On Bi-Decompositions of Logic Functions

Tsutomu Sasao  
Department of Computer Science  
and Electronics  
Kyushu Institute of Technology  
Fukuoka 812, Japan

Jon T. Butler  
Department of Electrical and  
Computer Engineering  
Naval Postgraduate School  
Monterey, CA 93943-5121, U.S.A.

Abstract

A logic function $f$ has a disjoint bi-decomposition iff $f$ can be represented as $f = h(g_1(X_1), g_2(X_2))$, where $X_1$ and $X_2$ are disjoint set of variables, and $h$ is an arbitrary two-variable logic function. $f$ has a non-disjoint bi-decomposition iff $f$ can be represented as $f(X_1, X_2, x) = h(g_1(X_1, x), g_2(X_2, x))$, where $x$ is the common variable. In this paper, we show a fast method to find bi-decompositions. Also, we enumerate the number of functions having bi-decompositions.

I Introduction

Functional decomposition is a basic technique to realize economical networks. If the function $f$ is represented as $f(X_1, X_2) = h(g(X_1), X_2)$, then $f$ can be realized by the network shown in Fig. 1.1. To find such a decomposition,

![Figure 1.1: A simple disjoint bi-decomposition chart](image)

2) Some programmable logic devices have two-input logic elements in the outputs.
3) If $f$ has a bi-decomposition, then the optimization of the expression is relatively easy.

A restricted class of bi-decompositions has been considered by [8]. The goals of this paper are
1) Present a fast method for finding bi-decompositions.
2) Enumerate the functions that have bi-decompositions.

Most of the proofs are omitted. They can be available from authors.

II Disjoint Bi-Decomposition

**Definition 2.1** Let $X = (X_1, X_2)$ be a partition of the variables. A logic function $f$ has a disjoint bi-decomposition iff $f$ can be represented as $f(X_1, X_2) = h(g_1(X_1), g_2(X_2))$, where $h$ is any two-variable logic function.

If $f$ has a disjoint bi-decomposition, then $f$ can be realized by the network shown in Fig. 1.2.

**Definition 2.2** Let $X = (X_1, X_2)$ be a partition of the variables. Let $n_1$ and $n_2$ be the number of variables in $X_1$ and $X_2$, respectively. A decomposition chart of the function $f$ for a partition $(X_1, X_2)$ consists of $2^{n_1}$ columns and $2^{n_2}$ rows of 0s and 1s. The $2^{n_1}$ distinct binary numbers for $X_1$ are listed across the top, and the $2^{n_2}$ distinct binary numbers for $X_2$ are listed down the side. The entry for the chart corresponds to the value of $f(X_1, X_2)$.

**Example 2.1** Two decomposition charts for the function $f(x_1, x_2, x_3, x_4) = x_1 x_2 \oplus x_3 x_4$ are shown in Fig. 2.1 (a) and (b).

Note that the decomposition chart is similar to the Karnaugh map with a different ordering for the cell locations.

**Definition 2.3** The number of distinct column (row) patterns in the decomposition chart is called column (row) multiplicity.
A logic function f has a disjoint bi-decomposition if f can be represented as f = h(g1(X1); g2(X2)), where X1 and X2 are disjoint sets of variables, and h is an arbitrary two-variable logic function. f has a non-disjoint bi-decomposition if f can be represented as f(X1; X2; x) = h(g1(X1; x); g2(X2; x)), where x is the common variable. In this paper, we show a fast method to find bi-decompositions. Also, we enumerate the number of functions having bi-decompositions.
Figure 2.1: Decomposition chart.

![Decomposition Chart](image)

**Figure 3.3: Functions in Example 3.2**

**Theorem 3.1** \( f(X_1, X_2, x) \) has a non-disjoint bi-decomposition of the form \( h(g_1(X_1, x), g_2(X_2, x)) \) iff \( f(X_1, X_2, 0) \) and \( f(X_1, X_2, 1) \) have disjoint bi-decompositions \( h(g_1(X_1), g_2(X_2)) \) and \( h(g_1(1), g_2(0)) \), respectively.

