ИНФОРМАТИКА

2013 октябрь-декабрь № 4

УДК 519.7

С.Х. Тагави Афшорд, Ю.В. Поттосин

ИССЛЕДОВАНИЕ СВОЙСТВ РАЗЛОЖИМОСТИ СИСТЕМ БУЛЕВЫХ ФУНКЦИЙ

Описывается компьютерная программа, анализирующая разложимость системы булевых функций и отыскивающая подходящее для декомпозиции разбиение множества аргументов. Представлены три задачи, связанные с двухблочной разделительной декомпозицией системы булевых функций: получение всех решений декомпозиции, поиск лучшего решения с точки зрения размера схемы и нахождение быстрого решения в случае разложимости системы. Приводятся результаты компьютерного эксперимента по определению разложимости систем булевых функций.

Введение

Задача декомпозиции булевых функций является одной из важных и сложных задач из области логического проектирования, успешное решение которой непосредственно влияет на качество и стоимость проектируемых цифровых устройств. Декомпозиция системы булевых функций, описывающей поведение некоторого дискретного устройства, ведет к разбиению его на отдельные блоки, что облегчает дальнейшую процедуру логического синтеза. Как показано в работах [1, 2], данной задаче посвящено значительное количество статей, однако вопрос еще требует исследований [3, 4]. В настоящей статье рассматривается задача декомпозиции системы булевых функций в следующей постановке. Задана система полностью определенных булевых функций $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, где $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$. Требуется найти суперпозицию $\boldsymbol{y} = \boldsymbol{\varphi}(\boldsymbol{w}, \boldsymbol{z}_2)$, $\boldsymbol{w} = \boldsymbol{g}(\boldsymbol{z}_1)$, где $\boldsymbol{z}_1$ и $\boldsymbol{z}_2$ – векторные переменные, компонентами которых служат соответственно переменные из подмножеств Z_1 и Z_2, образующих разбиение множества $X = \{x_1, x_2, \ldots, x_n\}$. При этом число компонент векторной переменной w должно быть меньше, чем у $\boldsymbol{z}_1$. Такой вид декомпозиции назван *двухблочной разделительной декомпозицией* [5]. Обычно переменные из Z_1 называют связанными, а переменные из Z_2 – свободными. В подавляющем большинстве публикаций, рассматривающих данную задачу, подмножества Z_1 и Z_2 считаются заданными. Вопросу поиска разбиения $\{Z_1, Z_2\}$, при котором данная задача имеет решение, посвящено не так много публикаций. Среди работ, где рассматривается данный вопрос, можно назвать [3, 6–10]. Ниже предлагается метод поиска разбиения $\{Z_1, Z_2\}$ множества аргументов X, который основан на использовании аппарата покрытий троичной матрицы [11].

1. Постановка задачи

Пусть система полностью определенных булевых функций $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ задана матрицами $\boldsymbol{U}$, $\boldsymbol{V}$, которые являются матричным представлением системы дизъюнктивных нормальных форм (ДНФ) заданных функций [5]. Матрица $\boldsymbol{U}$ является троичной матрицей размерности $l \times n$, где l – число элементарных конъюнкций в заданных ДНФ. Столбцы матрицы $\boldsymbol{U}$ помечены переменными $x_1, x_2, \ldots, x_n$, а строки представляют упомянутые элементарные конъюнкции (интервалы пространства переменных $x_1, x_2, \ldots, x_n$). Матрица $\boldsymbol{V}$ – булева, ее размерность – $l \times m$, а столбцы помечены переменными $y_1, y_2, \ldots, y_m$. Единицы в этих столбцах указывают элементарные конъюнкции из заданных ДНФ. Строка $\boldsymbol{u}$ троичной матрицы $\boldsymbol{U}$ *поглощает* булев вектор $\boldsymbol{a}$ той же размерности, если $\boldsymbol{a}$ принадлежит интервалу, представляемому строкой u. Пусть Z_1 и Z_2 – некоторые подмножества множества $X = \{x_1, x_2, \ldots, x_n\}$, такие, что $X = Z_1 \cup Z_2$ и $Z_1 \cap Z_2 = \varnothing$. Через $\boldsymbol{z}_1$ и $\boldsymbol{z}_2$ обозначим векторные переменные, компонентами которых служат соответственно переменные из подмножеств Z_1, Z_2.

Рассматриваемая задача декомпозиции ставится следующим образом: для системы полностью определенных булевых функций $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ требуется найти суперпозицию $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\varphi}(\boldsymbol{w}, \boldsymbol{z}_2)$, $\boldsymbol{w} = \boldsymbol{g}(\boldsymbol{z}_1)$, в которой число компонент векторной переменной $\boldsymbol{w}$ должно быть меньше, чем число компонент переменной z_1. Основное внимание в настоящей работе уделено поиску таких подмножеств Z_1 и Z_2, при которых данная задача может иметь решение.

