

Алгоритмы минимизации функций алгебры логики в классе дизъюнктивных нормальных форм и оценки их сложности

А.А. Байжуманов¹, М.А.Бердимуродов²

¹к.ф.-м.н. , доцент кафедры «Математика» Южно Казахстанского государственного педагогического университета, Шымкент.

²PhD, доцент кафедры «Цифровые технологии» Perfect university, Ташкент.

Аннотация. В работе рассматривается проблема минимизации функций алгебры логики в классе дизъюнктивных нормальных форм. Разрабатываются алгоритмы полного перебора и ограниченной сложности поиска кратчайших дизъюнктивных нормальных форм функций алгебры логики и вычисляются оценки их сложности. Строится алгоритм минимизации функций алгебры логики в классе дизъюнктивных нормальных форм на основе вычисления погрешности окрестности первого порядка элементарных конъюнкций. Исследуется локальный анализ в алгоритме направленного перебора и дается оценка его алгоритмической сложности.

Ключевые слова. кратчайших дизъюнктивных нормальных форма, алгебры логики, окрестности первого порядка, оценки, минимизации.

Diz'yunktiv normal shakllar sinfidagi mantiqiy algebra funksiyalarini minimallashtirish va ularning murakkabligini baholash algoritmlari

Annotatsiya. Maqolada disjunktiv normal shakllar sinfida mantiqiy algebra funksiyalarini minimallashtirish muammosi ko'rib chiqiladi. Mantiq algebrasidagi funksiyalarning eng qisqa dis'yunktiv normal shakllarini topish uchun to'liq tanlash usulidan optimallashtirilgan algoritmlari ishlab chiqilgan va ularning murakkabligi hisoblangan. Elementar kon'yunksiyalarning birinchi tartibli qo'shniligi xatosini hisoblash asosida mantiq algebrasining dis'yunktiv normal shakllar sinfidagi funksiyalarini minimallashtirish algoritmi tuziladi. Yo'naltirilgan qidiruv algoritmda lokal tahlil o'rganiladi va uning algoritmik murakkabligi bahosi beriladi.

Kalit so'zlar. Eng qisqa dis'yunktiv normal shakllar, mantiq algebrasi, birinchi tartibli atrof, baholash, minimallashtirish.

Введение. В алгебре логики существует несколько способов задания формул, реализующих её функции. Эти способы включают построение дизъюнктивных нормальных форм (д.н.ф.) из элементарных конъюнкций(э.к.). В статье рассматривается проблема минимизации функций алгебры логики в классе дизъюнктивных нормальных форм. Разрабатываются алгоритмы полного перебора и ограниченной сложности поиска кратчайших дизъюнктивных нормальных форм функций алгебры логики и вычисляются оценки их сложности.

В процессе минимизации целесообразно в первую очередь выделить ядровые конъюнкции (не покрываемые никаким набором конъюнкций) из сокращенной д. н. ф. Свойство конъюнкции входить или не входить во множество ядровых конъюнкций однозначно определяется окрестностью первого порядка, т. е. множеством конъюнкций, имеющих хотя бы одну общую точку с исследуемой конъюнкцией, на основании чего, например, строится алгоритм Квайна. Исследование свойств конъюнкций, входящих в окрестность первого порядка для произвольной д. н. ф., может в той или иной степени

повлиять на процесс упрощения алгоритмов минимизации. Для почти всех функций на основе работ Н.Шестеревой вычислена погрешность первого порядка (сумма рангов конъюнкций, входящих в окрестность первого порядка, за исключением ранга исследуемой конъюнкции) для произвольной конъюнкции, и данный результат применен в этой работе. Построен алгоритм минимизации функции алгебры логики, который основан на использовании свойств погрешностей первого порядка, в результате чего существенно уменьшается перебор различных д.н.ф.

1. Постановка задачи. Построение минимальных и кратчайших дизъюнктивных нормальных форм (д.н.ф.) функций алгебры логики от n переменных связано с удалением элементарных конъюнкций (э.к.) из сокращенной д.н.ф. этих функций.

Рассмотрим два алгоритма минимизации функций алгебры логики, первый алгоритм – направленного перебора, второй – локальный.

До начала работы каждого из алгоритмов из сокращенной д.н.ф. D_c функции f строим д.н.ф. Квайна D_Q [9]. Для этого по очереди для каждой э.к. K из д.н.ф. D_c проверяется, входит ли она в ядро или нет, т.е. для множества э.к. $\{K_1, K_2, \dots, K_r\}$ из окрестности первого порядка э.к. K в д.н.ф. D проверяется включение $N_K \subseteq \bigcup_{j=1}^r N_{K_j}$. Если включение не имеет места, то э.к. K отмечаем как входящую в ядро. Затем для всех э.к. K , не входящих в ядро, проверяется включение $N_K \subseteq \bigcup_{j=1}^s N_{K_j}$, где $K_j (j=1, S)$ – ядровые конъюнкции. Если включение имеет место, то данная э.к. удаляется из д.н.ф. D_c , из оставшихся формируется д.н.ф. Квайна D_Q .

Пусть д.н.ф. Квайна имеет вид: $D_Q = K_1 \vee K_2 \vee \dots \vee K_m$.

Множество всех ядровых конъюнкций $\{K_{i_1}, K_{i_2}, \dots, K_{i_s}\}$ входит в каждую тупиковую д.н.ф. функции f , и, следовательно, и в каждую минимальную и кратчайшую д.н.ф.

