф., число которых может оказаться настолько большим, что вычисление практически недоступно даже современным компьютерам.

4 Глава 4

При большем числе нулей булевой функции реализующая д. н. ф. становится неприемлемо большой. Следовательно, реальное применение методов распознавания, изложенных в предыдущей главе, становится невозможным. Однако, во многих случаях модифицирование и усложнение схемы алгоритмов, описанных ранее, могут быть применены.

Пусть число эталонных объектов в классах K_1, K_2 равно соответственно l и m, $l >> 6$, $m >> 6$.

В этом случае множество нулей, определяющих булевские функции для классов K_1, K_2, могут быть разрезаны на полосы.

Для каждой из полос следует применить один из методов, пригодных для малого числа нулей. Если множество разрезано на полосы $\Pi_1, \dots, \Pi_k$, то, построив д. н. ф.

$\mathcal{D}_1(\Pi_1), \ldots, \mathcal{D}_k(\Pi_k)$, можно записать д. н. ф. (6):

$$\mathcal{D} = \prod_{i=1}^{k} \mathcal{D}_i(\Pi_i).$$

Здесь возникают как минимум две проблемы:

1. полосы должны состоять из небольшого числа строк;

2. д. н. ф. $\mathcal{D}_i(\Pi_i)$ по возможности должны иметь небольшую сложность.

Указанные проблемы являются весьма трудными, их полное решение к настоящему времени не получены, но отдельные результаты удалось получить. Они будут изложены в следующей главе.

В данном разделе будут приведены некоторые алгоритмы, которые позволят в ряде случаев упрощать выполнение произведения (6).

В предыдущей главе было показано, что на результат распознавания влияют отдельные конъюнкции, а не д. н. ф. в целом. Поэтому в процессе умножения можно воспользоваться несколькими упрощающими соотношениями:

1. Пусть в одной из конъюнкций K д. н. ф. $\mathcal{D}_i(\Pi_i)$ содержится переменная x_u^{σ}. Тогда все конъюнкции K_u из других д. н. ф., содержащие сомножитель $x_u^{\bar{\sigma}}$, в произведении с K_u дают 0, и все члены, содержащие $K \cdot K_u$, равны 0 и могут быть удалены.

2. Если некоторая конъюнкция обращается в 0 на всех точках класса K_1, то она не представляет интерес для задачи распознавания и должна быть удалена.

3. Если $K = 0$ на распознаваемом объекте S', её следует удалить.

4. Необходимо выполнить упрощения типа $K \vee K \cdot K' = K$ (для чего используется алгоритм Блейка-Квайна):

   $x \cdot K \vee \bar{x} \cdot K' \to x \cdot K \vee \bar{x} \cdot K' \vee K \cdot K'$,

   $K \vee K \cdot K' = K$.

5. По окрестности первого порядка каждой конъюнкции в каждой д. н. ф. удаляются поглощаемые конъюнкции:

$$S_1(K, \mathcal{D}_f) = K \cup \bigcup_{i=1}^{l} K_i, \ \bigcup_{i=1}^{l} N_{K_i} \supseteq N_K, \ N_{K_i} \cap N_K \neq \varnothing, \text{ конъюнкция } K \text{ удаляется.}$$

Указанные преобразования позволяют существенно упростить процесс распознавания.

Кроме того, процесс умножения необязательно выполнять полностью. Пусть после частичного умножения получены д. н. ф.: $\mathcal{D}_1$, $\mathcal{D}_1$, ..., $\mathcal{D}_q$.

Вместо дальнейшего выполнения умножений $\prod\limits_{i=1}^{q} \mathcal{D}_i$ можно подсчитать определённые в предыдущей главе весовые коэффициенты для каждой из $\mathcal{D}_i$, $i = 1, \ldots, q$ отдельно, а затем, сложив их, применить решающие правила.

Описанный выше процесс можно выполнить не всегда. Но в тех случаях, когда это возможно, гарантировано эффективное построение распознающего алгоритма.

5 Глава 5

Заметим, что полосы должны состоять из небольшого числа строк и необязательно одинаковой длины. А. Г. Дьяконовым были построены д. н. ф. для полных функций с 5 и 6 нулями в [7]. Разрезать множество нулей будем на полосы длины 5 или 6, и тогда для каждой из полос можно воспользоваться формулой, выводимой алгоритмом А. Г. Дьяконова, осуществить их умножение и провести (в процессе умножения) приведённые выше упрощения.