2. Карта покрытия, компактная таблица

Покрытием π некоторого множества L назовем любую совокупность различных подмножеств множества L, объединение которых совпадает с множеством L. Элементом (блоком) покрытия может быть как пустое множество $\varnothing$, так и само множество L. Пусть $L = \{1, 2, \dots, l\}$ – множество номеров строк троичной матрицы $\boldsymbol{U}$. Покрытие π множества L назовем *покрытием троичной матрицы* $\boldsymbol{U}$, если каждому значению $\boldsymbol{x}^*$ векторной переменной $\boldsymbol{x}$ в покрытии π соответствует блок, содержащий номера всех тех и только тех строк матрицы $\boldsymbol{U}$, которые поглощают $\boldsymbol{x}^*$. Значению $\boldsymbol{x}^*$, не поглощаемому ни одной строкой матрицы $\boldsymbol{U}$, соответствует блок, представляющий собой $\varnothing$. Другие подмножества множества L не присутствуют в π.

Пусть $t(\boldsymbol{x}^*, \boldsymbol{U})$ – множество номеров тех строк троичной матрицы $\boldsymbol{U}$, которые поглощают булев вектор $\boldsymbol{x}^*$. Для каждого блока π_j покрытия π определим булеву функцию $\pi_j(\boldsymbol{x})$, положив, что для любого $\boldsymbol{x}^* \in \{0,1\}^n$ выполняется $\pi_j(\boldsymbol{x}^*) = 1$, если $t(\boldsymbol{x}^*, \boldsymbol{U}) = \pi_j$, и $\pi_j(\boldsymbol{x}^*) = 0$ в противном случае. Очевидно, покрытие π единственно для конкретной матрицы $\boldsymbol{U}$. Дизъюнкция всех булевых функций, приписанных блокам покрытия, равна логической единице, и конъюнкция любых двух булевых функций, приписанных различным блокам покрытия, равна логическому нулю [11], т. е. эти функции взаимно ортогональны.

Пусть для некоторых матриц $\boldsymbol{U}_1$ и $\boldsymbol{U}_2$ с общим множеством номеров строк L заданы соответственно покрытия π^1 и π^2. Сформируем мультимножество $\lambda = \{\pi^1_i \cap \pi^2_j \,/\, \pi^1_i \in \pi^1,\ \pi^2_j \in \pi^2,\ \pi^1_i(\boldsymbol{x}) \wedge \pi^2_j(\boldsymbol{x}) \neq 0\}$. Для каждого элемента $\lambda_{ij} = \pi^1_i \cap \pi^2_j$ мультимножества λ положим $\lambda_{ij}(\boldsymbol{x}) = \pi^1_i(\boldsymbol{x}) \wedge \pi^2_j(\boldsymbol{x})$. Образуем покрытие π', взяв в качестве его блоков все различные элементы мультимножества λ. Для всякого блока π'_s покрытия π' определим булеву функцию $\pi'_s(\boldsymbol{x})$ как дизъюнкцию всех булевых функций, приписанных тем элементам мультимножества λ, которые равны блоку π'_s. Покрытие π' есть *произведение покрытий* π^1 и π^2 ($\pi' = \pi^1 \times \pi^2$). Операция произведения дает возможность предложить простой способ вычисления покрытия для троичной матрицы $\boldsymbol{U}$ [11]. Этот способ состоит в получении произведения тривиальных покрытий всех одностолбцовых матриц, представляющих столбцы матрицы $\boldsymbol{U}$.

Определим операцию $\vee(\pi_i, \boldsymbol{V})$ над строками матрицы $\boldsymbol{V}$, результатом которой является вектор $\boldsymbol{y}^*$ ($\boldsymbol{y}^* = \vee(\pi_i, \boldsymbol{V})$), получаемый покомпонентной дизъюнкцией строк матрицы $\boldsymbol{V}$, номера которых входят в блок π_i. Если $\pi_i = \varnothing$, то все компоненты вектора $\boldsymbol{y}^*$ имеют значение 0. В работе [11] показано, что если $\pi_i(\boldsymbol{x}^*) = 1$, то $\boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{y}^* = \vee(\pi_i, \boldsymbol{V})$.

Удобным способом нахождения покрытия троичной матрицы $\boldsymbol{U}$ при небольшом числе аргументов является способ, использующий *карту покрытия* [12]. Карта покрытия аналогична карте Карно. Разница состоит в том, что в любой клетке карты Карно, соответствующей вектору $\boldsymbol{x}^*$, вместо значения $\boldsymbol{y}^* = \boldsymbol{f}(\boldsymbol{x}^*) = \vee(\pi_i, \boldsymbol{V})$ помещается множество $\pi_i = t(\boldsymbol{x}^*, \boldsymbol{U})$.