2. Синтез минимальных формул функций алгебры логики при помощи перебора.

Пусть D – дизъюнкция некоторого количества э.к., входящих в д.н.ф. $D_Q / \bigcup_{j=1}^s K_{i_j}$, где э.к. $K_{i_j} (j=1, S)$ – ядровые. Расположим э.к. в д.н.ф. $D_Q / \bigcup_{j=1}^s K_{i_j}$ в порядке возрастания их рангов: $r_1 \leq r_2 \leq \dots \leq r_{m-s}$.

Рассмотрим $(m-s)$ – мерный единичный куб ε^{m-s} , каждый набор которого с координатами $\tilde{a} = (a_1, a_2, \dots, a_{m-s})$ соответствует д.н.ф. D следующим образом: K_j входит в д.н.ф. D , если $a_j = 1$ и не входит в д.н.ф. D , если $a_j = 0, (j=1, m-S)$.

Назовем j – м уровнем куба ε^{m-s} множество всех наборов a , у которых $\|\tilde{a}\| = j, j = \overline{0, m-S}$ (количество единичных координат равно j). Упорядочим наборы по уровням в антилексикографическом порядке.

На множестве наборов $(m-s)$ – мерного куба ε^{m-s} введем функции $\varphi(a_1, a_2, \dots, a_{m-s})$ и $\varphi(a_1, a_2, \dots, a_{m-s})$ следующим образом: $\varphi(a_1, a_2, \dots, a_{m-s}) = t$, если для соответствующей данному набору $(a_1, a_2, \dots, a_{m-s})$ д.н.ф. D и множеству ядровых э.к. $K_{i_j}, (j=1, S)$

выполняется $|N_D \cup \bigcup_{j=1}^s N_{K_{i_j}}| = t$, $\varphi(a_1, a_2, \dots, a_{m-s}) = \sum_{j=1}^{m-S} |N_{K_j}| \cdot \alpha_j + \sum_{j=1}^S |N_{K_{i_j}}|$.

Очевидно, что $|\bigcup_{j=1}^s N_{K_{i_j}}| \leq \varphi(\tilde{a}) \leq |N_f|, \sum_{j=1}^S |N_{K_{i_j}}| \leq \varphi(\tilde{a}) \leq (m-s) |N_{K_j}| + \sum_{j=1}^S |N_{K_{i_j}}|$.

$\varphi(a_1, a_2, \dots, a_{m-s})$ и $\varphi(a_1, a_2, \dots, a_{m-s})$ – монотонные функции.

Если $\varphi(a_1, a_2, \dots, a_{m-s}) = |N_f|$, то дизъюнкция соответствующей данному набору $(a_1, a_2, \dots, a_{m-s})$ д.н.ф. D и множеству ядровых конъюнкций реализует функцию f .

Если для набора $\tilde{a} \in \varepsilon^{m-s}$ выполняется $\varphi(\alpha) = |N_f|$ и для любого набора $\tilde{a}$ такого, что $\|\tilde{a}\| = \|\tilde{a}\| + 1, \rho(\tilde{a}, \tilde{a}) = 1$ ($\rho(\tilde{a}, \tilde{\beta})$ – количество различных между собой координат наборов $\tilde{a}$ и $\tilde{\beta}$), то соответствующая этому набору д.н.ф. D с множеством ядровых конъюнкций образует тупиковую д.н.ф. функции f . Множество всех наборов, которых определяет тупиковые д.н.ф. функции f , образует нижнюю границу значения $|N_f|$ функции φ , а соответствующие им д.н.ф. с множеством ядровых конъюнкций – множество всех тупиковых д.н.ф. функции f .

Функция $\varphi(\tilde{a}) = \sum_{j=1}^{m-s} 2^{n-r_j} \cdot a_j + \sum_{j=1}^s 2^{n-r_j}$ и для ее вычисления не требуется большого объема работы, в то время как для вычисления значений функции $\varphi(a)$ необходимо рассмотреть множество точек, входящих в интервалы соответствующий д.н.ф. и просчитать в этом множестве количество различных между собой точек.

Если $\varphi(\tilde{a}) < |N_f|$, то и $\varphi(\tilde{a}) < |N_f|$, поэтому целесообразно сначала просчитать величину $\varphi(\tilde{a})$ для набора $\tilde{a}$, а затем, если есть необходимость, и значения функции $\varphi(\tilde{a})$ на этом наборе.

Алгоритм состоит из трех этапов.

1-й этап. Вычисляем значение $\varphi(\tilde{a}^1)$ на крайнем левом наборе произвольного уровня j . Пусть на i – м шаге вычислено значение $\varphi(\tilde{a}^i)$ на крайнем левом наборе $\tilde{a}^i$ – го уровня ($1 \leq i \leq m-s-j+1$). Если $\varphi(\tilde{a}^i) < |N_f|$, то на $(i+1)$ – м шаге вычисляем значение $\tilde{a}^i(j+i)$ на левом крайнем наборе $\tilde{a}^{i+1}(j+i)$ – го уровня. Если $\varphi(\tilde{a}^{i+1}) \geq |N_f|$, то переходим ко второму этапу алгоритма.

