ИНФОРМАТИКА

2007 **апрель-июнь** **№ 2**

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

УДК 519.7

А.Д. Закревский

ДЕКОМПОЗИЦИЯ ЧАСТИЧНЫХ БУЛЕВЫХ ФУНКЦИЙ – ПОИСК ПОДХОДЯЩЕГО РАЗБИЕНИЯ

Исследуется проблема последовательной двухблочной декомпозиции частичных булевых функций по нестрогому разбиению на множестве аргументов. Рассматривается ключевая комбинаторная задача: нахождение подходящего разбиения на множестве аргументов, т. е. такого, по которому функция разделима. Предлагается алгоритм, существенно ускоряющий поиск подходящего разбиения путем предварительного обнаружения его следов. Алгоритм формулируется в терминах булевых и троичных векторов и матриц с использованием эффективных параллельных операций над ними.

Введение

Рассматриваемая задача относится к проблеме функциональной декомпозиции булевых функций, поставленной первоначально в работах [1–3]. Уточним ее следующим образом.

Пусть задана некоторая булева функция $f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_n)$. Требуется декомпозировать ее, представив в виде следующей композиции двух функций g и h от меньшего числа переменных:

$$f(\boldsymbol{x}) = g(h(\boldsymbol{u}, \boldsymbol{w}), \boldsymbol{w}, \boldsymbol{v}).$$

При этом заданные в векторной форме множества аргументов связаны отношениями $\boldsymbol{x} = \boldsymbol{u} \cup \boldsymbol{w} \cup \boldsymbol{v}$, $\boldsymbol{u} \cap \boldsymbol{w} = \boldsymbol{u} \cap \boldsymbol{v} = \boldsymbol{w} \cap \boldsymbol{v} = \varnothing$ и пара множеств $\boldsymbol{u}$ и $\boldsymbol{v}$ задает нестрогое разбиение на множестве $\boldsymbol{x}$, обозначаемое через $\boldsymbol{u}/\boldsymbol{v}$. Такую композицию принято называть последовательной двухблочной.

Для решения сформулированной задачи необходимо, прежде всего, найти такое нестрогое разбиение $\boldsymbol{u}/\boldsymbol{v}$, при котором переменные из множества $\boldsymbol{u}$ входят в число аргументов только функции h , а переменные из $\boldsymbol{v}$ – только функции g. Должны выполняться также условия $|\boldsymbol{u}| > 1$ и $|\boldsymbol{v}| > 0$ − иначе композиция окажется тривиальной (существует всегда). Назовем такое разбиение *подходящим*, а функцию $f(\boldsymbol{x})$ *разделимой*, или *декомпозируемой*, по данному разбиению.

Нахождение подходящего разбиения является трудной задачей, для решения которой в работе [4] был предложен эффективный комбинаторный алгоритм, применимый к полностью определенной булевой функции. Эта задача еще более усложняется, когда функция $f(\boldsymbol{x})$ оказывается частичной, будучи определена не на всех наборах значений переменных из множества $\boldsymbol{x}$. Именно этот случай рассматривается ниже.

В работе [4] было показано, что вероятность декомпозируемости случайной полностью определенной булевой функции быстро стремится к нулю с ростом числа переменных n, так что уже при $n > 9$ эта функция, скорее всего, не декомпозируема. В случае частичной функции такая вероятность растет с ростом неопределенности, однако и тогда может оставаться достаточно малой.

Учитывая данное замечание, считаем, что рассматриваемая функция $f(\boldsymbol{x})$ разделима, будучи получена в результате композиции $g(h(\boldsymbol{u}, \boldsymbol{w}), \boldsymbol{w}, \boldsymbol{v})$ некоторых двух булевых функций g и h на некотором нестрогом разбиении $\boldsymbol{u}/\boldsymbol{v}$ множества аргументов $\boldsymbol{x}$. Требуется обнаружить (распознать) это разбиение, после чего получение функций g и h не составит труда.

В работе [5] предложен метод проверки частичной булевой функции на разделимость по заданному нестрогому разбиению. В случае, когда это подходящее разбиение априори неизвестно, можно организовать его поиск, перебирая различные нестрогие разбиения и проверяя функцию на разделимость по ним. Однако такой способ весьма трудоемок, поскольку число различных нестрогих разбиений на множестве n переменных аппроксимируется сверху величиной 3^n, быстро растущей с увеличением числа переменных.

