Научная статья
УДК 519.115.4
DOI 10.35266/1999-7604-2023-2-87-91

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ МИНИМИЗАЦИИ ДИЗЪЮНКТИВНОЙ НОРМАЛЬНОЙ ФОРМЫ

Антон Георгиевич Назин
Сургутский государственный университет, Сургут, Россия
nazin_ag@surgu.ru

Аннотация. В статье изложен алгоритм, позволяющий осуществить направленный перебор для поиска минимальной дизъюнктивной нормальной формы с использованием лексикографического порядка.
Ключевые слова: простая импликанта, ДНФ, МДНФ, алгоритм, сокращенная ДНФ

Для цитирования: Назин А. Г. Модифицированный алгоритм минимизации дизъюнктивной нормальной формы // Вестник кибернетики. 2023. Т. 22, № 2. С. 87–91. DOI 10.35266/1999-7604-2023-2-87-91.

Original article

A MODIFIED ALGORITHM FOR MINIMIZING THE DISJUNCTIVE NORMAL FORM

Anton G. Nazin
Surgut State University, Surgut, Russia
nazin_ag@surgu.ru

Abstract. The article describes an algorithm for a directed search to find the minimum disjunctive normal form using the lexicographic order.
Keywords: prime implicant, DNF, MDNF, algorithm, reduced DNF

For citation: Nazin A. G. A modified algorithm for minimizing the disjunctive normal form. *Proceedings in Cybernetics*. 2023;22(2):87–91. DOI 10.35266/1999-7604-2023-2-87-91.

ВВЕДЕНИЕ

Задача минимизации дизъюнктивной нормальной формы (ДНФ), как известно [1], относится к классу NP-полных задач, поэтому для ее решения представляет интерес поиск алгоритма, который близок по сложности к полиномиальному.

Проблема минимизации ДНФ обсуждается в литературе [2]. Для ее решения известны алгоритмы двух типов. Это алгоритмы, решающие задачу точно, и алгоритмы, позволяющие найти ДНФ, которые близки к минимальной. Приближенные алгоритмы описаны в [3–5], а точные – в [6–9]. Описанный в статье алгоритм является модификацией алгоритма, предложенного автором в работе [10].

МАТЕРИАЛЫ И МЕТОДЫ

Пусть задана произвольная функция алгебры логики в виде сокращенной дизъюнктивной нормальной формы (СДНФ):

$$f(x_1 \dots x_n) = K_1 \vee K_2 \vee \dots \vee K_m.$$

Пусть с помощью известного метода Квайна – Мак-Класки [11] получена ее сокращенная ДНФ: $f^{(s)}(x_1 \dots x_n) = K_1^{(s)} \vee K_2^{(s)} \vee \dots \vee K_l^{(s)}$.

Обозначим через $N_f = \{\alpha_1, \dots, \alpha_m\} \subseteq E^n$ множество вершин гиперкуба, в которых $f(x_1 \dots x_n) = 1$. Составим матрицу T размерности l на m. Заполним матрицу T. Элемент t_{ij} положим равным значению i-й простой импликанты в j-й вершине:

$$t_{ij} = \begin{cases} t_{ij} = 1, \text{если } K_i^{(s)}(\alpha_j) = 1 \text{ в } j\text{-й вершине} \\ t_{ij} = 0, \text{если } K_i^{(s)}(\alpha_j) = 0 \text{ в } j\text{-й вершине} \end{cases}.$$

Рассмотрим произвольную вершину с номером j из N_f. Условие, что хотя бы одна простая импликанта в этой вершине обращается в 1, выглядит как:

$$\vee_{i=1}^{l} t_{ij} = 1. \qquad (1)$$

Обозначим $I_p = (i_1 \ldots i_p)$, $p \le l$ – набор номеров строк матрицы *T*, упорядоченный по возрастанию. Для всех вершин гиперкуба условие (1) примет вид:

$$\mathop{\wedge}_{i=1}^{m} \mathop{\vee}_{i \in I_p} t_{ij} = 1. \qquad (2)$$

Заметим, для того чтобы ДНФ, составленная из простых импликант $K_i^{(s)}$, где *i* пробегает по всем номерам из I_p, реализовала функцию *f*, то необходимо выполнение условия (2).

