

В. Т. Тозик

АЛГЕБРАИЧЕСКИЙ МЕТОД ОПРЕДЕЛЕНИЯ ПОЛНОГО МНОЖЕСТВА ПРОСТЫХ РАЗРЕЗОВ В ДВУХПОЛЮСНЫХ СЕТЯХ

Рассматривается задача поиска простых разрезов в двухполюсных структурно-сложных сетях. В основу предлагаемого метода положена алгебраическая модель сети, базирующаяся на алгебре кубических комплексов. Это позволяет предложить эффективную с точки зрения трудоемкости процедуру определения полного множества простых разрезов.

Ключевые слова: двухполюсная сеть, простой разрез, структурная функция, алгебра кубических комплексов.

Задача определения полного множества простых разрезов в двухполюсных сетях нетривиальна. Конструктивный метод ее решения предложен только для плоских графов путем построения полного множества циклов в соответствующих двойственных графах. Однако для достаточно больших графов с нетривиальной структурой (не сводимой к плоской) задача становится непреодолимо сложной ввиду комбинаторных трудностей полного перебора.

В основу предлагаемого в настоящей работе метода положена алгебраическая модель графа, использующая введенную алгебру простых цепей [1] и алгебру кубических комплексов [2], что позволяет предложить достаточно эффективную с точки зрения трудоемкости процедуру определения простых разрезов.

Как было показано в работе [1], структура двухполюсной сети (α, β) может быть представлена булевой структурной функцией (СФ) в дизъюнктивной нормальной форме (ДНФ) с помощью простых цепей (элементарных конъюнкций K_j ранга r):

$$f_{\alpha\beta} = \bigvee_{j=1}^m K_j = \bigvee_{j=1}^m \left[\bigwedge_{i=1}^r \lambda x_i \right], \quad (1)$$

где m — число простых цепей сети.

Ту же функцию (1) можно представить в конъюнктивной нормальной форме (КНФ) с помощью простых разрезов (элементарных дизъюнкций D_j ранга s)

$$f_{\alpha\beta} = \bigwedge_{j=1}^t D_j = \bigwedge_{j=1}^t \left[\bigvee_{i=1}^s x_i \right], \quad (2)$$

где t — число простых разрезов двухполюсной сети.

В работе [1] представлен алгоритм, позволяющий определить кубическое покрытие $\Pi(L_1)$, соответствующее ДНФ булевой СФ, записанной в виде простых цепей (1). Теперь рассмотрим выражение (2). С помощью правила де Моргана [2] можно перейти к дизъюнктивной форме отрицания булевой СФ:

$$\bar{f}_{\alpha\beta} = \bigvee_{j=1}^t \bar{D}_j, \quad (3)$$

$$\bar{D}_j = \overline{x_1 \vee x_2 \vee x_2 \vee \dots \vee x_s} = \bar{x}_1 \bar{x}_2 \dots \bar{x}_s. \quad (4)$$

Отрицанию булевой СФ (3) соответствует покрытие $\Pi(L_0)$ множества вершин n -мерного куба L_0 , на которых данная функция принимает нулевые значения, причем каждой конъюнкции \bar{D}_j ранга s (4) соответствует некоторый куб $C_j \in \Pi(L_0)$. Целью настоящей работы является создание метода определения полного множества таких кубов $\bigcup_{j=1}^t C_j$, каждый

куб C_j которого соответствует конъюнкции \bar{D}_j (4), поставленной в соответствие простому разрезу. Приведем теоретико-множественное представление этой двойственной задачи:

L_1 — подмножество состояний связности сети ($f_{\alpha\beta} = 1$):

$$L_1 \subseteq \Pi(L_1) = \bigcup_{j=1}^m C_j \Rightarrow f_{\alpha\beta} = \bigvee_{j=1}^m K_j,$$

K_j — конъюнкция, поставленная в соответствие простой цепи;

L_0 — подмножество состояний несвязности сети ($f_{\alpha\beta} = 0$):

$$L_0 \subseteq \Pi(L_0) = \bigcup_{j=1}^t C_j \Rightarrow \bar{f}_{\alpha\beta} = \bigvee_{j=1}^t \bar{D}_j,$$

\bar{D}_j — конъюнкция, поставленная в соответствие простому разрезу.

Введем некоторые определения, основанные на положениях работы [2].

Определение 1. Куб $C = (a_1, a_2, \dots, a_n)$ размерности n есть n -мерный вектор, каждая координата которого a_i принимает значения из множества $\{0, 1, X\}$. Координаты $a_i \in \{0, 1\}$ называются связанными, $a_i = X$ — свободными.