В настоящей статье описывается метод поиска подходящего разбиения $\boldsymbol{u}/\boldsymbol{v}$ по его следам, существенно сокращающий число анализируемых разбиений. Первоначально он был разработан для полностью определенных булевых функций [4], но здесь обобщается на случай частичных булевых функций.

1. Базисные операции в булевом пространстве

В предлагаемом методе поиска подходящего разбиения используется параллелизм покомпонентных операций над длинными булевыми векторами, существенно ускоряющий проводимые вычисления.

Булевы функции $f(\boldsymbol{x}) = f(x_1, x_2, \dots, x_n)$ и $g(\boldsymbol{x}) = g(x_1, x_2, \dots, x_n)$ от n переменных можно рассматривать как соответствующие подмножества элементов булева n-мерного пространства $\{0, 1\}^n$. Будем задавать их 2^n-мерными булевыми векторами $\boldsymbol{f}$ и $\boldsymbol{g}$, показывая такие векторы в приводимых ниже примерах в более удобной для визуального восприятия матричной форме. В таком представлении двухместные булевы операции $f \vee g, f \wedge g$, $f \oplus g, f \rightarrow g$ легко реализуются как параллельные покомпонентные операции над соответствующими булевыми векторами, обозначаемые $\boldsymbol{f} \vee \boldsymbol{g}, \boldsymbol{f} \wedge \boldsymbol{g}$ (или просто $\boldsymbol{f}\boldsymbol{g}$), $\boldsymbol{f} \oplus \boldsymbol{g}$, $\boldsymbol{f} \rightarrow \boldsymbol{g}$. Включая эти операции в формируемый базис, дополним их некоторыми операциями взаимодействия различных компонент в рамках одного булева вектора.

Напомним, что функция $f(\boldsymbol{x})$ может быть представлена в виде дизъюнктивного разложения Шеннона по произвольной переменной x_i:

$$f(\boldsymbol{x}) = \bar{x}_i f_{i0} \vee x_i f_{i1} ,$$

коэффициентами f_{i0} и f_{i1} которого служат булевы функции, получаемые в результате подстановки вместо переменной x_i значения 0 или 1.

Пользуясь векторным представлением функции, обозначим эти операции через $\boldsymbol{f}–i$ и $\boldsymbol{f}+i$ соответственно. Они легко реализуются в булевом пространстве на парах элементов, соседних по переменной x_i. При выполнении операции $\boldsymbol{f}–i$ оба элемента пары получают значение элемента, определяемого условием $x_i = 0$, при выполнении операции $\boldsymbol{f}+i$ – значение другого элемента, соответствующего значению 1 переменной x_i. Приведем примеры таких операций, а также их композиций:

```
             f                     f-2                   f-2-5
          --------              --------              --------   4
      ----    ----          ----    ----          ----    ----  3
    --  --  --  --        --  --  --  --        --  --  --  -- 2
   - - - - - - - -       - - - - - - - -       - - - - - - - - 1
    0110110101011110      0101111101011111      0101111101011111
  | 0010010000010110      0000010100000101      0101111101011111
 |  1100101001110001      1111101001010000      1111101001010000
 || 0100010111010011      0101010111110000      1111101001010000

    1101110111101110      1010101000010001      0000000000110011
  | 0100010001100110      0101010100110011      0000000000110011
 |  1010101000010001      1010101000010001      1100110011111111
 || 0101010100110011      0101010100110011      1100110011111111
6 5          f+3                  f+3+6                f+1-3+5
```

Путем взаимодействия соседних элементов реализуются также операция $\mathrm{Inv}_i \boldsymbol{f}$ инвертирования функции $\boldsymbol{f}$ по переменной x_i (соседние элементы обмениваются своими значениями) и так называемые операции симметрирования $\mathrm{S}_i{*}\boldsymbol{f}$, в которых оба элемента получают значение, определяемое двухместной операцией $*$ над их исходными значениями [6]. В результате этих операций функция

$$f(\boldsymbol{x}) = \bar{x}_i f_{i0} \vee x_i f_{i1}$$

преобразуется соответственно в функции

$$\bar{x}_i f_{i1} \vee x_i f_{i0}\ ,\ \ \bar{x}_i (f_{i0} * f_{i1}) \vee x_i (f_{i0} * f_{i1}).$$