Сформулируем теперь известную задачу получения минимальной дизъюнктивной нормальной формы (МДНФ) из СДНФ [2]. Необходимо найти такой набор номеров $I_k \subseteq I_l$ строк матрицы *T*, для которого бы выполнялось: длина его должна быть минимальна и должно быть верно условие $\mathop{\wedge}_{j=1}^{m} \mathop{\vee}_{i \in I_k} t_{ij} = 1$. Тогда $\mathop{\vee}_{i \in I_k} K_i^{(s)}$ будет МДНФ для функции *f*.

Пусть $I_k \subseteq I_l$ – произвольный набор номеров строк матрицы *T* (набор номеров конъюнкций в СДНФ), для которого истинно условие (2). Обозначим через $M(I_k) = \left\{ I_{k-1}^{(1)}, \ldots, I_{k-1}^{(k)} \right\}$ – совокупность всех наборов номеров строк матрицы *T*, содержащихся в I_k и имеющих длину $k-1$. $M(I_k)$ лексикографически упорядочим по возрастанию: $I_{k-1}^{(1)} < I_{k-1}^{(2)} < \ldots < I_{k-1}^{(k)}$, где $I_{k-1}^{(1)}$ – наименьший, а $I_{k-1}^{(k)}$ – наибольший наборы. Верхний индекс обозначает порядковый номер набора в совокупности $M(I_k)$, а нижний – длину набора.

Определение.

ДНФ $\mathop{\vee}_{j \in I_k} K_j^{(s)}$ называется тупиковой ДНФ для функции *f*, если для любого набора $I_{k-1}^{(i)} \in M(I_k)$ верно условие $\mathop{\wedge}_{j=1}^{m} \mathop{\vee}_{i \in I_{k-1}^{(i)}} t_{ij} = 0$.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Обратимся к описанию алгоритма, позволяющего найти все МДНФ функции *f* по ее СДНФ.

Сперва опишем процедуру проверки ДНФ на тупиковость. Пусть I_l – набор, состоящий из всех номеров строк матрицы *T*, упорядоченных по возрастанию. Для него по построению истинно условие (4). Требуется проверить, является ли ДНФ $\mathop{\vee}_{i \in I_l} K_i^{(s)}$, соответствующая набору I_l, тупиковой. Рассмотрим $M(I_l) = \left\{ I_{l-1}^{(1)}, \ldots, I_{l-1}^{(l)} \right\}$ – лексикографически упорядоченную по возрастанию совокупность всех наборов $I_{l-1}^{(i)} \subset I_l, i = \overline{1, l}$. Согласно определению необходимо проверить условие $\mathop{\wedge}_{j=1}^{m} \mathop{\vee}_{i \in I_{l-1}^{(i)}} t_{ij} = 0$ для всех наборов из $M(I_l)$. Однако не всегда необходимо явно проверять условие $\mathop{\wedge}_{j=1}^{m} \mathop{\vee}_{i \in I_{l-1}^{(i)}} t_{ij} = 0$ для всех наборов из $M(I_l)$ и начинать проверку с наименьшего набора.

Рассмотрим ситуацию, когда явную проверку условия можно начинать не с младшего, а с некоторого другого набора из $M(I_l)$. Пусть для $q > 1$ наборов $I_{l-1}^{(1)} < I_{l-1}^{(2)} < \ldots < I_{l-1}^{(q)}$ из $M(I_l)$ истинно условие $\mathop{\wedge}_{j=1}^{m} \mathop{\vee}_{i \in I_{l-1}^{(i)}} t_{ij} = 0$. Обозначим совокупность этих наборов как $M^0(I_l)$. Пусть $I_{l-1}^{(q+1)} \in M(I_l)$ – первый по порядку набор, для которого условие ложно. Это означает, что ДНФ $\mathop{\vee}_{i \in I_l} K_i^{(s)}$ не тупиковая. Теперь проверим, является ли ДНФ $\mathop{\vee}_{i \in I_{l-1}^{(q+1)}} K_i^{(s)}$, соответствующая набору $I_{l-1}^{(q+1)}$, тупиковой. Рассмотрим $M(I_{l-1}^{(q+1)}) = \left\{ I_{l-2}^{(1)}, \ldots, I_{l-2}^{(q+1)}, \ldots, I_{l-2}^{(l-1)} \right\}$ – лексикографически упорядоченную по возрастанию совокупность всех наборов $I_{l-2}^{(i)} \subset I_{l-1}^{(q+1)}$. В этом случае можно начинать проверку с набора $I_{l-2}^{(q+1)}$. Покажем это, сформулировав сначала 2 вспомогательных утверждения.