Если на $m-s-j+1$ – м шаге получено $\varphi(\tilde{a}^{m-s-j+1}) = |N_f|$, то D_Q является неупрощаемый д.н.ф. функции f , и алгоритм заканчивает работу.

2-й этап. Повторяем операции 1-го этапа, но уже с функцией $\varphi(\tilde{a})$ и начинаем работу с крайнего левого набора $\tilde{a}^i(j+i-1)$ –го уровня.

Если после τ - го шага $\varphi(\tilde{a}^{i+\tau}) = |N_f|$, то переходим к третьему этапу ($1 \leq \tau \leq m-s-j+i-1$).

Если на $m-s-j-i+1$ – м шаге получаем $\varphi(\tilde{a}^{m-s-j-i+1}) = |N_f|$, то д.н.ф. D_Q является неупрощаемой д.н.ф. функции f и алгоритм заканчивает свою работу.

3-й этап. Пусть сделано $i+\tau$ шагов, и $\varphi(\tilde{a}^{i+\tau}) = |N_f|$, тогда на $i+\tau+1$ – м шаге вычисляем значение $\varphi(\tilde{a})$ на множестве наборов $\tilde{a}(j+i+\tau-2)$ - го уровня слева направо в порядке их расположения, для которых $\rho(\tilde{a}^{i+\tau}, \tilde{a}) = 1$. Если для всех таких наборов $\tilde{a}$ выполняется $\varphi(\tilde{a}) < |N_f|$, фиксируем набор $\tilde{a}^{i+\tau}$ как соответственно тупиковой д.н.ф. функции f и вычисляем значение функции φ на соседнем справа от $\tilde{a}^{i+\tau}$ наборе $(j+i+\tau-1)$ –го уровня; при $\varphi(\tilde{a}^{i+\tau+1}) < |N_f|$, вычисляем значение функции φ на левом крайнем наборе $\tilde{a}^{i+\tau+2}$ уровня $j+i+\tau$ таком, для которого значение φ еще не существует, то вычисляем значение функции φ для следующего справа от $\tilde{a}^{i+\tau+1}$ набора $j+i+\tau-1$ –го уровня; если все следующие наборы $j+i+\tau-1$ –го уровня дают значение функции φ меньше $|N_f|$, то алгоритм заканчивает работу. При $\varphi(\tilde{a}^{i+\tau+1}) = |N_f|$ вычисляем значение φ на

множестве наборов $\alpha(j+i+\tau-2)$ – го уровня, для которых $\rho(\tilde{a}^{i+\tau+1}, \tilde{a})=1$ и значение φ еще не определено в результате предыдущих шагов алгоритма.

В результате работы алгоритма получаем некоторое множество наборов, которые определяет тупиковые д.н.ф. функции f . Среди них наберем наборы, принадлежащие минимальному по величине уровню. Такие наборы будут определять кратчайшие д.н.ф. функции f . Если один из наборов, определяющих кратчайшие д.н.ф. функции f , является крайним левым набором на своем уровне, то он определяет минимальную д.н.ф. функции f (это следует из порядка расположения конъюнкций в д.н.ф. D_Q). В случае отсутствия такого набора минимальная д.н.ф. ищется перебором всех тупиковых д.н.ф. функции f , соответствующие наборы которых расположены по уровням (в порядке возрастания уровней) в антилексикографическом порядке.

Определим, с какого уровня следует начинать работу алгоритма. Так как для почти всех функций f алгебры логики от n – переменных длина $l_k(f)$ кратчайший д.н.ф. находится в пределах $\frac{2^{n-1}}{\log \log n \cdot \log n} \leq l_k(f) \leq \frac{2^n}{\log n}$,

начинать работу алгоритма следует при $j: \left\lceil \frac{2^{n-1}}{\log \log n \cdot \log n} \right\rceil \leq j \leq \left\lfloor \frac{2^n}{\log n} \right\rfloor + 1$.

Так как количество ядровых конъюнкций для почти всех функций f алгебры логики не больше $n^{(1+\varepsilon) \log \log n}$, $\lim \varepsilon = 0$, они не вносят существенной вклад в длину кратчайшей д.н.ф. и не изменяют границу для f .

При расположении конъюнкций в д.н.ф. D_Q в порядке возрастания рангов получается, что конъюнкции, содержащие большее количество точек, расположены в начале этого ряда, поэтому они и будут составлять основу покрытия множества N_f .

Для уменьшения шагов 1-го и 2-го этапов алгоритма будем разыскивать набор $\tilde{a}$ такой, что $\varphi(\tilde{a}) = |N_f|$ и для любого набора $\tilde{a}$ такого, что $\|\tilde{a}\| = \|\tilde{a}\| + 1, \rho(\tilde{a}, \tilde{a}) = 1, \varphi(\tilde{a}) < |N_f|$, используя метод деления отрезков пополам, для этого полагаем

$$j_0 = \left\lfloor \frac{\left(\left\lceil \frac{2^{n-1}}{\log \log n \cdot \log n} \right\rceil + \left\lfloor \frac{2^n}{\log n} \right\rfloor \right)}{2} \right\rfloor$$