Пусть пара матриц $\boldsymbol{U}$, $\boldsymbol{V}$ задает систему полностью определенных булевых функций $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, и пусть матрица $\boldsymbol{U}_1$ составлена из столбцов матрицы $\boldsymbol{U}$, помеченных переменными из множества Z_1, а матрица $\boldsymbol{U}_2$ – из столбцов, помеченных переменными из множества Z_2. Покрытиями матриц $\boldsymbol{U}_1$ и $\boldsymbol{U}_2$ являются соответственно $\pi^1 = \{\pi^1_1, \pi^1_2, \dots, \pi^1_r\}$ и $\pi^2 = \{\pi^2_1, \pi^2_2, \dots, \pi^2_s\}$. По покрытиям π^1 и π^2 строится *компактная таблица* [11]. Столбцам этой таблицы припишем блоки $\pi^1_1, \pi^1_2, \dots, \pi^1_r$ и булевы функции $\pi^1_1(\boldsymbol{z}_1), \pi^1_2(\boldsymbol{z}_1), \dots, \pi^1_r(\boldsymbol{z}_1)$, а строкам – блоки $\pi^2_1, \pi^2_2, \dots, \pi^2_s$ и булевы функции $\pi^2_1(\boldsymbol{z}_2), \pi^2_2(\boldsymbol{z}_2), \dots, \pi^2_s(\boldsymbol{z}_2)$. На пересечении i-го столбца ($1 \le i \le r$) и j-й строки ($1 \le j \le s$) компактной таблицы поместим значение $\boldsymbol{y}^* = \vee(\pi^1_i \cap \pi^2_j, \boldsymbol{V})$. Компактная таблица задает систему булевых функций $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ следующим образом: на любом наборе аргументов $\boldsymbol{x}^*$, обращающем функцию $\pi^1_i(\boldsymbol{z}_1) \wedge \pi^2_j(\boldsymbol{z}_2)$ в единицу, значением векторной булевой функции $\boldsymbol{f}(\boldsymbol{x}^*)$ является $\vee(\pi^1_i \cap \pi^2_j, \boldsymbol{V})$. Компактная таблица имеет ту же форму, что и известная карта декомпозиции [2], с той лишь разницей, что строкам и столбцам карты декомпозиции соответствуют наборы значений аргументов, а строкам и столбцам компактной таблицы – попарно непересекающиеся множества таких наборов.

Имея компактную таблицу для системы функций $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, легко построить искомые системы $\boldsymbol{y} = \boldsymbol{\varphi}(\boldsymbol{w}, \boldsymbol{z}_2)$ и $\boldsymbol{w} = \boldsymbol{g}(\boldsymbol{z}_1)$. Столбцы компактной таблицы кодируются двоичными кодами, причем одинаковые столбцы могут иметь одинаковые коды. Длина кода равна $\lceil \log_2 r' \rceil$, где r' – число различных столбцов компактной таблицы и $\lceil a \rceil$ – наименьшее целое число, не меньшее a. Таким образом, определена система функций $\boldsymbol{w} = \boldsymbol{g}(\boldsymbol{z}_1)$. Значением векторной переменной $\boldsymbol{w}$ при любом наборе значений векторной переменной $\boldsymbol{z}_1$, обращающем функцию $\pi^1{}_i(\boldsymbol{z}_1)$ в единицу, является код i-го столбца ($1 \le i \le r$). Естественно, что если длина кода не меньше длины вектора $\boldsymbol{z}_1$, то при данном разбиении $\{Z_1, Z_2\}$ множества аргументов X задача не имеет решения. В противном случае имеющуюся компактную таблицу, столбцам которой припишем значения переменной $\boldsymbol{w}$, можно считать формой представления другой искомой системы функций $\boldsymbol{y} = \boldsymbol{\varphi}(\boldsymbol{w}, \boldsymbol{z}_2)$. Значением векторной переменной $\boldsymbol{y}$ при значении переменной $\boldsymbol{w}$, приписанном i-му столбцу ($1 \le i \le r$), и при любом значении переменной $\boldsymbol{z}_2$, обращающем функцию $\pi^2{}_j(\boldsymbol{z}_2)$ в единицу ($1 \le j \le s$), является вектор, присутствующий в компактной таблице на пересечении i-го столбца и j-й строки.