Примеры таких операций показаны ниже:

```
           f                   Inv4 f                S4∨ f

            --------            --------            --------    4
        ----    ----        ----    ----        ----    ----   3
      --  --  --  --      --  --  --  --      --  --  --  --  2
     - - - - - - - -     - - - - - - - -     - - - - - - - - 1
    0110110101011110    0101111001101101    0111111101111111
 |  0010010000010110    0001011000100100    0011011000110110
|   1100101001110001    0111000111001010    1111101111111011
||  0100010111010011    1101001101000101    1101011111010111

    0000110000001100    0011001100110011    0111111101111111
 |  0000000000000000    0011001000110010    0111111101111111
|   1100000000110000    1011101110111011    1111111111111111
||  0000000011000011    1001011010010110    1111111111111111

6 5
          S1∧ f                S4⊕ f               S4,5∨ f
```

Через $S_{4,5}{}^{\vee}\boldsymbol{f}$ обозначена композиция $S_4{}^{\vee}(S_5{}^{\vee}\boldsymbol{f})$.

2. Метод поиска по следу. Триады и фрагменты

Предлагаемый ниже метод декомпозиции основан на следующих соображениях, ключевые моменты которых оформлены в виде утверждений.

Пусть заданы два нестрогие разбиения $\boldsymbol{u}/\boldsymbol{v}$ и $\boldsymbol{u}^*/\boldsymbol{v}^*$, связанные отношениями $\boldsymbol{u}^* \subseteq \boldsymbol{u}$ и $\boldsymbol{v}^* \subseteq \boldsymbol{v}$. Будем говорить, что в этом случае разбиение $\boldsymbol{u}^*/\boldsymbol{v}^*$ *подчиняется* разбиению $\boldsymbol{u}/\boldsymbol{v}$ или является его *следом*.

Утверждение 1. *Если частичная булева функция $f(\boldsymbol{x})$ декомпозируется по разбиению $\boldsymbol{u}/\boldsymbol{v}$, то она декомпозируется и по его следу $\boldsymbol{u}^*/\boldsymbol{v}^*$.*

Следствие. *Если функция $f(\boldsymbol{x})$ не декомпозируется по разбиению $\boldsymbol{u}^*/\boldsymbol{v}^*$, то она не декомпозируется и по $\boldsymbol{u}/\boldsymbol{v}$.*

Положим, что $|\boldsymbol{u}| = k$ и $|\boldsymbol{v}| = m$. Разбиение с $k = 2$ и $m = 1$ назовем *триадой*. Оно является простейшим из разбиений, по которому может иметь место нетривиальная декомпозиция.

Утверждение 2. *Булева функция $f(\boldsymbol{x})$ не декомпозируема, если она не декомпозируется ни по какой из триад.*

Поэтому в предлагаемом методе поиск разбиения $\boldsymbol{u}/\boldsymbol{v}$ начинается с поиска его следов на множестве триад, т. е. с нахождения подходящей триады. Просмотр триад может производиться достаточно быстро, поскольку их число относительно мало, будучи значительно меньше числа всех нестрогих разбиений.

Утверждение 3. *Число триад равно* $C_n^2\,(n-2) = \dfrac{n(n-1)(n-2)}{2}$.

Допустим, что обнаружена подходящая триада $(x_p, x_q)/x_r$. Если она подчиняется искомому разбиению $\boldsymbol{u}/\boldsymbol{v}$, то последнее можно найти, положив для начала $\boldsymbol{u} = (x_p, x_q)$ и $\boldsymbol{v} = (x_r)$, а затем последовательно расширяя эти два множества, перебирая остальные переменные и испытывая их на возможность включения в множество $\boldsymbol{u}$ или $\boldsymbol{v}$.