и для крайнего левого набора $\tilde{a}^0$ уровня j_0 проверяем значение $\varphi(\tilde{a}^0)$. При $\varphi(\tilde{a}^0) = |N_f|$ проверяем аналогичным образом крайний левый набор

$$\text{уровня } j_1 = \left\lfloor \frac{\left(j_0 + \left\lceil \frac{2^{n-1}}{\log \log n \cdot \log n} \right\rceil \right)}{2} \right\rfloor, \quad \text{при } \varphi(\tilde{a}^0) < |N_f| - \text{крайний левый набор уровня}$$

$$j_2 = \left\lfloor \left(j_0 + \left\lfloor \frac{2^n}{\log n} \right\rfloor + 1 \right) / 2 \right\rfloor.$$

И так до тех пор, пока не найдем нужного набора. Аналогичным образом ищем крайний левый набор $\tilde{a}$, у которого $\varphi(\tilde{a}) = |N_f|$ и для любого набора $\tilde{a}$, такого, что $\|\tilde{a}\| = \|\tilde{a}\| + 1, \rho(\tilde{a}, \tilde{a}) = 1, \varphi(\tilde{a}) < |N_f|$.

В случае, если достаточно найти одну кратчайшую д.н.ф. функции f , алгоритм прекращает работу как только найдется набор $\tilde{a}$ j -го уровня такой, что $\varphi(\tilde{a}) = |N_f|$ и для любого набора $\tilde{a}(j-1)$ – го уровня $\varphi(\tilde{a}) < |N_f|$.

Так как φ – монотонная функция, то данный алгоритм можно рассматривать как алгоритм расшифровки монотонной функции ψ от m переменных:

$$\psi(x_1, x_2, \dots, x_m) = \begin{cases} 1, & \text{при } \varphi(\tilde{a}) = |N_f|, \\ 0, & \text{при } \varphi(\tilde{a}) < |N_f|. \end{cases}$$

Работу алгоритма расшифровки монотонной функции от n переменных следует начинать с произвольного набора $\left\lfloor \frac{m}{2} \right\rfloor$ -го уровня, так как для почти всех функции f алгебры логики от m переменных

$$2^{m-1} - m \cdot 2^{\frac{m}{2}} \leq |N_f| \leq 2^{m-1} + m \cdot 2^{m/2}$$

и множество наборов из $|N_f|$ сосредоточено в узкой полосе куба ε^m шириной не более $2\sqrt{m} \cdot \log m$, состоящей из средних слоев.

$$\text{В [3] показано, что } G(m) = \min_F \max_{\psi} G(F, \psi) = C_m^{\left\lfloor \frac{m}{2} \right\rfloor} + C_m^{\left\lfloor \frac{m}{2} \right\rfloor + 1} \sim \frac{2^{m+\frac{3}{2}}}{\sqrt{\pi m}},$$

где $G(F, \psi)$ – число обращений к оператору, определяющему значение функции ψ на наборе $\tilde{a}$, достаточное для восстановления таблицы значений монотонной функции ψ при использовании алгоритма F .

Процесс поиска кратчайший д.н.ф функции f аналогичен алгоритму поиска нижней единицы (верхнего нуля) монотонной функции $\psi(\tilde{x})$.

В доказываемом, что в последнем случае $G(m) = C_m^{\left\lfloor \frac{m}{2} \right\rfloor} + 1 \sim \frac{2^{m+\frac{1}{2}}}{\sqrt{\pi m}}$ что не уменьшает существенно трудоемкость алгоритма.

Однако на функции $\psi(x_1, x_2, \dots, x_m)$, используемые в исследуемом алгоритме, не распространяется порядок расположения наборов в кубе ε^m , принадлежащих $|N_{\psi}|$. Как уже было отмечено, наборы, соответствующие кратчайшей д.н.ф. для почти всех функций алгебры логики лежат на полосе между уровнями с номерами $\left\lfloor \frac{2^{n-1}}{\log \log n \cdot \log n} \right\rfloor$ и $\left\lfloor \frac{2^n}{\log n} \right\rfloor + 1$.

Если использовать это свойство длин кратчайших д.н.ф. то для почти всех функций алгебры логики данным алгоритмом будет производиться количество обращений к оператору, определяющему значение функции ψ , не более чем $C_m^{\left\lfloor \frac{2^n}{\log n} \right\rfloor + 1}$ раз. Если учесть, что для почти всех функций алгебры логики $m \leq n^{(1+\varepsilon) \log \log n} \cdot 2^n$, $\lim \varepsilon = 0$, то алгоритм поиска нижней единицы (верхнего нуля) монотонной функции алгебры логики в нашем случае будет для почти всех функций от n переменных делать в

$$\frac{2^{2^n \cdot n^{\log \log n} + 1} \cdot (\log n)^{2^n \cdot n^{\log \log n} - 2^{\frac{n+1}{\log n} - \frac{1}{2}}}}{(n^{\log \log n})^{2^n / \log n + \frac{1}{2}}}$$

Больше шагов, чем алгоритм, учитывающий метрические свойства булевых функций.

3. Алгоритм минимизации функций алгебры логики при помощи погрешности первого порядка. Рассмотрим д.н.ф. Квайна

$$D_Q = K_1 \vee K_2 \vee \dots \vee K_m, \text{ где конъюнкции } \{K_{i_1}, K_{i_2}, \dots, K_{i_s}\} \text{ – ядровые.}$$

Для каждой э.к. $K \in \text{д.н.ф. } D$ определяем погрешность первого порядка (сумма рангов конъюнкций, входящих во множество $T_1 = \{S_1(K, D_Q) / K\}$).

