BLANKET ALGEBRA FOR MULTIPLE-VALUED FUNCTION DECOMPOSITION

J. A. Brzozowski and J. J. Lou*
Department of Computer Science
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1

ABSTRACT

We consider functions of the form $f : E^n \to D^m$, where $n, m > 0$ are integers, E is a finite, nonempty set, and $D = 2^E - \{\emptyset\}$. Such a function has n inputs, $x_1, \ldots, x_n$, each input taking values from E, and m outputs, $y_1, \ldots, y_m$, which are only partially specified. In general, f assigns to any output y_i a nonempty set $d \in D$ of elements of E. If $d = \{e\}$ for some $e \in E$, then y_i has the value e. If $d = E$, then y_i is a "complete don't care", i.e., is allowed to take on any value in E. In other cases, y_i is a "limited don't care", allowed to take on any value in d. Moreover, the functions are represented in a compact notation, which is a generalization of the cube notation for Boolean functions. In this representation, we call these functions "set functions." We consider the problem of decomposing a set function $f(X)$ in the form $h(U, g(V))$, where $X = \{x_1, \ldots, x_n\}$ is the set of input variables, U and V are two subsets of X whose union is X, and g and h are set functions. It has been previously shown that Boolean function decompositions can be obtained with the aid of blankets, which are a generalization of set systems. In this paper we extend blanket algebra methods to the decomposition of multiple-valued set functions.

1. Introduction

Decomposition of Boolean functions has been studied since the late 1950s, because it has many practical applications. In 1959 Ashenhurst studied the so-called disjunctive decompositions of the form $f(X) = h(U, g(V))$, where $U \cap V = \emptyset$; he used charts to represent Boolean functions [1]. These methods were extended by Curtis to decompositions of a function in terms of several functions [4]. A more compact representation of Boolean functions in terms of cubes was introduced by Roth and Karp [10]. The cube cover method has been extended to multiple-output functions in the 1980's [2]. More recently, Łuba and Selvaraj developed a generalized approach to Boolean function decomposition using set systems and their generalizations [8]. This approach has been formalized by Brzozowski and Łuba, who introduced the term

*This research was supported by the Natural Sciences and Engineering Research Council of Canada under grant No. OGP0000871. Manuscript completed June 26, 1997.

"blankets" for certain types of generalized set systems [3]. For a discussion of other works related to Boolean function decomposition and its applications see [3].

Decompositions of multiple-valued functions and relations have been studied recently in [5, 7, 9]; for a discussion of the literature in this field we refer the reader to these papers. In [7] "rough partitions" (blankets) are used to represent functions. The authors of [5, 9] use "generalized don't cares," i.e., situations in which an output may take on any value in some set that is a proper subset of the domain of that output. For example, for a domain of $\{0, 1, 2\}$, an output may be allowed to take the values 0 or 2, but not 1.

The present paper extends the work in [3] to multiple-valued, multiple-output functions with generalized don't care entries; such functions are represented by a compact "cube" notation. Blankets are used to characterize decompositions of $f(X)$ in the form $h(U, g(V))$.

2. Set Vectors

Let E be a finite, nonempty set, and let $D = 2^E - \{\emptyset\}$. An element $d \in D$ represents the set of possible values of a variable x. If $d = \{e\}$ for some element e in E, then x can only take the value e. If $d = E$, then x can take any value in the set E, and we can treat d as a "complete don't care" value. Let $|S|$ denote the cardinality of a set S. If $1 < |d| < |E|$, then d can be viewed as a "limited don't care" value. For example, let $E = \{0, 1, 2\}$; then a variable x over E can take on the values 0, 1, or 2, or it can be specified as a complete don't care $\{0, 1, 2\}$ or it may be one of the limited don't cares $\{0, 1\}$, $\{0, 2\}$, and $\{1, 2\}$.

The set D of possible domain elements is partially ordered by set inclusion. If the intersection of a set of elements of D is nonempty, then that intersection is the greatest lower bound *glb* of that set of elements.