Утверждение 1.

Любые два соседних набора $I_k^{(i)}$ и $I_k^{(i+1)}$ из одной совокупности $M(I_{k+1})$, $k+1 \le l$ содержат набор I_{k-1}, который стоит на i-м месте в совокупностях $M(I_k^{(i)})$ и $M(I_k^{(i+1)})$.

Утверждение 2.

Для любого $I_{k-1}^{(i)} \in M(I_k^{(j)}), 1 \le j \le i, i = \overline{1,k}$ справедливо: $I_{k-1}^{(i)}$ принадлежит совокупности $M(I_k^{(i+1)})$ и занимает в ней j-е место.

Покажем, что в совокупности $M(I_{l-1}^{(q+1)}) = \left\{I_{l-2}^{(1)}, ..., I_{l-2}^{(l-2)}\right\}$ существует непрерывная цепочка наборов $\left\{I_{l-2}^{(1)}, ..., I_{l-2}^{(q)}\right\}$ начиная с первого, для всех элементов которой истинно условие $\mathop{\wedge}\limits_{j=1}^{m} \mathop{\vee}\limits_{i \in I_{l-2}^{(i)}} t_{ij} = 0$. Для каждого $I_{l-1}^{(i)} \in M^0(I_l), i = \overline{1,q}$ существует совокупность $M(I_{l-1}^{(i)}) = \left\{I_{l-2}^{(1)}, ..., I_{l-2}^{(l-1)}\right\}, i = \overline{1,q}$. Для любого набора $I_{l-2}^{(i)}$, принадлежащего любой из совокупностей $M(I_{l-1}^{(1)}), i = \overline{1,q}$, верно условие $\mathop{\wedge}\limits_{j=1}^{m} \mathop{\vee}\limits_{i \in I_{l-2}^{(i)}} t_{ij} = 0$. Рассмотрим наборы $I_{l-1}^{(q)}$ и $I_{l-1}^{(q+1)}$. Напомним, что $\mathop{\wedge}\limits_{j=1}^{m} \mathop{\vee}\limits_{i \in I_{l-1}^{(q+1)}} t_{ij} = 1$, а $\mathop{\wedge}\limits_{j=1}^{m} \mathop{\vee}\limits_{i \in I_{l-1}^{(q)}} t_{ij} = 0$. Согласно утверждению 1 оба эти набора содержат набор $I_{l-2}^{(q)}$, который стоит на q-й позиции в совокупностях $M(I_{l-1}^{(q)})$ и $M(I_{l-1}^{(q+1)})$ соответственно. Для набора $I_{l-2}^{(q)}$ верно условие $\mathop{\wedge}\limits_{j=1}^{m} \mathop{\vee}\limits_{i \in I_{l-2}^{(q)}} t_{ij} = 0$. Осталось показать, что для всех наборов из $M(I_{l-1}^{(q+1)})$, младших набора $I_{l-2}^{(q)}$, также верно условие $\mathop{\wedge}\limits_{j=1}^{m} \mathop{\vee}\limits_{i \in I_{l-2}^{(i)}} t_{ij} = 0, i = \bar{1}, \overline{q-1}$. Согласно утверждению 2 любой набор $I_{l-2}^{(i)}, i = \bar{1}, \bar{q} < l-1$ содержится хотя бы в одной из совокупностей $M(I_{l-1}^{(i)}), i = \overline{1,q}$. Из этого следует, что для всех наборов из $M(I_{l-1}^{(q+1)})$, младших набора $I_{l-2}^{(q)}$, также верно условие $\mathop{\wedge}\limits_{j=1}^{m} \mathop{\vee}\limits_{i \in I_{l-2}^{(i)}} t_{ij} = 0, i = \overline{1,q}$.