Определение 2. Кубы $C_j = (a_1, a_2, \dots, a_n)$ и $C_s = (b_1, b_2, \dots, b_n)$ равны между собой $C_j = C_s$, если $\forall_i (a_i = b_i)$.

Определение 3. Кубы $C_j = (a_1, a_2, \dots, a_n)$ и $C_s = (b_1, b_2, \dots, b_n)$ находятся в отношении строгого включения $C_j \subset C_s$, если $\exists_i (a_i \in \{0, 1\} \& b_i = X)$ и не $\exists_i (a_i = X \& b_i \in \{0, 1\} \vee a_i = \bar{b}_i)$.

Определение 4. Кубы C_j и C_s находятся в отношении нестрогого включения $C_j \subseteq C_s$, если $C_j = C_s$ или $C_j \subset C_s$.

Определение 5. Множество кубов Π и множество вершин L_i находятся в отношении нестрогого включения $L_i \subseteq \Pi$, если любая вершина $l \in L_i$ включена в некоторый куб $C \in \Pi$, т.е. $l \subseteq C$. В дальнейшем будем говорить, что множество Π покрывает множество L_i , если $L_i \subseteq \Pi$.

Определение 6. Кубы C_j и C_s несравнимы ($C_j \sim C_s$), если $C_j \not\subseteq C_s$ и $C_s \not\subseteq C_j$.

Ниже дается определение операции объединения, которое отличается от приведенного в работе [2]. Чтобы различать эти две операции, будем обозначать определенную ниже операцию символом \bigcup^+ .

Определение 7. Результат операции объединения двух кубов C_j и C_s определяется как

$$C_j \bigcup^+ C_s = \begin{cases} C_j, & \text{если } C_s \subseteq C_j; \\ C_s, & \text{если } C_j \subseteq C_s; \\ \{C_s, C_j\}, & \text{если } C_j \sim C_s. \end{cases}$$

Операция объединения \bigcup^+ обладает свойством коммутативности и ассоциативности. Результат операции объединения двух множеств кубов Π_d и Π_e определяется как множество $\Pi = \Pi_d \bigcup^+ \Pi_e$, полученное объединением Π_d и Π_e в обычном теоретико-множественном смысле с последующим удалением кубов C_j таких, что $C_j \subseteq C_s$, или C_s таких, что $C_s \subseteq C_j$; $C_j, C_s \in \Pi$.

Объединение множества Π с самим собой приводит к множеству, в котором каждая пара кубов несравнима. В дальнейшем, чтобы отобразить это свойство, будем писать: $\Pi = \bigcup^+ \Pi$.

Определение 8. Результат $\#$ -операции двух кубов $C_j = (a_1, a_2, \dots, a_n)$ и $C_s = (b_1, b_2, \dots, b_n)$ определяется как

$$C_j \# C_s = \begin{cases} C_j, & \text{если } \exists_i (a_i = \bar{b}_i); \\ \emptyset, & \text{если } C_j \subseteq C_s; \\ \{C_1, C_2, \dots, C_t, \dots, C_k\}, & \text{причем каждой паре координат } \forall_{i=1}^n (a_i, b_i) \text{ таких, что} \\ & a_i = X, b_i \neq X, \text{ соответствует куб } C_t = (a_1, a_2, \dots, a_{i-1}, \bar{b}_i, a_{i+1}, \dots, a_{n-1}, a_n). \end{cases}$$

$\#$ -операция некоммутативна и неассоциативна. Для множества кубов и одного куба, а также для двух множеств кубов $\Pi_d = \{C_1^d, C_2^d, \dots, C_m^d\}$ и $\Pi_e = \{C_1^e, C_2^e, \dots, C_k^e\}$ $\#$ -операция определяется следующим образом:

$$\begin{aligned} \Pi_d \# C_j &= \{C_1^d \# C_j, C_2^d \# C_j, C_j, \dots, C_m^d \# C_j\}; \\ \Pi_d \# \Pi_e &= \left(\left(\dots \left(\left(\Pi_d \# C_1^e \right) \# C_2^e \right) \# \dots \right) \# C_k^e \right). \end{aligned}$$

Определение 9. Максимальным для заданного множества кубов Π_d называется такой куб C_j , что $C_j \# \Pi_d = \emptyset$, и при замене в C_j хотя бы одной связанной координаты на свободную $C_j \# \Pi_d \neq \emptyset$.

Множество, содержащее все максимальные для Π_d кубы, будем обозначать $\max(\Pi_d)$. Легко показать, что для Π_d множество $\max(\Pi_d)$ единственное.