Разбиение множества нулей на полосы можно выполнять различными способами. Далее будут предложены два алгоритма, разрезающие множество нулей на полосы любой длины.

5.1 Простейший алгоритм разбиения множества нулей на полосы

Сложностью множества строк назовём число различных столбцов в данном множестве.

Выполним следующие действия:

1. Выбираем в таблице нулевых наборов две произвольные строки.

2. Выполняем преобразования $x_{1j} \to \bar{x}^*_{1j}$ для всех x_{1j}, равных 0 (т. е. в первой строке).

3. Присоединяем к двум выбранным строкам третью строку так, чтобы получить множество минимальной сложности. Если таких строк несколько, выбираем произвольную.

4. Процесс можно повторить присоединением 4, 5, 6 строк.

5. Выполняем обратные преобразования. Процесс формирования первой полосы закончен.

Таким же образом можно формировать 2, 3, и т. д. полосы, удалив из исходного множества нулей строки, соответствующие уже построенным полосам.

Пример 2. Пусть дано множество нулей 15*10. Разрежем на 3 полосы длины 5.

Выберем первую и вторую строки. Выполним преобразования в первой строке.

Исходное множество нулей	После преобразования в первой строке
0101010101	0000000000
0010101010	0111111111
0001101101	0100111000
0000111100	0101101001
0001000100	0100010001
0010101010	0111111111
1010101010	1111111111
1101101101	1000111000
1110111011	1011101110
1111011110	1010001011
1101010101	1000000000
1111000011	1010010110
1110010010	1011000111
1110000011	1011010110
1100110011	1001100110

Присоединим к двум выбранным строкам третью строку так, чтобы получить множество минимальной сложности. Таких строк несколько, выберем произвольную (например, 6 строка из преобразованного множества).

0 0 0 0 0 0 0 0 0 0

0 1 1 1 1 1 1 1 1 1

0 1 1 1 1 1 1 1 1 1

Первый и второй столбцы задают сложность = 2. Повторим присоединением четвёртой, а затем и пятой строки (7 и 11 строки).

0 0 0 0 0 0 0 0 0 0

0 1 1 1 1 1 1 1 1 1

0 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 0 0 0 0 0 0 0 0 0

Сделаем обратные преобразования. Первая полоса готова. В неё вошли 1, 2, 6, 7, 11 строки из исходного множества нулей.

Рассмотрим оставшиеся строки из матрицы нулей. Построим вторую полосу. Выполним такие же преобразования.

Исходное оставшееся множество нулей	После преобразования в первой строке
0001101101	0000000000
0000111100	0001010001
0001000100	0000101001
1101101101	1100000000
1110111011	1111010110
1111011110	1110110011
1111000011	1110101110
1110010010	1111111111
1110000011	1111101110
1100110011	1101011110

К первой и второй строке присоединим третью, четвёртую, пятую строки последовательно.

0 0 0 0 0 0 0 0 0 0

0 0 0 1 0 1 0 0 0 1

1 1 1 0 1 0 1 1 1 0

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 0 1 1 1 0

Сделаем обратные преобразования. Вторая полоса готова. В неё вошли 3, 4, 12, 13, 14 строки из исходного множества нулей. В третью полосу вошли оставшиеся строки 5, 8, 9, 10, 15.

Первая полоса	Вторая полоса	Третья полоса
0101010101	0001101101	0001000100
0010101010	0000111100	1101101101
0010101010	1111000011	1110111011
1010101010	1110010010	1111011110
1101010101	1110000011	1100110011
Сложность = 2	Сложность = 3	Сложность = 6

5.2 Второй алгоритм разбиения множества нулей на полосы

В простейшем алгоритме, описанном ранее, строки для полосы выбирались последовательно по некоторому правилу. Размер полосы можно было изменить в процессе выполнения алгоритма (например, если число различных столбцов резко возрастало). Пусть размер полосы фиксирован и равен q. Тогда выбираем множество всех строк, образующих полосу, с минимальной сложностью.

Выполним следующие действия:

1. Строим таблицу сравнений из C_n^2 столбцов. Все столбцы таблицы нулевых значений попарно складываем по модулю два.