**Example 3.1** Consider the three-variable function \( f(x, y, z) = \overline{x} \overline{y} z \vee x y z \). Suppose modules that realize any function of two variables can be used. The straightforward realization shown in Fig. 3.1 requires five modules. The Shannon expansion with respect to \( x \) is

\[
\begin{align*}
000 & \rightarrow 000 & \rightarrow 000 & \rightarrow 000 \\
001 & \rightarrow 101 & \rightarrow 101 & \rightarrow 101 \\
010 & \rightarrow 011 & \rightarrow 011 & \rightarrow 011 \\
011 & \rightarrow 110 & \rightarrow 110 & \rightarrow 110 \\
100 & \rightarrow 000 & \rightarrow 000 & \rightarrow 000 \\
101 & \rightarrow 010 & \rightarrow 010 & \rightarrow 010 \\
110 & \rightarrow 110 & \rightarrow 110 & \rightarrow 110 \\
111 & \rightarrow 111 & \rightarrow 111 & \rightarrow 111
\end{align*}
\]

From this expansion, we have the network in Fig. 3.3. This network requires only three modules.

**Example 3.2** Consider the five-variable function \( f = \overline{x}_5 \overline{x}_0 \vee x_3 \overline{x}_1 \), where \( f_0 \) and \( f_1 \) are shown in Fig. 3.3. Since both \( f_0 \) and \( f_1 \) have disjoint bi-decompositions of the form \( h(g_1(X_1), g_2(X_2)) \), \( f = \overline{x}_5 \overline{x}_0 \vee x_3 \overline{x}_1 \) has a non-disjoint bi-decomposition as follows:

\[
\begin{align*}
f & = \overline{x}_5 \{ \overline{x}_1 \overline{x}_2 \oplus x_3 x_4 \} \vee x_5 \{ x_1 x_2 \oplus (x_3 \oplus x_4) \} \\
& = \{ \overline{x}_5 (\overline{x}_1 \overline{x}_2) \vee x_5 (x_1 x_2) \} \oplus \{ \overline{x}_5 (x_3 x_4) \vee x_5 (x_3 \oplus x_4) \}.
\end{align*}
\]

The converse is true also.

Up to now, we only considered the case where there is a single common variable. However, the theorem can be extended to \( k \) common variables, where \( k \geq 2 \).

**Definition 3.3** Let \( X_1, X_2, \) and \( X_3 \) be disjoint sets of variables. Let \( f(X_1, X_2, a) \) be the sub-functions, where \( X_3 \) is set to \( a \in \{0,1\}^k \), and \( k \) denotes the number of variables in \( X_3 \).

**Theorem 3.2** Let \( X_1, X_2, \) and \( X_3 \) be disjoint sets of variables. Then, \( f \) has a non-disjoint bi-decomposition of form \( f(X_1, X_2, X_3) = h(g_1(X_1, X_3), g_2(X_2, X_3)) \) iff \( f(X_1, X_2, a) \) has a decomposition of the form \( h(g_1(a, X_1), g_2(a, X_2)) \) for all possible \( a \in \{0,1\}^k \), where \( k \) denotes the number of variables in \( X_3 \).
IV A Fast Method for Bi-Decompositions

In this section, we show necessary and sufficient conditions for a function to have a disjoint bi-decomposition. Then, we show efficient algorithms to find disjoint bi-decompositions. In the previous sections, $h(g_1, g_2)$ is an arbitrary two-variable logic function. To find a disjoint bi-decomposition, we need to consider only three types:

1) OR type: $f = g_1(X_1) \lor g_2(X_2)$,
2) AND type: $f = g_1(X_1) \land g_2(X_2)$, and
3) EXOR type: $f = g_1(X_1) \oplus g_2(X_2)$.

Since $f$ has an AND type disjoint bi-decomposition iff $\tilde{f}$ has OR type disjoint bi-decomposition, we only consider the OR type and EXOR type bi-decompositions.

Definition 4.1 $x$ and $\bar{x}$ are literals of a variable $x$. A logical product which contains at most one literal for each variable is called a product term or a product. Product terms with OR operators form a sum-of-products expression (SOP).

Definition 4.2 A prime implicant (PI) $p$ of a function $f$ is a product term which implies $f$, such that the deletion of any literal from $p$ results in a new product which does not imply $f$.