Let D^n be the Cartesian product of n copies of D. For $n \geq 1$, an element of D^n will be called a *set vector*. In general, n-tuples will be denoted by unsubscripted letters, for example t, and their components by subscripted letters, for example t_i. Also, to simplify the notation we often write tuples of values without parentheses and commas. For example, $(\{0, 2\}, \{2\}, \{0, 1\})$ will be written $\{0, 2\}\{2\}\{0, 1\}$, and represents a possible value for a 3-tuple $x = (x_1, x_2, x_3)$ of variables.

The partial order of set inclusion is extended to D^n:

$$t \subseteq t' \text{ if and only if } t_i \subseteq t'_i \text{ for all } i,\ 1 \leq i \leq n,$$

where $t = (t_1, \ldots, t_n)$ and $t' = (t'_1, \ldots, t'_n)$. For example,

$$\{0, 2\}\{2\}\{0, 1\}) \subseteq \{0, 2\}\{1, 2\}\{0, 1, 2\}.$$

We extend intersection to D^n by defining it to be the component-by-component intersection. Such an intersection is empty if any one of its components is empty. A subset

T of D^n is said to be *compatible* if $\bigcap_{t \in T} t \neq \emptyset$. If T is compatible then the greatest lower bound of T exists and is equal to $\bigcap_{t \in T} t$. We denote the *glb* of T as shown below to stress the fact that the *glb* is equal to the intersection only if the intersection is nonempty.

$$glb\, T = \bigsqcap_{t \in T} t.$$

3. Blankets

In the 1960's Hartmanis and Stearns developed an algebra of partitions for decomposition of sequential machines [6]. They also generalized their partition algebra to set systems, which we define later. Blankets are a generalization of set systems; they were formally introduced recently by Brzozowski and Łuba for decomposition of Boolean functions specified by cubes [3]. Below, we briefly review the basic concepts from [3], and generalize them to the multiple-valued case.

A *blanket* on a set S is a collection $\beta = \{B_1, \ldots, B_k\}$ of nonempty and distinct subsets of S, called *blocks*, whose union is S. For example, if $S = \{1, 2, 3\}$, then $\beta = \{\{1\}, \{1, 2\}, \{2, 3\}\}$ is a blanket on S. To improve the notation, we often write $\beta = \{\overline{1};\ \overline{1,2};\ \overline{2,3}\}$. Also, when it is possible to avoid reference to the number k of blocks in a blanket $\beta = \{B_1, \ldots, B_k\}$, we simply write $\beta = \{B_i\}$.

Define the "nonempty" operator ne as follows: For any set $\{S_i\}$ of subsets of a set S,

$$ne\,\{S_i\} = \{S_i\} - \{\emptyset\}.$$

The *product* $\beta{*}\beta'$ of two blankets is

$$\beta{*}\beta' = ne\,\{B_i \cap B_j \mid B_i \in \beta \text{ and } B_j \in \beta'\}.$$

For example,

$$\{\overline{1,2,3};\ \overline{3,4,5}\}{*}\{\overline{1,2,3};\ \overline{3,4,5}\} = \{\overline{1,2,3};\ \overline{3};\ \overline{3,4,5}\};$$

this shows that blanket product is not idempotent.

Let $\mathcal{B}_S$ be the set of all blankets on a set S. Let $1_S = \{S\}$ be the one-block blanket, and let $0_S = \{\{s_i\} \mid s_i \in S\}$ be the blanket consisting of one-element blocks. Then the algebraic system $\langle \mathcal{B}_S, *, 1_S, 0_S \rangle$ is a commutative monoid with unit 1_S and zero 0_S, that is, for all β, β', β'', we have $\beta{*}\beta' = \beta'{*}\beta$, $(\beta{*}\beta'){*}\beta'' = \beta{*}(\beta'{*}\beta'')$, $\beta{*}1_S = 1_S{*}\beta = \beta$, and $\beta{*}0_S = 0_S{*}\beta = 0_S$.