Упорядочиваем конъюнкции $K_j (j=1, m)$ по возрастанию их погрешностей первого порядка, причем в этой последовательности отмечаем ядровые конъюнкции информационным символом (2). До начала работы алгоритма конъюнкции из д.н.ф. D_Q либо имеет отметку (2), либо имеет отметку (∇).

Пусть выполнено i шагов, и отметку (0) получили э.к. $K_{i_1}^{(0)}, K_{i_2}^{(0)}, \dots, K_{i_r}^{(0)}$, а отметку (1) – э.к. $K_{i_1}^{(1)}, K_{i_2}^{(1)}, \dots, K_{i_r}^{(1)}$.

$(i+1)$ -й шаг. Берем первую слева неядровую э.к. (т.е. имеющую отметку (∇)) из д.н.ф. $D_Q / \bigwedge_{j=1}^r K_{i_j}^{(0)} \vee \bigwedge_{j=1}^r K_{i_j}^{(1)}$.

Если эта д.н.ф. пустая либо состоит только из э.к., имеющих отметку (2), то алгоритм заканчивает работу.

Рассматриваем окрестность первого порядка данной э.к. и проверяем соотношение

$$N_K \cap \bigwedge_{j=1}^r N_{K_{i_j}^{(0)}} = \emptyset \quad (1)$$

Если оно выполнено, то даем э.к. K отметку (0), и $(i+1)$ -й шаг алгоритма заканчивается. Если соотношение не выполнено, то проверяем включение

$$N_K \subseteq N_{D_Q} / \bigwedge_{j=1}^r N_{K_{i_j}^{(0)}} \quad (2)$$

Если включение выполняется, то э.к. K получает отметку (0), в противном случае (1), и переходим к следующему шагу.

После окончания работы алгоритма э.к., получившие отметку (0), удаляются из д.н.ф. D_Q .

В результате работы алгоритма получается тупиковая д.н.ф. функции f .

Докажем корректность алгоритма.

Теорема 1. В процессе работы алгоритма минимизации с помощью погрешности первого порядка получается тупиковая д.н.ф. D_m функции f .

Доказательство. Действительно, в построенной д.н.ф. D остались э.к. с отметками (1) и (2). Ядровые конъюнкции входят во все тупиковые д.н.ф. функции f , и, следовательно, при удалении хотя бы одной из них, д.н.ф. D перестает реализовать функцию f .

Э.к. K получит отметку (1), если выполняется соотношение $N_K \not\subseteq N_{D_Q} / \bigwedge_{j=1}^r N_{K_{i_j}^{(0)}}$, где $K_{i_1}^{(0)}, K_{i_2}^{(0)}, \dots, K_{i_r}^{(0)}$ – э.к., получившие отметку (0) и расположенные в д.н.ф. D_Q перед э.к. K . Следовательно

$$N_K \not\subseteq \bigwedge_{j=1}^r N_{K_{i_j}^{(0)}} \cup \left\{ \bigwedge_{j=1}^r N_{K_{i_j}^{(1)}} \right\} \quad (3)$$

и при удалении конъюнкции K из д.н.ф. D получается д.н.ф., не реализующая функцию f . Теорема доказана.

Дадим оценку сложности работы данного алгоритма.

В алгоритме для обозначения э.к. K будем использовать n мерный набор, в котором на i – м месте стоит 0, если переменная x_i входит в K с отрицанием; 1, если x_i входит в K без отрицания; 2, если переменная x_i не входит в K .

После этого, чтобы определить, пересекаются ли две э.к., необходимо по координатно просуммировать их по модулю 3. Если в полученном n – мерном наборе нет единичных координат, то э.к. пересекаются хотя бы одной точкой. Для определения вхождения произвольной вершины в интервал N_K , достаточно по координатно

просуммировать по модулю 3 n – мерные наборы, соответствующие данной вершине и э.к. К. Если в полученном n – мерном наборе нет единиц, то вершина входит в интервал, соответствующий данной э.к. К.

Пусть оператор А выполняет сложение по модулю 3 n – мерных наборов. Определим число обращений к оператору А, достаточное для построения тупиковой д.н.ф. с помощью исследуемого алгоритма для произвольной функции.

Для определения окрестности первого порядка и вычисления погрешности неядровых коъюнкций требуется не более $(m-S)(m-1)$ обращений к оператору А.

Для проверки соотношения (1) требуется не более

$$(m-S)/2 \cdot (m-S-1)$$

обращений к оператору А.

Для проверки включения (2) требуется не более $n^{\log \log n} \cdot (m-S-1)$ обращений к оператору А. Так как для почти всех функций алгебры логики от n переменных длина сокращенной д.н.ф. функции не больше $n^{(1-\epsilon_n) \log \log n} \cdot 2^n$, $\lim \epsilon_n = 0, n \rightarrow \infty$, то можно записать, что для проверки включения (2) требуется не более $\frac{(m-S-1)}{2^n} \cdot m$ обращений к оператору А.

Окончательно получаем, что $G(A) \leq \frac{3m^2}{2} - O(m^2)$, где $G(A)$ – число обращений к оператору А, достаточное для построения тупиковых д.н.ф. произвольной функции f алгебры логики при использовании алгоритма минимизации, описанного выше.

Рассмотрим работу алгоритма на примере циклических функций, для которых нельзя проверить вхождение э.к. в D_{UM} , используя информацию из окрестности конечного порядка.