3. Поиск разбиения множества аргументов

Ранее было сказано, что получить покрытие троичной матрицы $\boldsymbol{U}$ можно, используя операцию произведения тривиальных покрытий всех одностолбцовых матриц, представляющих столбцы матрицы $\boldsymbol{U}$. Покрытие любого столбца троичной матрицы состоит ровно из двух блоков: один из них содержит номера одноэлементных строк, состоящих из нулей и знаков «–», другой – номера строк, состоящих из единиц и знаков «–». Если столбец состоит из одних нулей или из одних единиц, то один из блоков покрытия представляет собой пустое множество. Таким образом, если матрица $\boldsymbol{U}$ состоит из n столбцов, ее покрытие π можно получить как $\pi = \pi^1 \times \pi^2 \times \ldots \times \pi^n$, где π^i – покрытие i-го столбца ($1 \le i \le n$).

Пусть требуется найти небольшое количество свободных переменных, составляющих множество Z_2 (тогда множество связанных переменных Z_1 определится как $Z_1 = X \setminus Z_2$). Для этого используем *операцию деления* покрытия троичной матрицы на покрытие одного ее столбца.

Определим операцию деления покрытия π троичной матрицы $\boldsymbol{U}$ на покрытие π^i ее i-го столбца как $\pi / \pi^i = \pi^1 \times \pi^2 \times \ldots \times \pi^{i-1} \times \pi^{i+1} \times \ldots \times \pi^n$. Эту операцию легко выполнить с помощью карты покрытия, которая так же, как и карта Карно, имеет оси симметрии, связанные с переменными булева пространства, представляемого данной картой [5]. Чтобы преобразовать карту покрытия матрицы $\boldsymbol{U}$ в карту покрытия матрицы, получаемой из $\boldsymbol{U}$ удалением i-го столбца, надо совместить попарно элементы, симметричные относительно осей, соответствующих переменной x_i, и значением каждого из полученных элементов карты сделать объединение множеств из совмещаемых элементов. Полученная карта покрытия представит искомое покрытие.

Пример 1. Пусть система полностью определенных булевых функций $y = f(x)$ задана парой матриц $\boldsymbol{U}$, $\boldsymbol{V}$:

$$\boldsymbol{U} = \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & - \\ 0 & 1 & 0 & 0 & - \\ 0 & 1 & - & 0 & 1 \\ 0 & - & 0 & 0 & - \\ 0 & 0 & - & 0 & 1 \\ 1 & 1 & 0 & 1 & - \\ 1 & 1 & - & 1 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}, \quad \boldsymbol{V} = \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}.$$

При разбиении множества аргументов на подмножества $Z_1 = \{x_1, x_3, x_5\}$ и $Z_2 = \{x_2, x_4\}$ получим следующие матрицы:

$$\boldsymbol{U}_1 = \begin{array}{c} \begin{array}{ccc} x_1 & x_3 & x_5 \end{array} \\ \begin{bmatrix} 0 & 0 & - \\ 0 & 0 & - \\ 0 & - & 1 \\ 0 & 0 & - \\ 0 & - & 1 \\ 1 & 0 & - \\ 1 & - & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}, \qquad \boldsymbol{U}_2 = \begin{array}{c} \begin{array}{cc} x_2 & x_4 \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ - & 0 \\ 0 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}.$$

На рис. 1 показана карта покрытия троичной матрицы $\boldsymbol{U}$. Покрытием матрицы $\boldsymbol{U}$ является $\pi = \{\varnothing, \{1\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{2, 4\}, \{4, 5\}, \{6, 7\}, \{2, 3, 4\}\}$. Как видно из рис. 2, результатом деления покрытия π на покрытие столбца x_2 является $\{\varnothing, \{1\}, \{6\}, \{7\}, \{2, 4\}, \{3, 5\}, \{6, 7\}, \{2, 3, 4, 5\}\}$. Преобразовав эту карту покрытия относительно x_4, получаем $\pi^1 = \{\varnothing, \{6\}, \{7\}, \{3, 5\}, \{6, 7\}, \{1, 2, 4\}, \{1, 2, 3, 4, 5\}\}$ в качестве результата деления покрытия π на покрытия столбцов x_2 и x_4 (рис. 3). Аналогично разделив π на покрытия столбцов x_1, x_3 и x_5, получим $\pi^2 = \{\{1\}, \{4, 5\}, \{6, 7\}, \{2, 3, 4\}\}$.