При рассмотрении некоторой конкретной триады $\boldsymbol{u}/\boldsymbol{v}$ булево пространство $M = \{0, 1\}^n$, на котором задана частичная булева функция $f(\boldsymbol{x})$, разбивается на $2^{|\boldsymbol{w}|}$ интервалов, соответствующих различным значениям вектора $\boldsymbol{w}$. На каждом из них задан соответствующий коэффициент f_i дизъюнктивного разложения функции по переменным множества $\boldsymbol{w}$, представляющий собой некоторую частичную булеву функцию переменных x_p, x_q, x_r. Для удобства последующих рассуждений представим каждый из этих коэффициентов троичной матрицей размером 4×2, строки которой соответствуют значениям двухкомпонентного вектора $\boldsymbol{u}$, а столбцы – значениям однокомпонентного вектора $\boldsymbol{v}$. Обозначим эту матрицу $\boldsymbol{T}_i$ и назовем *фрагментом*. Таким образом, 2^n-элементная троичная матрица, представляющая функцию $f(\boldsymbol{x})$, разлагается на 2^{n-3} восьмиэлементных фрагментов, задающих функции $f_i(x_\mathrm{p}, x_q, x_r)$.

Ниже приводится конкретный пример такого разбиения на восемь фрагментов для частичной булевой функции $f(a, b, c, d, e, f)$ и триады $(a, b)/c$:

```
                -----------   f
          -----       -----  e
       --    --    --    --  d
     -  -  -  -  -  -  -  -  c

    10 0- 1- 11 0- 10 -1 10
 |  -1 00 -1 1- 00 -1 1- -0
|   0- 1- -0 10 10 0- 11 -1
||  1- 01 11 -1 00 10 -0 11

b a
```

Утверждение 4. *Функция $f(\boldsymbol{x})$ декомпозируется по триаде $(x_p, x_q)/x_r$, если и только если каждый из коэффициентов $f_i(x_p, x_q, x_r)$ также декомпозируется по этой же триаде.*

Отсюда следует, что вероятность декомпозируемости функции $f(\boldsymbol{x})$ по конкретной триаде равна γ^k, где k – число коэффициентов, равное 2^{n-3}, а γ – вероятность декомпозируемости одного из них. В случае полностью определенной булевой функции последняя вероятность аппроксимируется величиной 1/3, а с ростом неопределенности возрастает. Тем не менее, вероятность декомпозируемости функции $f(\boldsymbol{x})$ быстро падает с увеличением числа переменных в ней.

3. Проверка триады на пригодность

Итак, триада пригодна, если пригоден каждый из фрагментов соответствующего разбиения троичной матрицы, а фрагмент пригоден, если частичную функцию $f(\boldsymbol{x})$ можно доопределить таким образом, что фрагмент будет содержать не более двух типов булевых строк (строки одного типа равны). Троичные строки совместимы, т. е. могут стать равными при доопределении, если они не ортогональны. Значит, фрагмент пригоден, если он содержит не более двух классов совместимых строк и если граф ортогональности его строк бихроматичен [7].

Предложим следующий способ проверки фрагментов с целью выявления среди них подходящих. Фрагмент содержит четыре строки, поэтому граф ортогональности имеет четыре вершины. Он бихроматичен, если в нем не найдется цикла длины три. Выберем произвольно две различные вершины. Если такой цикл существует, то, по крайней мере, одна из выбранных вершин будет ему принадлежать. Следовательно, достаточно проверить каждую из этих двух вершин на вхождение в цикл длины три. Если такое вхождение не будет обнаружено, граф бихроматичен и, значит, триада подходящая.

Необходимое и достаточное условие вхождения вершины, т. е. соответствующей ей строки, в цикл длины три сформулируем так: среди строк, ортогональных данной, существуют взаимно ортогональные строки.

Например, первый из показанных ниже фрагментов оказывается подходящим, а второй – нет, поскольку существует цикл длины три, составленный из трех последних строк: каждая из них ортогональна двум другим.

```
10    -1   1
-1    00   2
0-    1-   3
01    01   4
```

Предложенный способ реализуется следующим алгоритмом, особенностью которого является то, что он проверяет на пригодность одновременно все 2^{n-3} фрагментов, соответствующих заданной триаде, и тем самым проверяет пригодность триады в целом. Частичная функция

$f(\boldsymbol{x})$ задается парой булевых векторов $\boldsymbol{f}^0$ и $\boldsymbol{f}^1$, в первом из которых единицами отмечены значения 0 функции $f(\boldsymbol{x})$, а во втором – значения 1. Пространство разбивается на фрагменты по триаде $(x_p, x_q) / x_r$.