Совокупность $M(I_{l-1}^{(q+1)})$ можно представить в виде $M(I_{l-1}^{(q+1)}) = \{I_{l-2}^{(1)}, I_{l-2}^{(2)}, ..., I_{l-2}^{(q)}, I_{l-2}^{(q+1)}, ..., I_{l-2}^{(l-1)}\}$, где каждый из наборов $I_{l-2}^{(i)}, i = \overline{1,q} < l-1$ содержится в совокупности $M(I_{l-1}^{(i)}), i = \overline{1,q}$. Следовательно, проверку можно начинать с набора $I_{l-2}^{(q+1)}$.

При получении $I_{l-2}^{(q+1)}$ из $I_{l-1}^{(q+1)}$ возможны два случая:

1. В наборе $I_{l-1}^{(q+1)}$ $i_1 = 0$. Начиная с первого номера находим в $I_{l-1}^{(q+1)}$ максимальную по длине цепочку номеров $I_r = i_1 i_2 ... i_r$, идущих по возрастанию, такую, что разность двух любых соседних номеров равна 1. Запишем $I_{l-1}^{(q+1)}$ в виде: $I_{l-1}^{(q+1)} = I_r\, i_{r+1} ... i_{l-1}$. Набор, который необходимо получить, обозначим $I_{l-2}^* = i_1^* i_2^* ... i_{l-2}^*$. Строим его следующим образом: первым $i_1^* i_2^* ... i_{r-1}^*$ элементам присваиваем $I_{r-1} = i_1 i_2 ... i_{r-1}$, далее $i_r^* = i_{r+1}, i_{r+1}^* = i_{r+2}, ..., i_{l-2}^* = i_{l-1}$.

2. В наборе $I_{l-1}^{(q+1)}$ $i_1 \ne 0$. Тогда ДНФ, соответствующая набору $I_{l-1}{}^{(q+1)}$, является тупиковой. Докажем это. Покажем сначала: $\forall I_{l-2} \subset I_{l-1}^{(q+1)}\ \exists\, I_{l-1}^{(\bar{q})} \supset I_{l-2} : I_{l-1}^{(\bar{q})} < I_{l-1}^{(q+1)}$.

Пусть $I_{l-2} = i_1 i_2 ... i_{l-2}$ – произвольный набор, содержащийся в $I_{l-1}^{(q+1)}$, очевидно, что в нем $i_1 \ne 0$. Тогда, добавив слева к $I_{l-2} = i_1 i_2 ... i_{l-2}\, i_1 = 0$, а остальные сдвинув вправо на единицу, получим $I_{l-1}^{(\bar{q})} < I_{l-1}^{(q+1)}$. Возможны только два случая:

1. $\mathop{\vee}\limits_{i \in I_{l-1}^{(\bar{q})}} K^{(s)_i}$ – тупиковая, тогда $\mathop{\wedge}\limits_{j=1}^{m} \mathop{\vee}\limits_{i \in I_{l-2}} t_{ij} = 0$ по определению.

2. $\mathop{\wedge}\limits_{j=1}^{m} \mathop{\vee}\limits_{i \in I_{l-1}} t_{ij} = 0$ очевидно $\mathop{\wedge}\limits_{j=1}^{m} \mathop{\vee}\limits_{i \in I_{l-2}} t_{ij} = 0$.

Итак, первым шагом проверки ДНФ на тупиковость является получение из текущего набора такого, с которого необходимо начать явную проверку условия.

Рассмотрим случай 1, т. е. в наборе $I_{l-1}^{(q+1)}$ $i_1 = 0$. Получаем, как описано выше, набор $I_{l-2}^{(q+1)}$. Если для какого-либо набора

$I_{l-2}^{(i)} \in \left\{ I_{l-2}^{(q+1}, ..., I_{l-2}^{(l-1)} \right\}$ выполняется условие $\overset{m}{\underset{j=1}{\wedge}} \underset{i \in I_{l-2}{}^{(i)}}{\vee} t_{ij} = 1$, выходим из процедуры. ДНФ, соответствующая набору $I_{l-1}^{(q+1)}$, не является тупиковой. Иначе проверяем условие для следующего набора из $\left\{ I_{l-2}^{(q+1)}, ..., I_{l-2}^{(l-1)} \right\}$. Если для всех наборов из $\left\{ I_{l-2}^{(q+1)}, ..., I_{l-2}^{(l-1)} \right\}$ условие $\overset{m}{\underset{j=1}{\wedge}} \underset{i \in I_{l-2}{}^{(i)}}{\vee} t_{ij} = 0$ истинно, ДНФ, соответствующая набору $I_{l-1}^{(q+1)}$, является тупиковой.