x_5 x_4 x_3

4	∅	∅	1	1	∅	5	4,5
2,4	∅	∅	∅	∅	∅	3	2,3,4
∅	∅	∅	6	6,7	7	∅	∅
∅	∅	∅	∅	∅	∅	∅	∅

x_1 x_2

Рис. 1. Карта покрытия матрицы $\boldsymbol{U}$ из примера 1

x_5 x_4 x_3

2,4	∅	∅	1	1	∅	3,5	2,3,4,5
∅	∅	∅	6	6,7	7	∅	∅

x_1

Рис. 2. Карта покрытия, полученного от деления π на покрытие столбца x_2

x_5 x_3

1, 2, 4	∅	3, 5	1,2,3,4,5
6	∅	7	6,7

x_1

Рис. 3. Карта покрытия, полученного от деления π на покрытия столбцов x_2 и x_4

Компактная таблица для покрытий $\pi^1 = \{\varnothing, \{6\}, \{7\}, \{3, 5\}, \{6, 7\}, \{1, 2, 4\}, \{1, 2, 3, 4, 5\}\}$ и $\pi^2 = \{\{1\}, \{4, 5\}, \{6, 7\}, \{2, 3, 4\}\}$ имеет четыре различных столбца, для кодирования которых достаточно двух переменных, коды столбцов выделены снизу (табл. 1).

Таблица 1

	∅	6	7	3,5	6,7	1,2,4	1,2,3,4,5
1	00	00	00	00	00	10	10
4,5	00	00	00	01	00	01	01
6,7	00	01	01	00	01	00	00
2,3,4	00	00	00	10	00	11	11
	00	**01**	**01**	**10**	**01**	**11**	**11**

Для построения систем функций $\boldsymbol{y} = \varphi(\boldsymbol{w}, \boldsymbol{z}_2)$ и $\boldsymbol{w} = \boldsymbol{g}(\boldsymbol{z}_1)$, являющихся решением задачи декомпозиции, требуется получить функции, связанные с блоками полученных покрытий. ДНФ функций, связанных с блоками покрытия π^1, можно получить по карте покрытий на рис. 3: $\pi^1{}_1(\boldsymbol{z}_1) = x_3\,\bar{x}_5$, $\pi^1{}_2(\boldsymbol{z}_1) = x_1\,\bar{x}_3\,\bar{x}_5$, $\pi^1{}_3(\boldsymbol{z}_1) = x_1\,x_3\,x_5$, $\pi^1{}_4(\boldsymbol{z}_1) = \bar{x}_1\,x_3\,x_5$, $\pi^1{}_5(\boldsymbol{z}_1) = x_1\,\bar{x}_3\,x_5$, $\pi^1{}_6(\boldsymbol{z}_1) = \bar{x}_1\,\bar{x}_3\,\bar{x}_5$, $\pi^1{}_7(\boldsymbol{z}_1) = \bar{x}_1\,\bar{x}_3\,x_5$. Аналогично получаются функции $\pi^2{}_1(\boldsymbol{z}_2) = \bar{x}_2\,x_4$, $\pi^2{}_2(\boldsymbol{z}_2) = \bar{x}_2\,\bar{x}_4$, $\pi^2{}_3(\boldsymbol{z}_2) = x_2\,x_4$, $\pi^2{}_4(\boldsymbol{z}_2) = x_2\,\bar{x}_4$. В результате несложной минимизации получим следующие матрицы, представляющие искомую суперпозицию $\boldsymbol{y} = \varphi(\boldsymbol{w}, \boldsymbol{z}_2)$, $\boldsymbol{w} = \boldsymbol{g}(\boldsymbol{z}_1)$:

$$
\begin{array}{c} w_1\ w_2\ x_2\ x_4 \\ \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & - & 1 & 0 \\ 1 & - & 0 & 0 \\ 1 & 1 & - & 0 \end{bmatrix} \end{array},
\begin{array}{c} y_1\ y_2 \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array};
\qquad
\begin{array}{c} x_1\ x_3\ x_5 \\ \begin{bmatrix} 0 & 0 & - \\ - & 0 & 0 \\ 1 & - & 1 \end{bmatrix} \end{array},
\begin{array}{c} w_1\ w_2 \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array}.
$$

4. Критерий оценки разбиения. Быстрое определение разложимости

Известно, что в худшем случае сложность схемы, реализующей булеву функцию от n переменных, оценивается как $O(2^n/n)$ [1]. Если $\boldsymbol{f}(\boldsymbol{x})$ допускает декомпозицию вида $\boldsymbol{f}(\boldsymbol{x}) = \varphi(\boldsymbol{g}(\boldsymbol{z}_1), \boldsymbol{z}_2)$, то в общем случае сложность реализации такой структуры оценивается как

$$O(2^{n_1}/n_1 + 2^{n_2+1}/(n_2+1)), \tag{1}$$

где $n_1 = |Z_1|$ и $n_2 = |Z_2|$. Когда $n = |X|$ велико, имеется возможность уменьшить сложность схемы, использовав декомпозицию [1]. Из выражения (1) видно, что сложность реализации тем меньше, чем n_1 и n_2 ближе к $n/2$. Таким образом, для получения лучшего решения поиск разбиения следует начинать с $n_1 = \lceil n/2 \rceil$. Если решение получено, процесс заканчивается. В противном случае происходит проверка по двум направлениям: в сторону уменьшения и в сторону увеличения n_1. Проверка заканчивается, когда получено решение или просмотрены все разбиения.