Сначала проверка производится по первым строкам фрагментов, образующим в совокупности начальный коэффициент f^- разложения функции $f(\boldsymbol{x})$ по переменным x_p и x_q. Строки, ортогональные проверяемой строке, отмечаются значением 1 в вычисляемом векторе $\boldsymbol{g}$, а их значения фиксируются парой векторов $\boldsymbol{h}^0$ и $\boldsymbol{h}^1$, проверяемой далее на ортогональность. Так же, как и исходные векторы $\boldsymbol{f}^0$ и $\boldsymbol{f}^1$, это булевы векторы с 2^n компонентами:

$\boldsymbol{h}^0 := (\boldsymbol{f}^0 - p) - q$ $\boldsymbol{h}^1 := (\boldsymbol{f}^1 - p) - q$	получение начального коэффициента f^-;
$\boldsymbol{g} := \mathrm{S}_r^\vee(\boldsymbol{h}^0\boldsymbol{f}^1 \vee \boldsymbol{h}^1\boldsymbol{f}^0)$	выделение коэффициентов, ортогональных f^-;
$\boldsymbol{h}^0 := \mathrm{S}_{\boldsymbol{u}}^\vee(\boldsymbol{f}^0\boldsymbol{g})$ $\boldsymbol{h}^1 := \mathrm{S}_{\boldsymbol{u}}^\vee(\boldsymbol{f}^1\boldsymbol{g})$	получение их пересечения.

Если при этом оказывается, что $\boldsymbol{h}^0\boldsymbol{h}^1 \neq \boldsymbol{0}$, то триада признается непригодной. В противном случае производится проверка по последним строкам, образующим конечный коэффициент f^+:

$\boldsymbol{h}^0 := (\boldsymbol{f}^0 + p) + q$ $\boldsymbol{h}^1 := (\boldsymbol{f}^1 + p) + q$	получение конечного коэффициента f^+;
$\boldsymbol{g} := \mathrm{S}_r^\vee(\boldsymbol{h}^0\boldsymbol{f}^1 \vee \boldsymbol{h}^1\boldsymbol{f}^0)$	выделение коэффициентов, ортогональных f^+;
$\boldsymbol{h}^0 := \mathrm{S}_{\boldsymbol{u}}^\vee(\boldsymbol{f}^0\boldsymbol{g})$ $\boldsymbol{h}^1 := \mathrm{S}_{\boldsymbol{u}}^\vee(\boldsymbol{f}^1\boldsymbol{g})$.	получение их пересечения.

Если $\boldsymbol{h}^0\boldsymbol{h}^1 \neq \boldsymbol{0}$, то триада непригодна, в противном случае триада признается пригодной.

Пример. Вернемся к рассмотрению частичной булевой функции $f(a, b, c, d, e, f)$, представив ее парой булевых векторов (свернутых в матрицы) $\boldsymbol{f}^0$ и $\boldsymbol{f}^1$:

```
                 -----------    f
           -----       -----   e
        --    --    --    --  d
      -  -  -  -  -  -  -  -  c

     10 0- 0- 11 0- 10 -1 10
 |   -1 10 -1 1- 00 -1 0- -0
|    0- -1 -0 00 10 0- 11 -1
||   1- -0 11 -1 -0 10 -0 11
b a
     01 10 10 00 10 01 00 01   f^0
     00 01 00 00 10 00 10 01
     10 00 01 11 00 10 00 00
     00 01 00 00 00 01 01 00

     10 00 00 11 00 10 01 00   f^1
     01 10 01 10 00 01 00 00
     00 01 00 00 10 00 11 01
     10 00 11 01 00 10 00 01
```

Проверка функции на декомпозируемость по разбиению $(a, b) / c$ сводится к проверке пригодности всех фрагментов, проводимой параллельно. В данном примере устанавливается пригодность всех восьми фрагментов, в результате чего выясняется, что функция декомпозируема по заданному разбиению.