If β and β' are blankets on S, we write $\beta \leq \beta'$ if for each B_i in β there exists a B_j in β' such that $B_i \subseteq B_j$. For example, $\{\overline{1};\ \overline{2};\ \overline{3}\} \leq \{\overline{1,2};\ \overline{2,3}\}$, and $\{\overline{1,2};\ \overline{2,3}\} \leq \{\overline{1,2};\ \overline{2,3};\ \overline{1,3}\}$. The relation $\leq$ is reflexive and transitive, but not antisymmetric, since $\{\overline{1};\ \overline{1,2}\} \leq \{\overline{1,2}\}$ and $\{\overline{1,2}\} \leq \{\overline{1};\ \overline{1,2}\}$. Also, we have the following property:

$$\beta{*}\beta' \leq \beta \text{ and } \beta{*}\beta' \leq \beta'.$$

A *set system* [6] on a set S is a blanket $\sigma = \{B_1, \ldots, B_k\}$ in which the blocks are maximal in the sense that $B_i \subseteq B_j$ implies $i = j$. Note that set systems are not closed under blanket product. For example $\{\overline{1,2};\ \overline{2,3}\}*\{\overline{1,2};\ \overline{2,3}\}$ is not a set system. If $\beta = \{B_i\}$ is any blanket on S, then

$$max\,\beta = \{B \subseteq S \mid B = B_i \text{ for some } i, \text{ and } B \subseteq B_j \text{ implies } B_j = B\}.$$

Note that max maps blankets to set systems. The *product* $\sigma \circ \sigma'$ of two set systems is defined as follows:

$$\sigma \circ \sigma' = max\,(\sigma*\sigma').$$

A *partition* on a set S is a set system $\pi = \{B_i\}$, where the blocks are disjoint, that is, $B_i \cap B_j = \emptyset$, if $i \neq j$.

A *set matrix* is a matrix M of elements of D. In the sequel, we will be using blankets on rows of set matrices. Let M be any $h \times k$ matrix with entries from D. Assume that the rows of M are numbered $1, \ldots, h$. Let T be the set of rows of M; then we denote blankets on T by blankets on the set $\{1, \ldots, h\}$.

A *singleton vector* is a set vector $e \in D^n$ whose components are all singletons, i.e., which has the form $e = \{\{e_1\}, \ldots, \{e_n\}\}$, $e_1, \ldots, e_n \in E$. Let $\Sigma = \{\{e_i\} \mid e_i \in E\}$; then the set of all the singleton vectors in D^n is Σ^n. For a singleton vector e, we define the corresponding *base vector* $\dot{e}$ to be $\dot{e} = \{e_1, \ldots, e_n\}$.

Given a set matrix M, we define a *row blanket* $\beta = ne\{T_{\supseteq e}\}$, where $e \in \Sigma^k$ and

$$T_{\supseteq e} = \{t \in T \mid t \supseteq e\}.$$

Table 1: Matrix M.

1	$\{1\}$	$\{0,1,2\}$	$\{0\}$
2	$\{1\}$	$\{1\}$	$\{0,2\}$
3	$\{0,1,2\}$	$\{0,1\}$	$\{0\}$
4	$\{1\}$	$\{0,1\}$	$\{0\}$
5	$\{0\}$	$\{0,1\}$	$\{1,2\}$

For example, consider matrix M in Table 1. One verifies that $T_{\supseteq 000} = T_{\supseteq 010} = T_{\supseteq 200} = T_{\supseteq 210} = \{3\}$, $T_{\supseteq 001} = T_{\supseteq 002} = T_{\supseteq 011} = T_{\supseteq 012} = \{5\}$, $T_{\supseteq 100} = \{1,3,4\}$, $T_{\supseteq 110} = \{1,2,3,4\}$, $T_{\supseteq 112} = \{2\}$, $T_{\supseteq 120} = \{1\}$, and there are no other nonempty blocks. Blanket β is therefore

$$\beta = \{\overset{B_1}{\overline{3}};\ \overset{B_2}{\overline{5}};\ \overset{B_3}{\overline{1,3,4}};\ \overset{B_4}{\overline{1,2,3,4}};\ \overset{B_5}{\overline{2}};\ \overset{B_6}{\overline{1}}\},$$

where the blocks have been numbered as shown.

The *representative* r_i of block B_i of a row blanket β is the *glb* of the vectors in B_i. Thus, in our example, $r_1 = \{0,1,2\}\{0,1\}\{0\}$, $r_2 = \{0\}\{0,1\}\{1,2\}$, $r_3 = \{1\}\{0,1\}\{0\}$, $r_4 = \{1\}\{1\}\{0\}$, $r_5 = \{1\}\{1\}\{0,2\}$, and $r_6 = \{1\}\{0,1,2\}\{0\}$.