Definition 4.3 An irredundant sum-of-products expression (ISOP) is an SOP, where each product is a PI, and no product can be deleted without changing the function represented by the expression.

Definition 4.4 Let $f(X)$ be a function and $p$ be a product of literals in $X$. The restriction of $f$ to $p$, denoted by $f(X|p)$, is obtained as follows: If $x_i$ appears in $p$, then set $x_i$ in 1 in $f$, and if $\bar{x}_i$ appears in $p$, then set $x_i$ in 0 in $f$.

Example 4.1 Let $f(x_1, x_2, x_3) = x_1x_2 \lor \bar{x}_2x_3$ and $p = x_1x_3$. $f(X|p)$ is obtained as follows: Set $x_1 = x_3 = 1$ in $f$, and we have $f(X|p) = 1 = x_2 \lor \bar{x}_2 = 1$. $\square$

Lemma 4.1 $p$ is an implicant of $f(X)$, if $f(X|p) = 1$.

Example 4.2 By Lemma 4.1, $x_1x_3$ is an implicant of $x_1x_2 \lor \bar{x}_2x_3$, shown in Example 4.1. $\square$

Theorem 4.1 (OR type disjoint bi-decomposition) $f$ has a disjoint bi-decomposition of form $f(X_1, X_2) = g_1(X_1) \lor g_2(X_2)$ if every product in an ISOP for $f$ consists of literals from $X_1$ only or $X_2$ only.

Definition 4.5 $x^0 = \bar{x}$, $x^1 = x$.

Corollary 4.1 If $f(x_1, x_2, \ldots, x_n)$ has a PI of the form $x_1^{a_1}x_2^{a_2}\cdots x_n^{a_n}$, where $a_i \in \{0, 1\}$, then $f$ has no OR type disjoint bi-decomposition.

Let $x_i(i = 1, 2, \ldots, n)$ be the input variables of $f$. Let $p_1 \lor p_2 \lor \cdots \lor p_t$ be an irredundant sum-of-products expression for $f$, where $p_i (i = 1, 2, \ldots, t)$ are PIs of $f$. Let $\Pi_0$ be the trivial partition of $\{1, 2, \ldots, n\}$. $\Pi_0 = \{\{1\}, \{2\}, \ldots, \{n\}\}$.

Algorithm 4.1 (OR type disjoint bi-decomposition: $f(X_1, X_2) = g_1(X_1) \lor g_2(X_2)$)

1. For $i = 1$ to $t$, form $\Pi_i$ from $\Pi_{i-1}$ by merging two blocks $\Omega_1$ and $\Omega_2$ of $\Pi_{i-1}$ if at least one literal in $p_i$ occurs in both $\Omega_1$ and $\Omega_2$.

2. If $\Pi_t$ has at least two blocks, then $f(X_1, X_2)$ has a disjoint bi-decomposition of the form $f(X_1, X_2) = g_1(X_1) \lor g_2(X_2)$, with $X_1$ the union of one or more blocks of $\Pi_0$ and $X_2$ the union of the remaining blocks.

Example 4.3 Consider the ISOP: $f(x_1, x_2, \ldots, x_6) = x_1x_2 \lor x_2x_3 \lor x_4x_5 \lor x_5x_6$. The products $x_1x_2$ and $x_2x_3$ show that $x_1$, $x_2$, and $x_3$ are in the same block. Also, the products $x_4x_5$ and $x_5x_6$ show that $x_1$, $x_5$, and $x_6$ are in the same block. Thus, we have the partition $\{1, 2, 3\}, \{4, 5, 6\}\}$.

The corresponding OR type disjoint bi-decomposition is $f(X_1, X_2) = g_1(X_1) \lor g_2(X_2)$, where $X_1 = (x_1, x_2, x_3)$ and $X_2 = (x_4, x_5, x_6)$. $\square$

Example 4.4 Consider the function $f$ with an ISOP: $f(x_1, x_2, x_3, x_4, x_5) = x_1x_2x_3 \lor x_3x_4x_5$.

1) The product $x_1x_2x_3$ shows that $x_1$, $x_2$, and $x_3$ belong to the same block.