Определение 10. Простым кубом C_j для булевой СФ называется такой, что $C_j \in \max(\Pi(L_1))$ либо $C_j \in \max(\Pi(L_0))$. Иными словами, простым кубом называется такой куб C_j , для которого в $\Pi(L_1)$ или $\Pi(L_0)$ не существует C_s такой, что $C_j \subseteq C_s$.

В дальнейшем множества простых кубов будем обозначать Z .

Можно показать, что

$$\bigcup^+ (I \# \Pi) = \max(I \# \Pi), \quad (5)$$

где I — куб размерности n , в котором все компоненты свободные.

Поиск простых разрезов может быть осуществлен в два этапа. Вначале определяется полное множество простых цепей, а затем с помощью введенных выше операций и отношений алгебры кубов определяется полное множество простых разрезов.

В работе [1] было предложено осуществлять решение первой задачи на n -мерном кубическом комплексе S^n , это позволило найти покрытие булевой СФ в виде множества кубов $\Pi(L_1) = \{C_1, C_2, \dots, C_m\}$, соответствующих простым цепям.

Отрицанию булевой СФ (3) соответствует покрытие $\Pi(L_0)$ множества вершин n -мерного куба L_0 , на которых данная функция принимает нулевые значения, причем каждой конъюнкции \bar{D}_j (4) соответствует $C_j \in \Pi(L_0)$.

Утверждение 1. Если куб $C_j \in \Pi(L_0)$ соответствует конъюнкции \bar{D}_j (4), поставленной в соответствие j -му простому разрезу, то $C_j \in Z(L_0)$, т.е. C_j является простым кубом для L_0 .

Доказательство. Каждый простой разрез отличается от другого по крайней мере одним элементом, поэтому любые две конъюнкции \bar{D}_j и \bar{D}_s , поставленные в соответствие j -му и s -му простым разрезам, отличаются друг от друга по крайней мере одной буквой. В соответствии с этим кубы C_j и C_s несравнимы ($C_j \sim C_s$), т.е. $C_j \not\subseteq C_s$ и $C_s \not\subseteq C_j$ для $C_j, C_s \in \Pi(L_0)$. Поскольку в каждую конъюнкцию \bar{D}_j буквы входят только в инверсном виде и в силу того, что простой разрез является минимальным по включению множеством элементов, для любого куба $C_j \in \Pi(L_0)$ с ценой R_j не существует куба $C_t \in \Pi(L_0)$ с ценой $R_t \leq R_j - 1$ такого, что $C_j \subset C_t$. Таким образом, все кубы $C_j \in \Pi(L_0)$ являются простыми.

Утверждение 2. Множество простых кубов $Z(L_0)$ является покрытием $\Pi(L_0)$ множества вершин n -мерного куба L_0 , каждый куб $C_j \in \Pi(L_0)$ которого соответствует конъюнкции \bar{D}_j , записанной с помощью простых разрезов.

Доказательство. Предположим обратное. Без потери общности можно допустить, что существует простой куб $C_j \in Z(L_0)$ такой, что $C_j \notin \Pi(L_0)$. В соответствии с утверждением 1 все кубы $C_j \in \Pi(L_0)$ являются простыми. Отсюда $\Pi(L_0) \subseteq Z(L_0)$ и можно записать

$$Z(L_0) = \{\Pi(L_0), C_j\}. \quad (6)$$

Так как оба множества простых кубов $Z(L_0)$ и $\Pi(L_0)$ покрывают одно и то же множество вершин L_0 , и только L_0 , то

$$Z(L_0) \equiv \Pi(L_0). \quad (7)$$

Поскольку множество простых кубов для L_0 единственно, то из (6) и (7) следует, что либо $C_j \notin Z(L_0)$, либо $C_j \in \Pi(L_0)$. Полученное противоречие доказывает данное утверждение.

На основании вышеизложенного можно предложить алгоритм определения множества кубов $\Pi(L_0)$, соответствующих множеству простых разрезов.

Алгоритм

1) определить множество кубов $\Pi(L_1)$, соответствующих множеству простых цепей, с помощью алгоритма, предложенного в [1];

2) тогда $\Pi(L_0) = \bigcup^+ [I \# \Pi(L_1)]$;

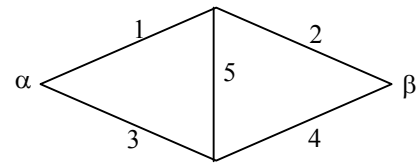
3) конец.