Рассмотрим случай 2, т. е. в наборе $I_{l-1}^{(q+1)}$ $i_1 \neq 0$. Как показано выше, при этом ДНФ, соответствующая набору $I_{l-1}^{(q+1)}$, является тупиковой.

Рассмотрим алгоритм получения всех МДНФ по шагам.

Пусть I_l – набор, состоящий из всех номеров строк матрицы *T*. Положим $k = 0$.

Шаг 1.

Вычисляем выражение:

$$\overset{m}{\underset{j=1}{\wedge}} \underset{i \in I_{l-k}{}^{(q)}}{\vee} t_{ij}. \tag{3}$$

Если (3) равно 1, переходим к шагу 2, если (3) равно 0, переходим к шагу 3.

Шаг 2.

Проверяем, является ли ДНФ $\underset{i \in I_{l-k}}{\vee} K^{(s)}$, соответствующая набору $I_{l-k}^{(q)}$, тупиковой. Если $\underset{i \in I_{l-k}{}^{(q)}}{\vee} K_i^{(s)}$ – тупиковая, переходим к шагу 4, если $\underset{i \in I_{l-k}{}^{(q)}}{\vee} K_i^{(s)}$ – не тупиковая, нашелся набор $I_{l-k}^{(\bar{q})}, k = k + 1$, для которого (5) равно 1. Переходим к шагу 5.

Шаг 3.

Если $I_{l-k}^{(q)}$ соответствует исходной матрице $(k = 0)$ или $I_{l-k}^{(q)}$ является наибольшим набором размерности $l - k$, алгоритм заканчивает работу. Если нет, получаем набор $I_{l-k}^{(q+1)}$, лексикографически следующий за набором $I_{l-k}^{(q)}$. Переходим к шагу 1.

Шаг 4.

Присваиваем ДНФ $\underset{i \in I_{l-k}{}^{(q)}}{\vee} K_i^{(s)}$ метку Δ. Переходим к шагу 3.

Шаг 5.

Если у ДНФ $\underset{i \in I_{l-k}{}^{(q)}}{\vee} K_i^{(s)}$ есть метка Δ, удаляем ее. Переходим к шагу 2.

Все ДНФ с меткой Δ будут МДНФ.

ЗАКЛЮЧЕНИЕ

Описанный алгоритм был реализован в виде программы, проведены предварительные расчеты. В дальнейшем планируется проведение исследования по определению зависимости времени работы алгоритма от размерности матрицы.

Список источников

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи / пер. с англ. Е. В. Левнера, М. А. Фрумкина. М. : Мир, 1982. 416 с.
2. Андреев А. Е. К проблеме минимизации дизъюнктивных нормальных форм // Доклады Академии наук СССР. 1984. Т. 274, № 2. С. 265–269.
3. McGeer P., Sanghavi J., Brayton R. et al. ESPRESSO-SIGNATURE: A new exact minimizer for logic functions. In: *30th ACM/IEEE Design Automation Conference*, Dallas, TX, USA. 1993. p. 618–624.
4. Hong S. J., Cain R. G., Ostapko D. L. MINI: A heueristic approach for logic minimization. *IBM Journal Research and Development.* 1974;18(5):443–458.
5. Hlavička J., Fišer P. BOOM – a Heuristic Boolean Minimizer. In: *IEEE/ACM International Conference on Computer Aided Design ICCAD-2001,* San Jose, California (USA), November 4–8, 2001. p. 439–442.