В процессе эксперимента и анализа результатов установлено, что при максимальном значении $|Z_1|$ декомпозиция существует с высокой вероятностью. Эксперимент начат со значения $|Z_1| = n - 1$ (n – число аргументов), и далее осуществлен лексикографический перебор всех возможных разбиений. Исследованы все возможные разбиения, начиная от наибольшей мощности и кончая наименьшей мощностью множества Z_1.

5. Реализация метода

Для исследования задач декомпозиции системы булевых функций разработана компьютерная программа на языке C++ в операционной системе Windows. Эксперименты проводились на компьютере Pentium 2.26GHz CPU с оперативной памятью 3 Gb. В качестве примеров генерировались системы полностью определенных булевых функций с помощью средств, описанных в работах [13, 14]. Общая схема реализации алгоритма, определяющего разложимость системы, имеет следующий вид:

```
for Con ← ConDownto ConUp
    for Arg ← ArgDownto ArgUp
        for Fun ← FunDownto FunUp
           ➢Generate new SOP(Con,Arg,Fun)
            (генерирование матриц U и V)
           ➢Expand Matrix U
            (расщепление строк матрицы U с целью превращения ее в эквивалентную ей
            булеву матрицу)
           ➢Compute Cover Map
            (генерирование последовательности кодов Грея длины 2^n и построение
            матрицы покрытия)
           ➢for k ← n-1 downto2
               »Combination Generator(n,k)
               »for each combination of (n k)
                   Check Current Partition
                    1- Divide Cover Map Over Z1
                    2- Divide Cover Map over Z2
                    3- Compute Compact Table
                    4- Compute Number of Different Columns (r) of Compact
                     Table
                    5- Encode the Columns of the Compact Table
                    6- if r ≤ 2^(k-1) then
                      a Solution Founded
                     (Produce Matrices ϕ, Y, X and W)
               »ifa Solution Founded then
                   Declare the current System is Decomposable and Stop
           ➢if Solution not Founded then
               Declare the current System is not Decomposable
```

Алгоритм был модифицирован для нахождения всех решений и лучшего решения. Рассматривались три параметра системы булевых функций: число строк матрицы $\boldsymbol{U}$ (число различных элементарных конъюнкций), число столбцов матрицы $\boldsymbol{U}$ (число аргументов системы) и число столбцов матрицы $\boldsymbol{V}$ (число функций).

Для удобства вычисления компактной таблицы карта покрытия представляется в памяти компьютера как одномерный массив. Пример такого представления для трех переменных показан на рис. 4.

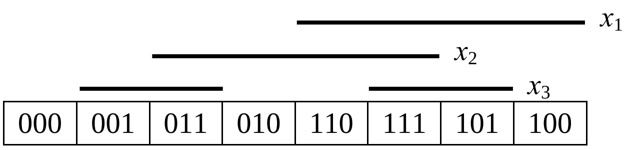

Рис. 4. Карта покрытия в виде одномерного массива

Для того чтобы получить очередное разбиение, после построения карты покрытия для очередной системы булевых функций генерируются k-элементные подмножества n-элементного множества аргументов с помощью алгоритма Кнута [15]. Элементы этих подмножеств составляют множество Z_1, а остальные – множество Z_2.

Если найдено разбиение, приводящее к решению задачи декомпозиции, программа строит четыре матрицы: матрицы $\boldsymbol{\Phi}$ и $\boldsymbol{Y}$ задают систему $\boldsymbol{\varphi}(\boldsymbol{w}, \boldsymbol{z}_2)$, а матрицы $\boldsymbol{X}$ и $\boldsymbol{W}$ – систему $\boldsymbol{g}(\boldsymbol{z}_1)$. Эти матрицы представляют решение задачи декомпозиции.

Для нахождения всех решений требуется просмотреть все возможные разбиения множества аргументов X. Для этого используется лексикографический перебор. Если подходящего разбиения не найдено, программа сообщает о том, что текущая система булевых функций неразложима, в противном случае программа выдает число решений для этой системы. Для нахождения лучшего решения используется критерий (1), а для быстрого определения разложимости системы булевых функций – лексикографический перебор разбиений с началом при $|Z_1| = n - 1$.