Утверждение 3. В результате алгоритма получается множество кубов $\Pi(L_0)$, соответствующее полному множеству простых разрезов.

Доказательство. Для полностью определенной булевой СФ множество простых кубов $Z(L_0)$ совпадает с множеством максимальных кубов для L_0 . Поэтому из (5) следует $Z(L_0) = \bigcup^+ [I \# \Pi(L_1)]$, а из утверждения 2 — $\Pi(L_0) = Z(L_0)$. Утверждение доказано.

Замечание. Выполнение \bigcup^+ -операции можно осуществлять как после завершения $[I \# \Pi(L_1)]$, так и после каждого $\#$ -вычитания куба $C_j \in \Pi(L_1)$, проводимого на i -м шаге алгоритма. В приведенном ниже примере используется вторая модификация выполнения \bigcup^+ -операции.

Пример. Рассмотрим предложенный метод на примере сети, представленной на рисунке. Применение алгоритма определения полного множества простых цепей к данной сети разобрано в [1]. Поэтому рассмотрим только работу алгоритма определения полного множества простых разрезов, считая заданным исходное множество простых цепей и соответствующих им кубов $\Pi(L_1)$. Для упрощения примера предположим существование только реберных разрезов



$$\Pi(L_1) = \Pi_1 = \begin{cases} X & X & 1 & 1 & X \\ 1 & 1 & X & X & X \\ 1 & X & X & 1 & 1 \\ X & 1 & 1 & X & 1 \end{cases} \begin{matrix} C_1^1 \\ C_2^1 \\ C_3^1 \\ C_4^1 \end{matrix},$$

$$I = (X \ X \ X \ X \ X).$$

В соответствии со свойствами $\#$ -операции

$$\Pi_2 = I \# C_1^1 = \begin{cases} X & X & 0 & X & X \\ X & X & X & 0 & X \end{cases} \begin{matrix} C_1^2 \\ C_2^2 \end{matrix},$$

$$\Pi_3 = \Pi_2 \# C_2^1 = \begin{cases} 0 & X & 0 & X & X \\ X & 0 & 0 & X & X \\ 0 & X & X & 0 & X \\ X & 0 & X & 0 & X \end{cases} \left. \begin{matrix} C_1^3 \\ C_2^3 \\ C_3^3 \\ C_4^3 \end{matrix} \right\} \begin{matrix} C_1^2 \# C_2^1 \\ C_2^2 \# C_2^1 \end{matrix},$$

$$\Pi_4 = \Pi_3 \# C_3^1 = \begin{cases} 0 & X & 0 & X & X \\ 0 & 0 & 0 & X & X \\ X & 0 & 0 & 0 & X \\ X & 0 & 0 & X & 0 \\ 0 & X & X & 0 & X \\ X & 0 & X & 0 & X \end{cases} \left. \begin{matrix} C_1^4 = C_1^3 \# C_3^1 = C_1^3 \\ C_2^4 \\ C_3^4 \\ C_4^4 \\ C_5^4 = C_3^3 \\ C_6^4 = C_4^3 \end{matrix} \right\} C_2^3 \# C_3^1,$$

$$\Pi_5 = \bigcup^+ \Pi_4 = \left\{ \begin{array}{ccccc} 0 & X & 0 & X & X \\ X & 0 & 0 & X & 0 \\ 0 & X & X & 0 & X \\ X & 0 & X & 0 & X \end{array} \right\} \begin{array}{l} C_1^5 = C_1^4 \supseteq C_2^4 \\ C_2^5 = C_4^4 \\ C_3^5 = C_5^4 \\ C_4^5 = C_6^4 \supseteq C_3^4 \end{array},$$

$$\Pi_6 = \Pi_5 \# C_4^1 = \left\{ \begin{array}{ccccc} 0 & X & 0 & X & X \\ X & 0 & 0 & X & 0 \\ 0 & 0 & X & 0 & X \\ 0 & X & 0 & 0 & X \\ 0 & X & X & 0 & 0 \\ X & 0 & X & 0 & X \end{array} \right\} \begin{array}{l} C_1^6 = C_1^5 \\ C_2^6 = C_2^5 \\ C_3^6 \\ C_4^6 \\ C_5^6 \\ C_6^6 = C_4^5 \end{array} \left. \vphantom{\begin{array}{l} C_1^6 \\ C_2^6 \\ C_3^6 \\ C_4^6 \\ C_5^6 \\ C_6^6 \end{array}} \right\} C_3^5 \# C_4^1,$$

$$\Pi(L_0) = \Pi_7 = \bigcup^+ \Pi_6 = \left\{ \begin{array}{ccccc} 0 & X & 0 & X & X \\ X & 0 & 0 & X & 0 \\ 0 & X & X & 0 & 0 \\ X & 0 & X & 0 & X \end{array} \right\} \begin{array}{l} C_1^7 = C_1^6 \supseteq C_4^6 \\ C_2^7 = C_2^6 \\ C_3^7 = C_5^6 \\ C_4^7 = C_6^6 \supseteq C_3^6 \end{array}.$$