2) The product $x_3x_4x_5$ shows that $x_3$, $x_4$, and $x_5$ belong to the same block.

Thus, all the variables belong to the same block. From this, it follows that $f$ has no OR type decomposition. $\square$

Theorem 4.2 (AND type disjoint bi-decomposition) $f$ has a disjoint bi-decomposition of form $f(X_1, X_2) = g_1(X_1) \land g_2(X_2)$ if every product in an ISOP for $f$ consists of literals from $X_1$ only or $X_2$ only.

Lemma 4.2 [15] An arbitrary $n$-variable function can be uniquely represented as

$$f(x_1, x_2, \ldots, x_n) = g_0 \oplus (a_1x_1 \oplus a_2x_2 \oplus \cdots \oplus a_nx_n)$$

$$\oplus (a_1x_1x_2 \oplus a_3x_1x_3 \oplus \cdots \oplus a_{n-1}x_{n-1}x_n)$$

$$\oplus \cdots \oplus a_{2^n-1}x_1x_2\cdots x_n,$$  

(4.1)

where $a_i \in \{0, 1\}$. The above expression is called a positive polarity Reed-Muller expression (PRPM).

For a given function $f$, the coefficients $g_0, a_1, a_2, \ldots, a_{2^n-1}$ are uniquely determined. Thus, the PRPM is a canonical representation. The number of products in (4.1) is at most $2^n$, and all the literals are positive (unimplemented).
Theorem 4.3 (EXOR type disjoint bi-decomposition) f has a disjoint bi-decomposition of the form \( f(x_1, x_2) = g_1(x_1) \oplus g_2(x_2) \) if every product in the PPRM for f consists of literals from \( X_1 \) only or \( X_2 \) only.

Corollary 4.2 If the PPRM of an n-variable function has the product \( x_1x_2\cdots x_n \), then f has no EXOR type disjoint bi-decomposition.

Theorem 4.4 When f has an EXOR type disjoint bi-decomposition, the number of true minterms of f is an even number.

Corollary 4.3 When the number of true minterms of f is an odd number, then f does not have an EXOR type disjoint bi-decomposition.

The significance of this observation is that at least one half of the functions can be quickly rejected as candidates for EXOR type disjoint bi-decomposition.

Let \( x_i \) (\( i = 1, 2, \ldots, n \)) be the input variables of f.

Let \( p_1 \oplus p_2 \oplus \cdots \oplus p_t \) be PPRM for f, where \( p_i (i = 1, 2, \ldots, t) \) are products. Let \( \Pi_0 \) be the trivial partition of \{1, 2, \ldots, n\}. \( \Pi_0 = \{\{1\}, \{2\}, \ldots, \{n\} \} \).

Algorithm 4.2 (EXOR type disjoint bi-decomposition: \( f(x_1, x_2) = g_1(x_1) \oplus g_2(x_2) \)).

1. For \( i = 1 \) to \( t \), form \( \Pi_i \) from \( \Pi_{i-1} \) by merging two blocks \( \Omega_1 \) and \( \Omega_2 \) of \( \Pi_{i-1} \) if at least one literal in \( p_i \) occurs in both \( \Omega_1 \) and \( \Omega_2 \).

2. If \( \Pi_i \) has at least two blocks, then \( f(X_1, X_2) \) has a disjoint bi-decomposition of form \( f(X_1, X_2) = g_1(x_1) \oplus g_2(x_2) \), with \( X_1 \) the union of one or more blocks of \( \Pi_i \) and \( X_2 \) the union of the remaining blocks.

Example 4.5 Consider the PPRM: \( f(x_1, x_2, \ldots, x_6) = x_1x_2 \oplus x_2x_3 \oplus x_4x_5 \oplus x_5x_6 \). The products \( x_1x_2 \) and \( x_2x_3 \) show that \( x_1 \) and \( x_2 \) are in the same block. Also, the products \( x_4x_5 \) and \( x_5x_6 \) show that \( x_4 \) and \( x_5 \) are in the same block. Thus, we have the partition \( \{\{1, 2, 3\}, \{4, 5, 6\} \} \). The corresponding EXOR type disjoint bi-decomposition is \( f(X_1, X_2) = g_1(x_1) \oplus g_2(x_2) \), where \( X_1 = (x_1, x_2, x_3) \) and \( X_2 = (x_4, x_5, x_6) \).