6. Обсуждение результатов

Ограничение памяти компьютера и временные ограничения, к сожалению, не дают проводить исследования на больших примерах. Результаты, сведенные в табл. 2–4, относятся лишь к системам булевых функций с небольшими значениями числа аргументов и числа функций. Результаты показывают, что более чем 95 % сгенерированных систем являются разложимыми и для них всех имеется несколько решений. Генерировались системы дизъюнктивных нормальных форм, где средний ранг элементарных конъюнкций составлял половину числа аргументов.

Таблица 2

Результаты получения всех решений

Con	Arg	Fun	TNP	NS	PS	ET
8	5	2	25	21	84	<1
10	6	2	56	49	87	4
10	6	4	56	18	32	4
15	6	8	56	26	46	5
20	7	3	119	25	21	21
20	7	10	119	15	13	25
25	7	14	119	7	6	34
15	8	4	246	40	16	134
20	8	6	246	104	42	123
40	8	10	246	8	3	201
30	9	5	501	125	25	926
25	9	9	501	30	6	650
25	9	16	501	61	12	824
30	10	3	1012	58	6	3634
30	10	5	1012	55	5	3197
30	10	8	1012	491	49	1362
30	12	6	4082	673	16	20413

Таблица 3

Результаты получения лучшего решения

Con	Arg	Fun	TNP	NIP	PIP	$\lvert Z_1 \rvert$	Opt	ET
8	5	2	25	1	4,0	2	да	<1
10	6	2	56	1	1,79	3	да	<1
10	6	3	56	10	17,86	3	да	<1
15	6	5	56	1	1,79	3	да	<1
15	6	8	56	3	5,36	3	да	1
20	7	3	119	3	2,52	3	да	2
20	7	10	119	2	1,68	3	да	2
15	8	4	246	2	0,81	4	да	3
20	8	5	246	14	5,69	4	да	10
25	8	8	246	2	1,68	4	да	10
25	8	12	246	72	29,27	5	нет	71
20	9	4	501	14	2,79	4	да	24

Окончание таблицы 3

Con	Arg	Fun	TNP	NIP	PIP	$\lvert Z_1\rvert$	Opt	ET
30	9	8	501	6	1,20	4	да	33
38	9	16	501	36	7,19	4	да	47
40	10	4	1012	6	0,59	5	да	147
30	10	6	1012	1	0,099	5	да	125
20	11	7	2035	111	5,45	5	да	407
20	12	5	4082	35	0,86	6	да	452
17	12	10	4082	1	0,025	6	да	308
Среднее	*8,26*	*6,42*	*800*	*12*	*4,77*	*4*	*95*	*87*

В табл. 2–4 первые три столбца представляют соответственно число элементарных конъюнкций (Con), число аргументов (Arg) и число функций (Fun), которые являются параметрами генерируемых систем булевых функций. Общее число разбиений (TNP) при $2 \le |Z_1| \le n - 1$ подсчитывается по формуле $\sum_{k=2}^{n-1}\binom{n}{k} = 2^n - (n + 2)$. Столбец ET представляет время выполнения программы в секундах.

В столбцах NS и PS табл. 2 показаны соответственно число решений и доля решений относительно TNP в процентах. В табл. 3 NIP представляет число исследуемых разбиений, которые пришлось перебрать, чтобы получить подходящее разбиение, а PIP – долю этих разбиений относительно общего числа разбиений в процентах. Столбец Opt показывает оптимальность полученного решения. Решение оптимально, если мощность $|Z_1|$ близка к $n/2$. В табл. 4 столбец NTS представляет общее число систем, обработанных программой, ND – число разложимых систем среди них, PD – долю разложимых систем в процентах.

Таблица 4

Результаты быстрого определения разложимости

Con		Arg		Fun		NTS	PD	ND
от	до	от	до	от	до			
6	15	4	6	2	4	90	90	81
10	15	5	8	2	6	120	96	116
10	25	6	8	4	8	240	93	224
10	30	5	10	3	8	756	95	725
10	30	6	10	4	12	945	90	851
20	50	8	10	4	8	465	99	461
15	40	5	12	2	6	1040	99	1032
20	30	8	12	6	9	220	99	218
59	60	10	12	4	8	30	100	30
59	60	10	12	6	10	30	100	30
50	50	10	14	6	8	15	100	15
60	60	12	15	6	12	28	100	28
50	50	14	16	6	10	15	100	15

Из табл. 2 видно, что для рассматриваемых систем среди всех разбиений подходящими для декомпозиции являются от 3 до 87 %. Результаты, приведенные в табл. 3, показывают, что примерно 5 % разбиений дают лучшее решение. Около 95 % получаемых решений являются оптимальными по критерию (1), т. е. число аргументов в каждом блоке разбиения близко к $n/2$. Как видно из табл. 4, все системы имели решение при максимальной мощности множества Z_1. Это означает, что для определения разложимости системы достаточно исследовать только разбиения с максимальным множеством Z_1.