References

1. Garey M. R., Johnson D. S. Computers and intractability. Levner E. V., Frumkin M. A., translators. Moscow: Mir; 1982. 416 p. (In Russian).
2. Andreev A. E. On the problem of minimization of disjunctive normal forms. *Doklady Akademii nauk SSSR.* 1984;274(2):265–269. (In Russian).
3. McGeer P., Sanghavi J., Brayton R. et al. ESPRESSO-SIGNATURE: A new exact minimizer for logic functions. In: *30th ACM/IEEE Design Automation Conference,* Dallas, TX, USA. 1993. p. 618–624.
4. Hong S. J., Cain R. G., Ostapko D. L. MINI: A heueristic approach for logic minimization. *IBM Journal Research and Development.* 1974;18(5):443–458.
5. Hlavička J., Fišer P. BOOM – a Heuristic Boolean Minimizer. In: *IEEE/ACM International Conference on Computer Aided Design ICCAD-2001,* San Jose, California (USA), November 4–8, 2001. p. 439–442.

6. Brayton R. K., Hachtel G. D., McMullen C. et al. Logic minimization algorithms for VLSI synthesis. Hingham, MA: Kluwer Academic Publishers; 1984. 194 p.
7. Hachtel G. D., Somenzi F. Logic synthesis and verification algorithms. Boston, MA: Kluwer Academic Publishers; 1996. 564 p.
8. Журавлев Ю. И. Алгоритмы построения минимальных дизъюнктивных нормальных форм для функций алгебры логики // Дискретная математика и математические вопросы кибернетики / под общ. ред. С. В. Яблонского, О. Б. Лупанова. Т. 1. М. : Наука, 1974. 312 с.
9. Стрыгин В. З. Полиномиальный алгоритм минимизации случайных («почти всех») булевых функций. М. : ЦАГИ, 1992. 8 с.
10. Назин А. Г. Алгоритм получения всех минимальных дизъюнктивных нормальных форм из сокращенной дизъюнктивной нормальной формы с использованием лексикографического порядка // Исследовано в России. 2002. С. 942–947. URL: https://cyberleninka.ru/article/n/algoritm-polucheniya-vseh-minimalnyh-dizyunk-tivnyh-normalnyh-form-iz-sokraschyonnoy-dizyunk-tivnoy-normalnoy-formy-s-ispolzovaniem (дата обращения: 01.03.2023).
11. Колдуэлл С. Х. Логический синтез релейных устройств / пер. с англ. Г. К. Москатова, А. Д. Таланцева ; под ред. М. А. Гаврилова. М. : Изд-во иностр. лит., 1962. 740 с.

6. Brayton R. K., Hachtel G. D., McMullen C. et al. Logic minimization algorithms for VLSI synthesis. Hingham, MA: Kluwer Academic Publishers; 1984. 194 p.
7. Hachtel G. D., Somenzi F. Logic synthesis and verification algorithms. Boston, MA: Kluwer Academic Publishers; 1996. 564 p.
8. Zhuravlev Yu. I. Algoritmy postroeniia minimalnykh diziunktivnykh normalnykh form dlia funktsii algebry logiki. In: Yablonsky S. V., Lupanov O. B., editors. Diskretnaia matematika i matematicheskie voprosy kibernetiki. Vol. 1. Moscow: Nauka; 1974. 312 p. (In Russian).
9. Strygin V. Z. Polinomialnyi algoritm minimizatsii sluchainykh (“pochti vsekh”) bulevykh funktsii. Moscow: Central Aerohydrodynamics Institute; 1992. 8 p. (In Russian).
10. Nazin A. G. Algoritm polucheniia vsekh minimalnykh diziunktivnykh normalnykh form iz sokrashchennoi diziunktivnoi normalnoi formy s ispolzovaniem leksikograficheskogo poriadka. *Investigated in Russia.* 2002:942–947. URL: https://cyberleninka.ru/article/n/algoritm-polucheniya-vseh-minimalnyh-dizyunk-tivnyh-normalnyh-form-iz-sokraschyonnoy-dizyunk-tivnoy-normalnoy-formy-s-ispolzovaniem (accessed: 01.03.2023). (In Russian).
11. Caldwell S. H. Switching circuits and logical design. Moskatov G. K., Talantsev A. D., translators; Gavrilov M. A., editor. Moscow: Izd-vo inostr. lit.; 1962. 740 p. (In Russian).

Информация об авторе

А. Г. Назин – кандидат физико-математических наук, доцент.

Information about the author

A. G. Nazin – Candidate of Sciences (Physics and Mathematics), Docent.