Определение. Множество вершин куба ε^n называется циклом, если его точки можно расположить в виде последовательности $\tilde{a}^1, \tilde{a}^2, \dots, \tilde{a}^e$ так, что

$$a) \rho(\tilde{a}^i, \tilde{a}^{i+1}) = 1, \rho(\tilde{a}^1, \tilde{a}^e) = 1;$$

$$б) \text{ для всех остальных пар } \rho(\tilde{a}^i, \tilde{a}^j) > 1.$$

Функция φ называется циклической, если N_φ есть цикл.

Для циклической функции $D_c = \{j=1 \dots e \mid K_j = D_Q\}$,

где K_j – э.к., соответствующий интервал которой есть ребро, проходящее через точки $\tilde{a}^j$ и $\tilde{a}^{j(mod e)+1}$ ($j = \overline{1, e}$). В циклической функции ядровых точек нет, погрешность первого порядка у всех э.к. одинакова.

Лемма 1. Длина тупиковой д.н.ф. $l(D_T)$ циклической функции $\varphi = \{j=1 \dots G_m \mid K_j\}$ не превосходит величины $4m$.

Доказательство. Множество N_φ содержит G_m точек, каждый максимальный интервал покрывает только две точки. Неприводимыми для функции φ будут лишь те покрытия, в которых каждый интервал N_{K_j} либо не имеет общих точек ни с одним другим интервалом из данного покрытия (первый тип), либо имеет в окрестности первого порядка только один интервал $N_{K_{j-1}}$ либо $N_{K_{j+1}}$ (второй тип).

В случае, если тупиковая д.н.ф. функции φ состоит только из интервалов первого типа, то количество э.к. в ней будет равно $3m$, соответствующая тупиковая д.н.ф. будет и кратчайшей. Если тупиковая д.н.ф. функция φ состоит из интервалов только второго типа, то таких интервалов – $4m$ (три точки покрываются двумя интервалами), данная

тупиковая будет самой длинной из множества тупиковых для функции φ д.н.ф. Лемма доказана.

Будем считать порядок расположения э.к. в д.н.ф. $D_c(\varphi)$ основным, если э.к. расположены в том же порядке, что и последовательность ребер в цикле, состоящем из точек $\tilde{a}^1, \tilde{a}^2, \dots, \tilde{a}^{G_m}$. Остальные всевозможные расположения э.к. представляют собой некоторую перестановку основного порядка.

Если э.к. расположены в основном порядке, то в процессе работы алгоритма отметку (0) получают все конъюнкции с четными порядковыми номерами, полученная тупиковая д.н.ф. будет содержать в себе интервалы только первого типа, данная д.н.ф. будет и кратчайшей д.н.ф. функции φ .

Если сделать перестановку основного порядка так, что в начале ряда будут стоять конъюнкции, чей порядковый номер кратен трем, то алгоритм всем этим конъюнкциям ставит отметку (0), а все остальные получают отметку (1). В данном случае все интервалы будут второго типа, и полученная тупиковая д.н.ф. будет длиннейшей из всех тупиковых д.н.ф. функции φ .

4. Локальный анализ в алгоритме направленного перебора. На примере алгоритмов минимизации булевых функций видно, что:

а) использование любой схемы направленного перебора затруднено в связи с экспоненциальной оценкой трудоемкости, хотя и дает возможность конструировать кратчайшую д.н.ф. функции;

б) локальный анализ окрестности существенно уменьшает сложность алгоритма, но полученная в результате тупиковая д.н.ф. может не только не быть кратчайшей, но и даже самой длинной из всех тупиковых д.н.ф. функции.

Естественно возникает задача построения алгоритма минимизации функций алгебры логики, использующий локальный анализ окрестности первого порядка в одной из схем направленного перебора так, чтобы длина полученной тупиковой д.н.ф. отличалась от длины кратчайшей не более чем на определенную величину. Для этого можно использовать схему направленного перебора, являющуюся аналогом алгоритма поиска максимального верхнего нуля [3], исходя из сокращенной д.н.ф., предварительно упрощенной в результате анализа э.к. по их окрестности первого порядка.

До начала работы алгоритма э.к. либо имеет отметку (2)-ядровые, либо отметку (∇). Упорядочим э.к. в сокращенной д.н.ф. по возрастанию их рангов. Каждому набору $\tilde{\square} \in N_f$ поставим в соответствие индекс $q(\tilde{\square}) = |S_1(\tilde{\square}), D|$, где д.н.ф. D – дизъюнкция тех э.к. из сокращенной д.н.ф., которые на данном шаге работы алгоритма имеют отметку (∇).

Набор $\tilde{\square} \in N_K$ называется **регулярным** относительно (K, D) (э.к. K входит в д.н.ф. D), если существует набор $\tilde{\square}'$ такой, что

- 1) $\tilde{\square}' \in N_D / N_K$;
- 2) $S_1(\tilde{\square}', D) \subseteq S_1(\tilde{\square}, D)$.

Э.к. K называется **регулярной** относительно (K, D) , если все ее наборы регулярны относительно (K, D) .

Алгоритм позволяет выявить все регулярные э.к., которые не входят в д.н.ф. $D_{UT}[2]$.

Алгоритм состоит из двух этапов.

1-й этап. Пусть выполнено i шагов, и отметку (1) получили э.к. $K_{i_1}, K_{i_2}, \dots, K_{i_r}$.