Заключение

Результаты исследований разложимости систем булевых функций показали, что обычно для разложимой системы имеется несколько решений. В большинстве случаев число решений тем больше, чем меньше число функций в системе и чем меньше ранги элементарных конъюнкций в заданных ДНФ. Решение обычно оказывается оптимальным, если ранги конъюнкций невелики. Все рассмотренные системы функций допускали нетривиальную декомпозицию при максимальной мощности множества связанных переменных.

Список литературы

1. Hassoun, S. Logic Synthesis and Verification. The Springer International Series in Engineering and Computer Science / S. Hassoun, T. Sasao. – Kluwer Academic Publishers, 2001. – 472 p.

2. Perkowski, M.A. A Survey of Literature on Functional Decomposition, Version IV / M.A. Perkowski, S. Grygiel ; Portland State University, Department of Electrical Engineering. – Portland, USA, 1995. – 188 p.

3. Muthukumar, V. An efficient variable partitioning approach for functional decomposition of circuits / V. Muthukumar, R.J. Bignall, H. Selvaraj // Journal of Systems Architecture. – 2007. – Vol. 53, no. 1. – P. 53–67.

4. An improved functional decomposition method based on FAST and the method of removal and operation / F. Yu [et al.] // International Conference on System Science and Engineering (ICSSE), Dalian, China, Jun. 2012. – Dalian, 2012. – P. 487–492.

5. Закревский, А.Д. Логические основы проектирования дискретных устройств / А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.

6. Бибило, П.Н. Декомпозиция булевых функций на основе решения логических уравнений / П.Н. Бибило. – Минск : Беларус. навука, 2009. – 211 с.

7. Jóźwiak, L. An effective and efficient method for functional decomposition of boolean functions based on information relationship measures / L. Jóźwiak, A. Chojnacki // 3rd Design and Diagnostics of Electronic Circuits and Systems Workshop (DDECS), Bratislava, Slovakia, Apr. 2000. – Bratislava, 2000. – P. 242–249.

8. Закревский, А.Д. Декомпозиция частичных булевых функций – проверка на разделимость по заданному разбиению / А.Д. Закревский // Информатика. – № 1(13). – 2007. – С. 16–21.

9. Files, C.M. New mutivalued functional decomposition algorithms based on MDDs / C.M. Files, M.A. Perkowski // IEEE Transactions on Computer-Aided Design of Integrated Ciruits and Systems. – 2000. – Vol. 19, no. 9. – P. 1081–1086.

10. Rawski, M. Input variable partitioning method for decomposition-based logic synthesis targeted heterogeneous FPGAs / M. Rawski // International Journal of Electronics and Telecommunications. – 2012. – Vol. 58, no. 1. – P. 15–20.

11. Поттосин, Ю.В. Табличные методы декомпозиции систем полностью определенных булевых функций / Ю.В. Поттосин, Е.А. Шестаков. – Минск : Белорус. наука, 2006. – 327 с.

12. Поттосин, Ю.В. Применение аппарата покрытий троичных матриц для поиска разбиения множества аргументов при декомпозиции булевых функций / Ю.В. Поттосин, Е.А. Шестаков // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2011. – № 3 (16). – С. 100–107.

13. Романов, В.И. Разработка инструментальных средств логического проектирования / В.И. Романов // Логическое проектирование. Вып. 6. – Минск : Ин-т техн. кибернетики НАН Беларуси, 2001. – С. 151–170.

14. Romanov, V.I. Tools for programming Boolean calculations / V.I. Romanov // Abstracts of ECCO XVIII Conf. «Combinatorics for Modern Manufacturing, Logistics and Supply Chains», Minsk, Belarus. – Minsk, 2005. – P. 57–58.

15. Кнут, Д.Э. Искусство программирования. Т. 4, А. Комбинаторные алгоритмы. Ч. 1 / Д.Э. Кнут. – М. : Вильямс, 2013. – 960 с.

Поступила 01.03.13

Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: pott@newman.bas-net.by

S.H. Taghavi Afshord, Yu.V. Pottosin

INVESTIGATION OF DECOMPOSABILITY OF A SYSTEM OF BOOLEAN FUNCTIONS

A computer program is described which analyzes the decomposability of a system of Boolean functions and searches for an appropriate partition of the argument set. Three tasks linked with a system of Boolean functions are given: producing all solutions, searching for the best solution from the circuit complexity point of view, and finding a solution as quickly as possible, if the system is decomposable. The results of computer experiment for determining decomposability of systems of Boolean functions are given.