Algorithm 4.3 (Non-disjoint bi-decomposition).

\( f(x_1, x_2, x_i) = g_1(x_1, x_i) \oplus g_2(x_2, x_i) \), where \( \oplus \) denotes either OR, AND, or EXOR. Let \( (X_1, X_2, x_i) \) be a partition of the variables \( x_1, x_2, \ldots, x_n \). For \( i = 1 \) to \( n \), do

i) Let \( f_0 = f(X_1, x_2, 0) \). (Set \( x_1 \) to \( 0 \).)

ii) If both \( f_0 \) and \( f_1 \) have the same type of disjoint bi-decompositions with the same partition, then \( f \) has a non-disjoint bi-decomposition.

V Complexity Analysis of the Algorithms

5.1 OR type disjoint bi-decomposition

We assume that the function is given as an ISOP with \( t \) products. Note that \( t \leq 2^n-1 \). The time to form the partition of variables is \( O(n \cdot t) \).

5.2 EXOR type disjoint bi-decomposition

A PPRM can be represented by a functional decision diagram (FDD [5, 15]). Each path from the root node to the constant 1 node corresponds to a product in the PPRM.

Thus, the partition of the input variables is directly generated from the FDD. The number of paths in an FDD is \( O(2^n) \), where \( n \) is the number of the input variables.

However, we can avoid exhaustive generation of paths as follows: Let \( p_1 \) and \( p_2 \) be products in a PPRM. If all the literals in \( p_1 \) also appear in \( p_2 \), then \( p_2 \) need not be generated in the Algorithm, since the product \( p_1 \) that contains more literals than \( p_2 \) is more important. By searching the paths with more literals first, we can efficiently detect functions with no disjoint bi-decomposition.

Example 5.1 Consider the function \( f(X) \) given as a PPRM: \( f(X) = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \). In constructing the partition of \( X \), we need not consider the products \( x_1 \) or \( x_2 \), since \( x_1x_2x_3x_6 \) has the literals of \( x_1 \) and \( x_2 \).

In this case, the product \( x_1x_2x_3x_6 \) shows that \( x_1, x_2, x_3, \) and \( x_6 \) belong to the same group. Also, the product \( x_4x_5 \) shows that \( x_3 \) and \( x_4 \) belong to the same group. Thus, \( X \) is partitioned as \( X = (X_1, X_2) \), where \( X_1 = (x_1, x_2, x_3, x_6) \) and \( X_2 = (x_3, x_4) \).

Definition 5.1 Let \( p \) be a product. The set of variables in \( p \) is denoted by \( V(p) = \{x_i \mid x_i \text{ appears in } p\} \). For example, \( V(x_1x_2x_4) = \{x_1, x_2, x_4\} \).

Definition 5.2 Let \( F \) be a PPRM. A product \( p \) is said to have maximal variable set \( V(p) \) if there is no other product \( p' \) such that \( V(p) \subseteq V(p') \).

Example 5.2 For the PPRM, \( F = x_1x_2 \oplus x_1x_3 \oplus x_2x_4 \oplus x_3 \). \( V(x_1x_2) = \{x_1, x_2\} \), \( V(x_1x_3) = \{x_1, x_3\} \), \( V(x_2x_4) = \{x_2, x_4\} \), and \( V(x_3) = \{x_3\} \). Thus, \( x_1x_2x_4 \) and \( x_4 \) have maximal variable sets.

Theorem 5.1 A function \( f \) has an EXOR type disjoint bi-decomposition if a function \( f' \) from the PPRM of \( f \) by eliminating insignificant and having maximal variable sets has an EXOR type disjoint bi-decomposition.

The following theorem says that if a function has an EXOR type disjoint bi-decomposition, then the number of products in the PPRM is relatively small.