2. Из таблицы сравнений удаляем столбцы, у которых множество нулей меньше q.

3. Выписываем для каждого столбца таблицы сравнений номера строк, в которых стоят нули в этом столбце. Обозначим количество номеров строк за m_i, $i = 1, \ldots, N$, где $N \leq C_n^2$

4. Если $m_i > q$, $i = 1, \ldots, N$, то выписываем все наборы из q номеров строк, которые можно построить из полученного множества из m_i номеров строк. Таких наборов $C_{m_i}^q$.

5. Среди всех выписанных наборов искать те, которые повторились наибольшее число раз.

6. Найденные наборы будут являться множеством строк в полосе. Процесс формирования первой полосы закончен.

Таким же образом можно формировать 2, 3, и т. д. полосы, удалив из исходного множества нулей строки, соответствующие уже построенным полосам.

Пример 3. Пусть дано множество нулей 8*5. Разрежем на 2 полосы длины 4.

1 0 0 0 0

0 1 0 1 0

0 1 0 0 1

0 0 1 0 0

1 0 1 0 1

1 1 0 1 1

0 0 1 1 1

1 1 1 1 0

Построим таблицу сравнений из $C_5^2 = 10$ столбцов. Все столбцы таблицы нулевых значений попарно складываем по модулю два.

$1 \oplus 2$	$1 \oplus 3$	$1 \oplus 4$	$1 \oplus 5$	$2 \oplus 3$	$2 \oplus 4$	$2 \oplus 5$	$3 \oplus 4$	$3 \oplus 5$	$4 \oplus 5$
1	1	1	1	0	0	0	0	0	0
1	0	1	0	1	0	1	1	0	1
1	0	0	1	1	1	0	0	1	1
0	1	0	0	1	0	0	1	1	0
1	0	1	0	1	0	1	1	0	1
0	1	0	0	1	0	0	1	1	0
0	1	1	1	1	1	1	0	0	0
0	0	0	1	0	0	1	0	1	1

Из таблицы сравнений удалим столбцы, у которых множество нулей меньше 4. Для данного примера такой столбец $2 \oplus 3$.

Выпишем для каждого столбца таблицы сравнений номера строк, в которых стоят нули в этом столбце.

$1 \oplus 2 : \{4, 6, 7, 8\}$

$1 \oplus 3 : \{2, 3, 5, 8\}$

$1 \oplus 4 : \{3, 4, 6, 8\}$

$1 \oplus 5 : \{2, 4, 5, 6\}$

$2 \oplus 4 : \{1, 2, 4, 5, 6, 8\}$

$2 \oplus 5 : \{1, 3, 4, 6\}$

$3 \oplus 4 : \{1, 3, 7, 8\}$

$3 \oplus 5 : \{1, 2, 5, 7\}$

$4 \oplus 5 : \{1, 4, 6, 7\}$

Заметим, что в $2 \oplus 4$ количество номеров строк > 4. Выпишем всевозможные наборы из 4 номеров строк. Всего таких наборов $C_6^4 = 15$:

$\{1, 2, 5, 8\}$

$\{1, 2, 5, 6\}$

$\{1, 2, 6, 8\}$

$\{1, 2, 4, 5\}$

$\{1, 4, 5, 6\}$

$\{1, 5, 6, 8\}$

$\{1, 2, 4, 6\}$

$\{1, 2, 4, 8\}$

$\{1, 4, 6, 8\}$

$\{1, 4, 5, 8\}$

$\{2, 5, 6, 8\}$

$\{2, 4, 5, 6\}$

$\{2, 4, 5, 8\}$

$\{2, 4, 6, 8\}$

$\{4, 5, 6, 8\}$

Среди всех выписанных наборов найдём те, которые повторились наибольшее число раз. Заметим, что набор $2, 4, 5, 6$ встретился дважды, у столбца $1 \oplus 5$ и у столбца $2 \oplus 4$. Все остальные наборы не повторяются. Это означает, что в исходном множестве нулей для 2, 4, 5, 6 строки 1 и 5, 2 и 4 столбцы совпадают. То есть 4 строки и 3 различных столбца. Значит первая полоса состоит из строк под номерами 2, 4, 5, 6, а вторая - из оставшихся 1, 3, 7, 8.