(i+1)-й шаг. Рассматриваем первую слева э.к. K с отметкой (∇), ищем во множестве $N_K / \bigcup_{j=1}^r N_{K_{i_j}^{(2)}} \cup \bigcup_{j=1}^r N_{K_{i_j}^{(1)}}$ набор $\tilde{\square}$ минимального индекса

(здесь достаточно рассмотреть э.к. $K_{i_j}^{(1)}$ и $K_{i_j}^{(2)}$ из окрестности первого порядка э.к. K). Последовательно в порядке уменьшения рангов даем отметку (0) э.к. из окрестности первого порядка $S_1(\tilde{\square}, D)$ с отметкой (∇), одновременно на единицу понижая индексы всех наборов, входящих в очередную получившую отметку (0) э.к. Так как индексы всех

наборов ограничены ($q(\tilde{\square}) \leq l_c(f)$), наступит момент, когда либо индекс набора $\tilde{\square}$ будет равен 1, либо набора $\tilde{\square}'$ в одной э.к. из $S_1(\tilde{\square}, D)$.

В первом и втором случаях э.к. K получает отметку (1) и $(i+1)$ -й шаг алгоритма заканчивается. В третьем случае отметку (1) получает э.к., содержащая набор $\tilde{\square}'$ единичного индекса, а в э.к. K набор $\tilde{\square}$ фиксируется как слабо регулярный (очевидно, что регулярные наборы также будут отмечены среди множества слабо регулярных наборов, так как $S_1(\tilde{\square}', D) \subseteq S_1(\tilde{\square}, D)$ и $\tilde{\square}' \notin N_K$), и все операции повторяются со следующим набором минимального индекса из э.к. K . Если все наборы э.к. K слабо регулярны, то э.к. K получает отметку (0), и $(i+1)$ -й шаг алгоритма заканчивается.

2-й этап. Пусть сделано l шагов алгоритма. Удалим э.к. с отметкой (0) и будем считать с отметкой (1) входящими во все тупиковые д.н.ф. K оставшиеся д.н.ф. D , состоящей из э.к. с отметкой (∇), применим алгоритм минимизации, использующий схему направленного перебора [1,9].

Докажем корректность построенного алгоритма.

Все э.к., получившие отметку (1), имеют хотя бы один набор с единичным индексом (ядровой относительно д.н.ф. $D_c / \{j=1\} p K_{i_j}^{(0)}$). Алгоритм перебора строит кратчайшее покрытие множества N_D .

Оценим длину полученной тупиковой д.н.ф. Рассмотрим наборы с величиной индекса не меньше v (обозначим это множество $N_f^{\square}$) и э.к. размерностью не меньше d (величины v и d определим позже). Применим алгоритм ко множеству $D_c^{\square}$ э.к. K из сокращенной д.н.ф. D_c таких, что $N_K \subseteq N_f^{\square}$.

Пусть δ_l - доля наборов из $N_f^{\square}$, не покрытых после l -го алгоритма. Каждый из оставшихся $\delta_l |N_f^{\square}|$ наборов имел первоначально величину индекса не меньше v , следовательно среди д.н.ф. с отметкой (∇) найдется э.к., покрывающая не менее $\delta_l / N_f^{\square} / \frac{\square}{m}$ ($m = l_c(f)$) оставшихся наборов, в случае, если все ее точки окажутся слабо регулярными, то из ее окрестности первого порядка можно выделить не более 2^d э.к., покрывающих не менее $\delta_l / N_f^{\square} / \frac{\square}{m}$ наборов. Значит

$$\delta_{l+1} / N_f^{\square} / \frac{\square}{m} \leq \delta_l / N_f^{\square} / \frac{\square}{m} - \delta_l / N_f^{\square} / \frac{\square}{m} \cdot 2^d,$$

откуда

$$\delta_{l+1} \leq \delta_l (1 - \frac{\square}{m} \cdot 2^d).$$

Очевидно, что для любого натурального l длина $l(f)$ полученной после l -го шага д.н.ф. не превосходит $l \cdot 2^d$, и остается построить покрытие не более чем $\delta_l / N_f^{\square} / + |N_f / N_f^{\square}|$ наборов из оставшихся э.к.

Для определения величины v используем среднее значение $L_{;k}(f)$ - числа максимальных k -мерных интервалов функции f , содержащих $\tilde{\square}$ и определим величину отклонения $L_{;k}(f)$ от среднего значения для почти всех функций. Будем считать, что $\bar{L}_{\tilde{\square},k}(n)$ - математическое ожидание случайной величины $\xi_{n,k}$, принимающей значение $L_{;k}(f)$ с вероятностью

$L_{;k}(f) / 2^{2^n-1}$. Найдем дисперсию этой величины:

$$M \square_{n,k}^2 = \frac{1}{2^{2^n-1}} \sum_{s,j=1}^{C_n^k} \Phi_{\tilde{\square}}(K_j, K_s),$$

где $\Phi_{\tilde{\square}}(K_j, K_s)$ - число таких функций $f (1 \leq i \leq 2^{2^n-1})$, для которых

$N_{K_j} \subseteq N_{f_i}, N_{K_s} \subseteq N_{f_i}, \tilde{\square} \in N_{K_j} \cap N_{K_s}$. Очевидно, что величина $\Phi_{\tilde{\square}}(K_j, K_s)$ равна сумме величин $\Phi_{\square}(K)$, где размерность интервалов $N_{\square}$ и N_K равны K , а размерность их пересечения t изменяется от 0 до $k-1$.