Theorem 5.2 If \( f \) has a disjoint bi-decomposition of the form \( f(X, X_2 = g_1(x) \oplus g_2(x) \), then the number of products in the PPRM is at most \( 2^{n_1} + 2^{n_2} - 1 \), where \( n_1 \) is the number of variables in \( X \) (\( i = 1, 2 \)).
VI  Number of Functions with Bi-Decompositions

6.1 Functions with a small number of variables

In the previous sections, we showed that disjoint bi-decompositions are easy to find. In this section, we will enumerate the functions with disjoint bi-decompositions.

Definition 6.1 A function $f$ is said to be nondegenerate if for all $x_i$, $f(\{x_i\}) \neq f(\{x_i\})$.

Definition 6.2 Two functions $f$ and $g$ are NP-equivalent, denoted by $f \sim g$, if $g$ is derived from $f$ by the following operations:

1) Permutation of the input variables.
2) Negations of the input variables.

The following is easy to prove.

Lemma 6.1 If $f$ has a disjoint bi-decomposition and if $f \sim g$, then $g$ has also the same type of disjoint bi-decomposition.

Lemma 6.2 All the two-variable functions have disjoint bi-decompositions.

Example 6.1 There are $2^{2^2} = 256$ three-variable logic functions of which 218 are nondegenerate. These nondegenerate functions are grouped into 16 NP-equivalence classes as shown in Table 6.1 [9]. In this table, the column headed by $N$ denotes the number of functions in that equivalence class. Eight classes have disjoint bi-decompositions, and three have non-disjoint bi-decompositions. Note that 194 functions have bi-decompositions.

The number of functions with AND type disjoint bi-decompositions is equal to the number of functions with OR type disjoint bi-decompositions.

In the case of disjoint bi-decompositions, a function has exactly one type of decomposition (Lemma 6.4). On the other hand, in the case of non-disjoint bi-decompositions, a function may have more than one type of bi-decompositions.

Example 6.2 Consider the three-variable function $f = x_1 \land x_3 \lor x_1 \land x_2$. This function has three types of non-disjoint bi-decompositions:

$f = x_1 \land x_3 \lor x_1 \land x_2$ (OR type bi-decomposition)
$= x_1 \lor x_3 \lor x_1 \lor x_2$ (EXOR type bi-decomposition)
$= (x_1 \lor x_3) \land (x_1 \lor x_2)$ (AND type bi-decomposition)

Table 6.1: NP-representative functions of three variables.

<table>
<thead>
<tr>
<th>Representative functions</th>
<th>$N$</th>
<th>Type</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 \land x_2 \land x_3$</td>
<td>2</td>
<td>EXOR</td>
<td>Disjoint</td>
</tr>
<tr>
<td>$x_1 \land x_2 \lor x_3$</td>
<td>8</td>
<td>AND</td>
<td>Bi-Decomposition</td>
</tr>
<tr>
<td>$x_1 \lor x_2 \land x_3$</td>
<td>8</td>
<td>OR</td>
<td>Bi-Decomposition</td>
</tr>
<tr>
<td>$x_1 \lor x_2 \lor x_3$</td>
<td>24</td>
<td>AND</td>
<td>Bi-Decomposition</td>
</tr>
<tr>
<td>$x_1 \land x_2 \lor x_3$</td>
<td>4</td>
<td>Non-Disjoint</td>
<td>Bi-Decomposition</td>
</tr>
<tr>
<td>$x_1 \lor x_2 \land x_3$</td>
<td>8</td>
<td>No</td>
<td>Bi-Decomposition</td>
</tr>
</tbody>
</table>

$N$: Number of the functions in the class.

Table 6.2: Number of functions.

<table>
<thead>
<tr>
<th>Functions with bi-decomposition</th>
<th>Non-Disjoint</th>
<th>Disjoint AND</th>
<th>OR</th>
<th>EXOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 2$</td>
<td>16</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>$n = 3$</td>
<td>256</td>
<td>44</td>
<td>44</td>
<td>26</td>
</tr>
<tr>
<td>$n = 4$</td>
<td>6536</td>
<td>6500</td>
<td>6500</td>
<td>914</td>
</tr>
<tr>
<td>Total</td>
<td>10</td>
<td>194</td>
<td>8094</td>
<td></td>
</tr>
</tbody>
</table>

6.2 The number of functions with bi-decompositions

Lemma 6.3 [4]: Let $a(n)$ be the number of nondegenerate functions on $n$ variables. Then,

$$a(n) = \sum_{k=0}^{n} \left( \binom{n}{k} \right) (-1)^{n-k} 2^k \sim 2^n,$$

where $a(n) \sim b(n)$ means $\lim_{n \to \infty} \frac{a(n)}{b(n)} = 1$.