Первая полоса	Вторая полоса
01010	10000
00100	01001
10101	00111
11011	11110
Сложность $= 3$	Сложность $= 4$

6 Приложения

А. Г. Дьяконов в [7] привёл тупиковые д. н. ф. небольшой сложности, реализующие полные функции при $k \in \{5, 6\}$.

Далее будет предложен другой алгоритм построения д. н. ф. для полных функций без нулевых строк и без нулевых столбцов при $k \in \{3, 5\}$.

Пусть дана булева функция, которая принимает значение 0, допустим, на q наборах, причём зависит от $2^q - 1$ переменных, все столбцы различны, и нет нулевого столбца.

Мы будем рассматривать только нулевые наборы.

α_1	...	α_t	...	α_k	...	α_r	...	α_{2^q-1}	f
α_1^1		α_t^1		1		α_r^1		α_n^1	0
...		...		...		...		...	...
α_1^q		α_t^q		1		α_r^q		α_n^q	0

Зададим преобразования для наборов данной булевой функции: $\alpha_i \to x_i$ при $i = 1, \ldots, 2^q - 1$ (Перегруппируем столбцы для удобства: сначала поставим столбцы с одной единицей, потом с двумя единицами и т. д.).

Рассмотрим подробнее случай $q = 3$

Пусть $q = 3 \Rightarrow 2^3 - 1 = 7$.

Перегруппируем столбцы:

1. x_1, x_2, x_3 с одной единицей;
2. x_4, x_5, x_6 с двумя единицами;
3. x_7 с тремя единицами.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	f
0	0	1	0	1	1	1	0
0	1	0	1	0	1	1	0
1	0	0	1	1	0	1	0

$f_1 = x_1x_2 \vee x_2x_3 \vee x_1x_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3$

$f_2 = \bar{x}_4\bar{x}_5 \vee \bar{x}_5\bar{x}_6 \vee \bar{x}_4\bar{x}_6 \vee x_4x_5x_6$

Теперь совместим x_1, x_2, x_3 с x_4, x_5, x_6 и x_7, и просуммируем с f_1 и f_2:

$f = x_1x_2 \vee x_2x_3 \vee x_1x_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_4\bar{x}_5 \vee \bar{x}_5\bar{x}_6 \vee \bar{x}_4\bar{x}_6 \vee x_4x_5x_6 \vee x_3x_4 \vee x_2x_5 \vee x_1x_6 \vee \bar{x}_7$

Получили 12 конъюнкций.

Проверим. Вернёмся к совершенной к. н. ф.:

$f = (x_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \vee \bar{x}_5 \vee \bar{x}_6 \vee \bar{x}_7) \wedge (x_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4 \vee x_5 \vee \bar{x}_6 \vee \bar{x}_7) \wedge (\bar{x}_1 \vee x_2 \vee x_3 \vee \bar{x}_4 \vee \bar{x}_5 \vee x_6 \vee \bar{x}_7) = x_1x_2 \vee x_1x_3 \vee x_1x_6 \vee \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_2x_3 \vee x_2x_5 \vee x_3x_4 \vee \bar{x}_4\bar{x}_5 \vee \bar{x}_4\bar{x}_6 \vee x_4x_5x_6 \vee \bar{x}_5\bar{x}_6 \vee \bar{x}_7$.

Случай $q = 5$

Пусть $q = 5 \Rightarrow 2^5 - 1 = 31$.

Проделаем те же операции, что и для предыдущей задачи. Перегруппируем столбцы таким образом, чтобы их можно было бы объединить в блоки: с одной единицей, с двумя единицами, с тремя, с четырью и с пятью.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
0	0	0	0	1	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	0	1	1	1	0	0	0	1
0	0	1	0	0	0	1	1	0	0	1	0	0	1	0
0	1	0	0	0	1	0	1	0	1	0	0	1	0	0
1	0	0	0	0	1	1	0	1	0	0	1	0	0	0

x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}	x_{31}	f
0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	0
0	1	1	1	0	0	0	1	1	1	1	0	1	1	1	1	0
1	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0
1	1	0	1	1	0	1	0	1	0	1	1	1	0	1	1	0
1	1	1	0	1	1	1	1	0	0	1	1	1	1	0	1	0

Рассмотрим отдельно блоки 1, 2, 3, 4, 5. Найдём д. н. ф. f_i для каждого блока, и далее совместим блоки друг с другом.