Let R be the set of block representatives of β; in our example, $R = \{r_1, \ldots, r_6\}$. One verifies that the mapping $\mu : \beta \rightarrow R$ associating with each block B of β its representative r is a one-to-one correspondence. In fact, we have the following result:

Proposition 3.1. *Let β be the row blanket of a matrix M. Let $\langle \beta, \supseteq \rangle$ be the set of blocks of β partially ordered by $\supseteq$, and let $\langle R, \subseteq \rangle$ be the set of representatives partially ordered by $\subseteq$. Then the two partially ordered sets are isomorphic.*

It is a consequence of the proposition that $r_i \subseteq r_j$ if and only if $B_i \supseteq B_j$. Two representatives r_i and r_j have a *glb* if and only if $B_i \cup B_j$ is a block of β. In fact, $glb\{r_i, r_j\} = r_l$, if and only if $B_i \cup B_j = B_l$. For example, $glb\{r_1, r_6\} = r_3$ and $B_1 \cup B_6 = B_3$.

4. Set Functions

We will be dealing with functions of the form

$$f : \mathcal{S} \rightarrow D^m,$$

where $\mathcal{S} \subseteq D^n$. Such a function will be called a *set function*, and will be represented by a set matrix M with $n+m$ columns. The set of rows of M will be denoted by F. For an $(n+m)$-tuple $t = (t_1, \ldots, t_{n+m}) \in F$, the *input projection* of t is $t_\downarrow = (t_1, \ldots, t_n)$, and the *output projection* is $t^\uparrow = (t_{n+1}, \ldots, t_{n+m})$. *To reduce the number of rows in the representation of a set function we omit from F all the rows in which $t^\uparrow = E^n$, since all such rows correspond to complete don't cares of all the outputs.*

A set function is said to be *consistent* if it satisfies the following condition: For any set T of rows from F, if $\{t_\downarrow \mid t \in T\}$ is compatible, then $\{t_\uparrow \mid t \in T\}$ is compatible. *From now on we consider only consistent set functions.*[1]

A singleton input vector $e \in \Sigma^n$ is *involved* in a set vector $t \in F$, if $e \subseteq t_\downarrow$. We call e *relevant to* f if e is involved in some vector $t \in F$.

With each set function f we associate a *base function* $\hat{f} : E^n \rightarrow D^m$, as follows: For a singleton vector $e \in \Sigma^n$, i.e., for $\dot{e} \in E^n$, if e is relevant to f, then

$$\hat{f}(\dot{e}) = \bigsqcap_{t \in F,\, t_\downarrow \supseteq e} t^\uparrow.$$

In case e is not relevant to f, we define $\hat{f}(\dot{e}) = E^m$. In other words, the *glb* of the empty set of vectors is interpreted as E^m.

A set function, or its associated base function, is thought of as describing an incompletely specified component with n inputs, $x_1, \ldots, x_n$, and m outputs, $y_1, \ldots, y_m$.

[1] A comment concerning terminology is in order. In [5] functions are allowed to have either specific values or complete don't cares, but not limited don't cares. We consider limited don't cares a natural extension of complete don't cares, and we use the term "function" as long as the specification is consistent.

Table 2: Set function f.

Row	x_1	x_2	x_3	f_1	f_2
1	0	0	0, 2	0, 1	0
2	1	1, 2	0, 2	0, 2	1
3	2	0, 1, 2	0	1, 2	2
4	1, 2	0, 2	0, 2	0, 1	1, 2

Example 4.1. *In Table 2 we illustrate a set function with three inputs and two outputs. For simplicity, we omit the curly brackets around the sets in D. The underlying set is $E = \{0, 1, 2\}$. The base function $\hat{f}$ associated with f is shown in Table 3, which shows all the rows, except those for which both $\hat{f}_1$ and $\hat{f}_2$ are complete don't cares. To illustrate the construction of $\hat{f}$, we show three calculations. The base vector* 000 *is contained only in Row 1; hence $\hat{f}$ has the value $\{0, 1\}\{0\}$ as specified in Row 1. The base vector* 001 *is not contained in any row; hence it is absent from the table, meaning that the value of $\hat{f}$ is the complete don't care $\{0, 1, 2\}\{0, 1, 2\}$. The base vector* 120 *appears in Rows 2 and 4; hence the value of $\hat{f}$ is the glb of the values of f specified in Rows 2 and 4, i.e., $\hat{f}(1, 2, 0) = \{0\}\{1\}$.*