Ограничимся иллюстрацией операции проверки одного из фрагментов, третьего слева, демонстрируя исходные значения соответствующих компонент исходного троичного вектора $\boldsymbol{f}$ и получаемых последовательно алгоритмом векторов $\boldsymbol{f^0}$, $\boldsymbol{f^1}$, $\boldsymbol{h^0}$, $\boldsymbol{h^1}$, $\boldsymbol{h^0f^1}$, $\boldsymbol{h^1f^0}$, $\boldsymbol{g}$, $\boldsymbol{f^0g}$, $\boldsymbol{f^1g}$, $\boldsymbol{h^0}$ и $\boldsymbol{h^1}$. Проверка проводится как по начальному f^-, так и по конечному f^+коэффициентам:

f	f^0	f^1	h^0	h^1	h^0f^1	h^1f^0	g	f^0g	f^1g	h^0	h^1	
0-	10	00	10	00	00	00	00	00	00	00	11	**Начальный**
-1	00	01	10	00	00	00	00	00	00	00	11	**коэффициент**
-0	01	00	10	00	00	00	00	00	00	00	11	f^-
11	00	11	10	00	10	00	11	00	11	00	11	
												$h^0h^1 = 0$
			00	11	00	10	11	10	00	11	00	**Конечный**
			00	11	00	00	00	00	00	11	00	**коэффициент**
			00	11	00	01	11	01	00	11	00	f^+
			00	11	00	00	00	00	00	11	00	
												$h^0h^1 = 0$

Триада пригодна.

4. Поиск подходящего разбиения по следу

Если рассматриваемая триада $(p, q)/r$ оказалась пригодной, можно предположить, что она является следом искомого подходящего разбиения. В этом случае последнее можно найти, двигаясь по следу, т. е. используя полученное на предыдущем этапе значение вектора $\boldsymbol{g}$ и последовательно расширяя множества $\boldsymbol{u}$ и $\boldsymbol{v}$ с начальными значениями $\boldsymbol{u} = (p, q)$ и $\boldsymbol{v} = (r)$.

Расширение множества $\boldsymbol{v}$. Начнем с множества $\boldsymbol{v}$. Перебирая последовательно все элементы s из множества $\boldsymbol{x} \setminus (\boldsymbol{u} \cup \boldsymbol{v})$, будем находить среди них такие, при включении которых в множество $\boldsymbol{v}$ разбиение $\boldsymbol{u}/\boldsymbol{v}$ остается пригодным. С этой целью для каждого элемента s выполняются три операции:

$$\boldsymbol{e} := \mathrm{S}_s^{\vee}\boldsymbol{g};$$
$$\boldsymbol{h}^0 := \mathrm{S}_{\boldsymbol{u}}^{\vee}(\boldsymbol{f}^0\boldsymbol{e});$$
$$\boldsymbol{h}^1 := \mathrm{S}_{\boldsymbol{u}}^{\vee}(\boldsymbol{f}^1\boldsymbol{e});$$

и если $\boldsymbol{h}^0\boldsymbol{h}^1 = \boldsymbol{0}$, то s включается в $\boldsymbol{v}$, что реализуется операциями

$$\boldsymbol{v} := \boldsymbol{v} \cup \{s\}; \ \boldsymbol{g} := \boldsymbol{e}.$$

Так находится заключительное значение множества $\boldsymbol{v}$.

Расширение множества $\boldsymbol{u}$. Аналогично находится максимальное расширение множества $\boldsymbol{u}$. Если известно, что искомое разбиение является строгим, можно положить $\boldsymbol{u} = \boldsymbol{x} / \boldsymbol{v}$ и, возможно, проверить разделимость функции по нему, поскольку используемый алгоритм является эвристическим. Заметим, однако, что вероятность получения этим алгоритмом ошибочного решения быстро стремится к нулю с ростом числа переменных n.

Если же искомое разбиение может быть не строгим, следует проверить все элементы из предшествующего значения множества $\boldsymbol{x} \setminus (\boldsymbol{u} \cup \boldsymbol{v})$ и, если это возможно, включить их в множество $\boldsymbol{u}$.