$f_1 = x_1x_2 \vee x_1x_3 \vee x_1x_4 \vee x_1x_5 \vee x_2x_3 \vee x_2x_4 \vee x_2x_5 \vee x_3x_4 \vee x_3x_5 \vee x_4x_5 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5$

$f_4 = \bar{x}_{26}\bar{x}_{27} \vee \bar{x}_{26}\bar{x}_{28} \vee \bar{x}_{26}\bar{x}_{29} \vee \bar{x}_{26}\bar{x}_{30} \vee \bar{x}_{27}\bar{x}_{28} \vee \bar{x}_{27}\bar{x}_{29} \vee \bar{x}_{27}\bar{x}_{30} \vee \bar{x}_{28}\bar{x}_{29} \vee \bar{x}_{28}\bar{x}_{30} \vee x_{29}\bar{x}_{30} \vee x_{26}x_{27}x_{28}x_{29}x_{30}$

$f_5 = \bar{x}_{31}$

Блок 2 разобьём на подблоки (2.1, 2.2, 2.3):

x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
0	0	0	0	0	0	1	1	1	1
0	0	0	1	1	1	0	0	0	1
0	1	1	0	0	1	0	0	1	0
1	0	1	0	1	0	0	1	0	0
1	1	0	1	0	0	1	0	0	0

$f_{2.1} = x_6x_7x_8 \vee \bar{x}_6\bar{x}_7x_8 \vee \bar{x}_6x_7\bar{x}_8 \vee x_6\bar{x}_7\bar{x}_8$

$f_{2.2} = x_9x_{10}\bar{x}_{11} \vee \bar{x}_9x_{10}x_{11} \vee x_9\bar{x}_{10}x_{11}$

$f_{2.3} = \bar{x}_{12}\bar{x}_{13}\bar{x}_{14}\bar{x}_{15} \vee \bar{x}_{12}\bar{x}_{13}x_{14}x_{15} \vee \bar{x}_{12}x_{13}\bar{x}_{14}x_{15} \vee \bar{x}_{12}x_{13}x_{14}\bar{x}_{15} \vee x_{12}\bar{x}_{13}\bar{x}_{14}x_{15} \vee \bar{x}_{12}\bar{x}_{13}x_{14}\bar{x}_{15} \vee x_{12}x_{13}\bar{x}_{14}\bar{x}_{15} \vee \bar{x}_{12}x_{13}x_{14}x_{15} \vee x_{12}\bar{x}_{13}x_{14}x_{15} \vee x_{12}x_{13}\bar{x}_{14}x_{15} \vee x_{12}x_{13}x_{14}\bar{x}_{15}$

Воспользуемся $\bar{x} \cdot K \vee x \cdot K = \bar{x} \cdot K \vee x \cdot K \vee K = K$:

$f_{2.3} = \bar{x}_{12}\bar{x}_{13}\bar{x}_{14}\bar{x}_{15} \vee x_{12}x_{13}\bar{x}_{15} \vee \bar{x}_{12}x_{14}x_{15} \vee x_{12}\bar{x}_{14}x_{15} \vee \bar{x}_{12}x_{14}\bar{x}_{15} \vee \bar{x}_{13}x_{14}x_{15}$

Совместим $f_{2.1}$ с $f_{2.2}$, а $f_{2.2}$ с $f_{2.3}$:

$f_{2.12} = x_6x_{11} \vee x_7x_{10} \vee x_8x_9$

$f_{2.23} = x_9x_{13} \vee x_9x_{14} \vee x_{10}x_{12} \vee x_{10}x_{14} \vee x_{11}x_{12} \vee x_{11}x_{13}$

После совмещения блока (2.1) с (2.2), а (2.2) с (2.3), совмещения блоков (2.1) и (2.3) уже не требуется. Теперь мы можем найти f_2:

$f_2 = f_{2.1} \vee f_{2.2} \vee f_{2.3} \vee f_{2.12} \vee f_{2.23}$

С блоком 3 всё обстоит аналогично блоку 2.