Lemma 6.4 A nondegenerate function $f$ has at most one type of disjoint bi-decomposition:

1. $f(X_1, X_2) = g_1(X_1) \cdot g_2(X_2)$,
2. $f(X_1, X_2) = g_1(X_1) \lor g_2(X_2)$, or
3. $f(X_1, X_2) = g_1(X_1) \lor g_2(X_2)$,

where $g_1$ and $g_2$ are nondegenerate functions on one or more variables.

Theorem 6.1 The number of functions $N_{\text{disjoin}}(n)$ with disjoint bi-decompositions is $N_{\text{disjoin}}(n) = A_{\text{dis}}(n) + O_{\text{dis}}(n) + E_{\text{dis}}(n)$, where

$$A_{\text{dis}}(n) = \sum \prod_{i=1}^{n} \frac{a(i) - A_{\text{dis}}(i)}{i} \frac{1}{k_i!}$$
Table 7.1: Number of functions with bi-decompositions.

<table>
<thead>
<tr>
<th>Decomposition Type</th>
<th>Number of Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disjoint</td>
<td>AND: 853 OR: 264 EXOR: 73</td>
</tr>
<tr>
<td>Non-disjoint</td>
<td>AND: 162 OR: 91 EXOR: 42</td>
</tr>
</tbody>
</table>

\[
O_{dis}(n) = n! \sum_{k_1, k_2, \ldots, k_n \geq 0, \sum k_i = n} \frac{n!}{i_1! \cdots i_n!} k_1^{i_1} k_2^{i_2} \cdots k_n^{i_n} \\
E_{dis}(n) = 2n! \sum_{k_1, k_2, \ldots, k_n \geq 0, \sum k_i = n} \frac{n!}{i_1! \cdots i_n!} k_1^{i_1} k_2^{i_2} \cdots k_n^{i_n} \cdot \frac{1}{2^{i_1} k_1!} 
\]

where the sums are over all partitions of \( n \) except the trivial partition \( n = 0 \cdot 1 + 0 \cdot 2 + \cdots + 0 \cdot (n-1) + 1 \cdot n \) (i.e., the sum is over all partitions where \( k_n = 0 \)), and where \( A_{dis}(1) = O_{dis}(1) = E_{dis}(1) = 0 \).

Table 6.2 shows the number of functions with disjoint bi-decompositions up to \( n = 4 \).

VII Experimental Results

We analyzed the bi-decomposability of 136 benchmark functions. Over these multiple-output functions, the total number of outputs (functions) is 1908. For each function, we determined whether there exists a disjoint bi-decomposition. If none existed, we determined if there exists a non-disjoint bi-decomposition (with a single common variable). Table 7.1 summarizes our results. It is interesting that 1190 out of 1908 functions, or 62 percent, have disjoint bi-decompositions. Of the remaining 718 functions, 295 have non-disjoint bi-decompositions. It should be noted that more than 295 functions have disjoint bi-decompositions, since a function with a disjoint bi-decomposition may also have a non-disjoint bi-decomposition.

VIII Conclusions and Comments

In this paper, we presented the bi-decomposition, a special case of functional decomposition. Disjoint bi-decompositions have the following features:

1) They are easy to detect; we use ISOPs or PPRMs rather than decomposition charts.

2) Programmable logic devices exist that realize bi-decompositions.

3) If the function has an OR (AND) type bi-decomposition, then we can optimize the expression separately.

We enumerated functions with bi-decompositions. Among 218 nondegenerate functions of 4 variables, 194 have bi-decompositions. Also, we derived formulae for the number of disjoint bi-decompositions.

Since the fraction of functions with decompositions approaches to zero as \( n \) increases [4], the fraction of functions with bi-decompositions also approaches to zero as \( n \) increases. However, for 1908 functions we analyzed about 78% of them had either disjoint or non-disjoint bi-decompositions.

References