Покрытие Π_7 представляет собой множество кубов $\Pi(L_0)$, соответствующее полному множеству простых разрезов: $\{(1,3), (2,3,5), (1,4,5), (2,4)\}$.

Оценка трудоемкости метода. Как правило, наиболее эффективными являются алгоритмы с трудоемкостью, степенной относительно размерности задачи. Понятие „степенного“ алгоритма близко к принятому в зарубежной литературе определению „хорошего“ алгоритма. Степенная оценка наглядно поясняется с позиций программирования. Линейную оценку имеют алгоритмы, просматривающие информацию единственный раз. Квадратичная оценка связана с „циклом в цикле“. Дальнейший рост порядка оценки соответствует наличию в алгоритме более длинных цепочек вложенных друг в друга циклов.

Другой класс составляют переборные алгоритмы, связанные с просмотром возможных ситуаций — „кандидатов в ответ“ задачи. Обычно число ситуаций экспоненциально возрастает относительно размерности задачи. Тем самым экспоненциальная (а тем более факториальная) трудоемкость переборного алгоритма растет быстрее, чем любая степенная функция.

В работе [1] доказана теоретическая эффективность алгоритма определения простых цепей. Оценка трудоемкости для него, измеряемая числом Δ -операций, ниже квадратичной относительно размерности задачи, представляемой числом простых цепей сети.

Трудоемкость алгоритма определения простых разрезов оценим по числу $\#$ -операций и операций сравнения кубов при выполнении \bigcup^+ -операции. Число простых цепей будем по-прежнему обозначать буквой m , а число простых разрезов буквой t . Размерность задачи определяется числом простых разрезов в сети.

Число $\#$ -операций может быть оценено как произведение числа кубов m в исходном покрытии $\Pi(L_1)$ на среднее число кубов в покрытии Π_i при выполнении $(\Pi_i \# C_j^1)$. Среднее число кубов в Π_i может быть принято равным t , поскольку после выполнения

$\Pi_{i+1} = \bigcup^+ (\Pi_i \# C_j^1)$ число кубов в Π_{i+1} приблизительно равно t . Таким образом, число #-операций при выполнении алгоритма приблизительно равно mt .

Число попарных операций сравнения при выполнении $\bigcup^+ \Pi_i$ равно $g(g-1)/2$, где g — число кубов в Π_i . Приняв поправочный коэффициент, учитывающий превышение числа кубов в $(\Pi_i \# C_j^1)$ над $\bigcup^+ (\Pi_i \# C_j^1)$, равным в среднем k , можно записать, что число сравнений при выполнении \bigcup^+ -операции равно $kt(kt-1)/2 \approx (kt)^2/2$. Таким образом, общая трудоемкость алгоритма может быть оценена как $T \cong mt + (kt)^2/2$, т.е. трудоемкость алгоритма пропорциональна квадрату размерности задачи. Таким образом, можно сделать вывод об эффективности предложенного метода определения полного множества простых разрезов в двухполюсных сетях. Представление структуры сети в виде кубов кубического комплекса позволяет возможность предложить чрезвычайно эффективные методы поиска простых цепей и разрезов, поскольку множества кубов хорошо представляются в памяти компьютера массивами двоичных кодов, а операции над ними — соответствующими программными средствами. Предложенные методы алгоритмически просты и не накладывают никаких ограничений на структуру исследуемых сетей.

СПИСОК ЛИТЕРАТУРЫ

1. Тозик В. Т. Математический аппарат для анализа структурных свойств сетей // Изв. вузов. Приборостроение. 2010. Т. 53, № 12. С. 22—30.
2. Миллер Р. Теория переключаемых схем. Т.1. М.: Наука, 1970. 416 с.

Сведения об авторе

Вячеслав Трофимович Тозик

— канд. техн. наук, доцент; Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кафедра инженерной и компьютерной графики; заведующий кафедрой;
E-mail: tozik@mail.ifmo.ru

Рекомендована кафедрой инженерной и компьютерной графики

Поступила в редакцию 11. 02.10 г.