Осталось совместить 1 с 4, 2 с 3, 1 с 3 блоки:

$f_{14} = x_5x_{26} \vee x_4x_{27} \vee x_3x_{28} \vee x_2x_{29} \vee x_1x_{30}$

$f_{23} = x_6x_{25} \vee x_7x_{24} \vee x_8x_{23} \vee x_9x_{22} \vee x_{10}x_{21} \vee x_{11}x_{20} \vee x_{12}x_{19} \vee x_{13}x_{18} \vee x_{14}x_{17} \vee x_{15}x_{16}$

$f_{13} = x_5x_{16} \vee x_5x_{17} \vee x_5x_{18} \vee x_5x_{19} \vee x_4x_{16} \vee x_4x_{20} \vee x_4x_{21} \vee x_4x_{22} \vee x_3x_{17} \vee x_3x_{20} \vee x_3x_{23} \vee x_3x_{24} \vee x_2x_{18} \vee x_2x_{21} \vee x_2x_{23} \vee x_2x_{25} \vee x_1x_{19} \vee x_1x_{22} \vee x_1x_{24} \vee x_1x_{25}$

Наконец, остаётся сложить все f_i, $i = 1, 2, 3, 4, 5$ и f_{14}, f_{23}, f_{13}. В результате получим д. н. ф. для функции f. Подсчитаем количество слагаемых: $11+2*(4+3+6+3+6)+11+1+5+10+20 = 102$. Что существенно меньше числа конъюнкций, получаемых при прямом умножении скобок в совершенной к. н. ф. Заметим, что алгоритм не столь эффективен как алгоритм, предложенный А. Г. Дьяконовым. Однако, он тоже выдаёт неплохие результаты и достаточно прост в реализации.

Список литературы

[1] Журавлёв Ю. И. Оценки сложности алгоритмов построения минимальных дизъюнктивных форм для функций алгебры логики. — Сб. Трудов Института математики Сиб. отд. АНСССР. Дискретный Анализ. 1964. Вып. 3.

[2] Журавлёв Ю. И., Коган А. Ю. Реализация булевых функций с малым числом нулей дизъюнктивными нормальными формами и смежные задачи. — Докл. АН СССР. 1985. Т. 285. № 4. С. 795-799.

[3] Журавлёв Ю. И. О различных понятиях минимальности дизъюнктивных нормальных форм. — Сиб. матем. журнал. 1960. Т. 1. № 4. С. 609-610.

[4] Журавлёв Ю. И. Алгоритмы построения минимальных дизъюнктивных нормальных форм для функций алгебры логики. — Дискретная математика и математические вопросы кибернетики. Под редакцией Яблонского С. В., Лупанова О. Б. М.:Наука, 1974

[5] Максимов Ю. В. Реализация булевых функций с ограниченным числом нулей в классе дизъюнктивных нормальных форм. — Ж. вычисл. матем. и матем. физ. 2013. Т. 53. № 9. С. 1569-1588.

[6] Журавлёв Ю. И., Коган А. Ю. Алгоритм построения дизъюнктивной нормальной формы, эквивалентной произведению левых частей булевых уравнений нельсоновского типа. — Ж. вычисл. матем. и матем. физ. 1986. Т. 26. № 8. С. 1243-1249.

[7] Дьяконов А. Г. Реализация одного класса булевых функций с малым числом нулей дизъюнктивными нормальными формами. — Ж. вычисл. матем. и матем. физ. 2001. Т. 41. № 5. С. 821-828.

[8] Дьяконов А. Г. Построение дизъюнктивных нормальных форм в задачах распознавания образов с бинарной информацией. — Докл. АН. 2002. Т. 383. № 6. С. 747-749.

[9] Дьяконов А. Г. Построение дизъюнктивных нормальных форм в логических алгоритмах распознавания. — Ж. вычисл. матем. и матем. физ. 2002. Т. 42. № 12. С. 1899-1907.

[10] Коган А. Ю. О дизъюнктивных нормальных формах булевых функций с малым числом нулей. — Ж. вычисл. матем. и матем. физ. 1987. Т. 27. № 6. С. 924-931.

[11] Журавлёв Ю. И. Об алгебраическом подходе к решению задач распознавания или классификации. — Пробл. кибернетики. М.: Наука, 1978. Вып. 33. С. 5-68.