Проверку очередного элемента s будем производить следующим эвристическим алгоритмом, который частично реализует процедуру, описанную в работе [5]. Действия алгоритма ограничиваются тем, что он рассматривает начальный коэффициент f^- разложения функции f по текущему значению множества $\boldsymbol{u}$, отыскивает ортогональные ему коэффициенты, проверяя их на совместимость, и в случае совместимости включает элемент s в множество $\boldsymbol{u}$ без дальнейшей проверки:

$$\begin{aligned}
&\boldsymbol{e} := \boldsymbol{u} \cup \{s\};\\
&\boldsymbol{h}^0 := \boldsymbol{f}^0 - \boldsymbol{e};\\
&\boldsymbol{h}^1 := \boldsymbol{f}^1 - \boldsymbol{e};\\
&\boldsymbol{g} := \mathrm{S}_{\boldsymbol{v}}^{\vee} (\boldsymbol{h}^0 \boldsymbol{f}^1 \vee \boldsymbol{h}^1 \boldsymbol{f}^0);\\
&\boldsymbol{h}^0 := \mathrm{S}_{\boldsymbol{u}}^{\vee} (\boldsymbol{f}^0 \boldsymbol{g});\\
&\boldsymbol{h}^1 := \mathrm{S}_{\boldsymbol{u}}^{\vee} (\boldsymbol{f}^1 \boldsymbol{g}).
\end{aligned}$$

Если $\boldsymbol{h}^0 \boldsymbol{h}^1 = \boldsymbol{0}$, то s включается в $\boldsymbol{u}$, что реализуется операцией $\boldsymbol{u} := \boldsymbol{e}$. Так находится множество $\boldsymbol{u}$ и, следовательно, искомое разбиение $\boldsymbol{u}/\boldsymbol{v}$ в целом.

Заметим, что операция нахождения коэффициента f^- представляется в этом алгоритме сокращенно, выражениями $\boldsymbol{h}^0 := \boldsymbol{f}^0 - \boldsymbol{e}$ и $\boldsymbol{h}^1 := \boldsymbol{f}^1 - \boldsymbol{e}$, вместо более детальных

$$\begin{aligned}
&\boldsymbol{h}^0 := (\ldots((\boldsymbol{f}^0 - e_1) - e_2) - \ldots) - e_t;\\
&\boldsymbol{h}^1 := (\ldots((\boldsymbol{f}^1 - e_1) - e_2) - \ldots) - e_t,
\end{aligned}$$

где $\boldsymbol{e} = (e_1, e_2, \ldots, e_t)$.

Заключение

В статье предложен эвристический алгоритм нахождения такого нестрогого двухблочного разбиения на множестве переменных булевой функции, по которому она декомпозируема. Алгоритм эффективен, если существует хорошее решение, «спрятанное» в векторном представлении функции многих переменных. В этом случае поиск разбиения сводится к его распознаванию.

Работа была поддержана Белорусским республиканским фондом фундаментальных исследований (проект Ф07МС-034).

Список литературы

1. Поваров, Г.Н. О функциональной разделимости булевых функций / Г.И. Поваров // Доклады АН СССР. – Т. 94, № 5. – 1954.

2. Ashenhurst, R.L. The decomposition of switching functions / R.L. Ashenhurst // Proc. International Symposium on the Theory of Switching. Part 1. – Cambridge: Harward University Press, 1959. – P. 75–116.

3. Curtis, H.A. Design of switching circuits / H.A. Curtis. – Van Nostrand, Princeton, N.-J., 1962.

4. Закревский, А.Д. Комбинаторный поиск подходящих разбиений при декомпозиции булевых функций / А.Д. Закревский // Вестник Томского государственного университета. Приложение № 18. – 2006. – С. 4–9.

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

6. Закревский А.Д. Универсальная система для решения задач типа синтеза релейных схем / А.Д. Закревский // Труды СФТИ. – Вып. 42. – 1963. – С. 9–37.

7. Харари, Ф. Теория графов / Ф. Харари. – М.: Мир, 1973.

Поступила 20.02.07

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

A.D. Zakrevskij

DECOMPOSITION OF PARTIAL BOOLEAN FUNCTIONS – FINDING AN APPROPRIATE PARTITION

A novel method is proposed for finding sequential two-block decompositions of partial Boolean functions, generally non-disjunctive. The key problem is considered: finding such a weak partition on the set of arguments at which the regarded function can be decomposed. Its solution is essentially speeded up by means of preliminary discovering traces of the sought-for partition. Boolean and ternary vectors and matrices are used, with efficient parallel combinatorial operations over them.