Тогда

$$M_{n,k}^2 = \left\{ \sum_{t=0}^{k-1} 2^{2^k-2^t+1} + 1 \cdot 2^{n-2k+t} \cdot 2^{2^n-(2^{k+1}-2^t)} \cdot 2^{(2^t-2^k)i} \right\} \cdot \frac{1}{2^{2^n-1}} \leq \frac{2^{2^k}}{2^{2^{k+1}-2}} \left(1 - \frac{1}{2^{2^k}}\right)^{2^{n-2k}},$$

$$\text{и } D\xi_{n,k} = M\xi^2 - (M\xi)^2 \leq (D\xi_{n,k})^2 \cdot \frac{2^{2^k}k}{2^{2^n}}.$$

Применяя неравенства Чебышева, находим, что среди функций для которых $\tilde{\square} \in N_f$, доля тех, что удовлетворяют соотношению

$$|L_{i,k}(f) - \overline{L}_{\tilde{\square},k}(n)| \geq \frac{1}{\sqrt{C_n^k}} \cdot \overline{L}_{\tilde{\square},k}(n) \quad (4)$$

не превосходит $\frac{2^{2^k}k}{C_n^k}$ или число таких функций не превосходит

$$\frac{2^{2^k}}{2^{2^n-2^k} C_n^k} \quad (5)$$

Обозначим через $m^k(f)$ число наборов $\tilde{\square} \in N_f$, для которых выполняется (4), тогда $\overline{m}^k(n) = \frac{1}{2^{2^n}} \sum_{i=1}^{2^n} m^k(f_i) = \frac{1}{2^{2^n}} \sum_{\tilde{\square} \in E^n} \Phi_k(\tilde{\square}) \leq 2^{n+1} \cdot \frac{2^{2^k}k}{C_n^k}$, где $\Phi_k(\tilde{\square})$ число функций f_i , для которых

$\tilde{\square} \in f_i$ и выполнено соотношение (4). Полагая $\varepsilon = \frac{2}{k}$, получаем $m^k(f) < \frac{k^2 2^{2^k}}{C_n^k} \cdot 2^n$.

Положим $v = \frac{C_n^k-1}{2^{2^k-1}} \left(1 - \frac{1}{2^{2^k}}\right)^{n-k}$, тогда $|N_f / N_f^{\square}| \leq m^k(f) < \frac{k^2 2^{2^k}}{C_n^k} \cdot 2^n$. Обозначим через $\{S_n\}$ множество функций f , для которых $|S_1(\tilde{\square}, D_c)| \geq v$ для любого набора $\tilde{\square} \in N_f$ и размерность интервалов не превосходит

$$D = [\log \log n + \log \log \log n] + 1.$$

В силу предыдущих лемм [1,4,5,6,7,9] и соотношения (5) имеем $\log_n \square \frac{n}{2^{2^k}} = 1$.

Полагая $l = [\frac{m}{\square} \ln |N_f^{\square}| \frac{\square}{m}] + 1$, получим длину д.н.ф. после l -го шага не больше $(\frac{2^n}{\log n})(1+O(1))$ и мощность множества непокрытых наборов не более

$$\delta_l / N_f^{\square} + |N_f / N_f^{\square}| = O\left(\frac{2^n}{\log n}\right).$$

В конечном счете длина полученной в результате работы алгоритма д.н.ф. $L(f) < \frac{2^n}{\log n}$ для почти всех функций алгебры логики, что не более чем в $c \cdot \log \log n$ (c -константа) раз больше длины кратчайших д.н.ф. для почти всех булевых функций [2,8].

Заключение

Исследована проблема минимизации функций алгебры логики в классе д.н.ф. Разработаны алгоритмы полного перебора и ограниченной сложности поиска кратчайших д.н.ф. булевых функций и даны оценки их сложности. Предложен алгоритм минимизации функций алгебры логики на основе вычисления погрешности окрестности первого порядка элементарных конъюнкций д.н.ф.. Исследован локальный анализ в алгоритме направленного перебора и дана оценка его алгоритмической сложности.

Литература.

1. Яблонский С.В. Дискретная математика и математические вопросы кибернетики.- М,Наука,1974
2. Байжуманов А.А «Дискреттік математика» Шымкент 2007 жыл

3. Байжұманов А.А Арнайы кластағы сызықты емес логикалық теңдеулер жүйесін ықшамдау критеріі. «Педагогикалық кадрларды дайындаудың қазіргі заманғы өзекті мәселелері» Халықаралық ғылыми – практикалық конференция материалы. Шымкент, 2007ж.
4. Mendelson E. Introduction to mathematical logic. Fifth edition. – NY.: «Chapman&Hall/CRC» , 2010.
5. Гиндикин С.Г. Алгебра логики в задачах. - М.: «Наука», 1972.
6. Ершов Ю.Л., Палютин Е.А. Математическая логика. - М.: «Наука», 2011.
7. Игошин В.И. Математическая логика и теория алгоритмов. -М.: «Наука», 2008.
8. Клини С.К. Математическая логика. - М.: «Мир», 1973.
9. Яблонский С.В. Введение в дискретную математику. - М.: «Высшая школа», 2006.