Table 3: Base function $\hat{f}$.

x_1	x_2	x_3	$\hat{f}_1$	$\hat{f}_2$
0	0	0	0, 1	0
0	0	2	0, 1	0
1	0	0	0, 1	1, 2
1	0	2	0, 1	1, 2
1	1	0	0, 2	1
1	1	2	0, 2	1
1	2	0	0	1
1	2	2	0	1
2	0	0	1	2
2	0	2	0, 1	1, 2
2	1	0	1, 2	2
2	2	0	1	2
2	2	2	0, 1	1, 2

For simplicity, we have chosen to use the same domain E for each variable. In general, variables may take their values from different domains. However, this case can be also handled in our model. We can use the largest of the domains as E and relabel the elements of the other domains (such as E', say) so that those domains become subsets of E. The values of the outputs corresponding to inputs in $E - E'$ can be treated as complete don't cares.

Let f be a set function having F as its set of rows. Suppose $V = \{x_{i_1}, \ldots, x_{i_r}\}$ is a subset of the set X of input variables, and t is a set vector in F; denote by $t_\downarrow^V$ the r-tuple $(t_{i_1}, \ldots, t_{i_r})$. Define blanket β_V as follows: $\beta_V = ne\{F_{V \supseteq e}\}$, where $e \in \Sigma^r$, and

$$F_{V \supseteq e} = \{t \in F \mid t_\downarrow^V \supseteq e\}.$$

We also define the output blanket β_f of f as $\beta_f = ne\{F_{Y \supseteq e}\}$, where $e \in \Sigma^m$ and

$$F_{Y \supseteq e} = \{t \in F \mid t^\uparrow \supseteq e\}.$$

For the example of Table 2, suppose $U = \{x_1\}$ and $V = \{x_2, x_3\}$. Then

$$\beta_U = \{\overline{1};\ \overline{2,4};\ \overline{3,4}\},$$

$$\beta_V = \{\overline{1,3,4};\ \overline{1,4};\ \overline{2,3};\ \overline{2};\ \overline{2,3,4};\ \overline{2,4}\},$$

and

$$\beta_f = \{\overline{1};\ \overline{2,4};\ \overline{4};\ \overline{3,4};\ \overline{2};\ \overline{3}\}.$$

5. Serial Separations

We use the following notation. Let $X = \{x_1, \ldots, x_n\}$ be the set of input variables of a set function F with $n > 0$ inputs and $m > 0$ outputs. Let U and V be two subsets of X such that $U \cup V = X$. Without loss of generality, we assume the variables $x_1, \ldots, x_n$ have been relabeled in such a way that $U = \{x_1, \ldots, x_r\}$ and $V = \{x_{n-s+1}, \ldots, x_n\}$. Consequently, for an n-tuple x, the first r components are denoted by x^U, and the last s components, by x^V.

Definition 5.1. *Let g be a set function with $s \geq n - r$ inputs and p outputs, and let h be a set function with $r + p$ inputs and $m > 0$ outputs. The pair (g, h) is a* serial separation *of f* with respect to (U, V), *if, for every singleton vector $e \in \Sigma^n$ relevant to f, $\hat{g}(\dot{e}^V)$ is a singleton vector in Σ^p, the vector $(e^U, \hat{g}(\dot{e}^V))$ is relevant to h, and, if $\hat{g}(\dot{e}^V) = k$, then*

$$\hat{f}(\dot{e}) \supseteq \hat{h}(\dot{e}^U, \dot{k}). \tag{1}$$

A serial separation is called disjoint *if $U \cap V = \emptyset$.*

Example 5.2. *Consider set function f' specified by Table 4. This function has four inputs and two outputs, and the underlying set is $E = \{0, 1, 2\}$. Let $U = \{x_1, x_2\}$ and $V = \{x_2, x_3, x_4\}$. Function g' of Table 5 and function h' of Table 6 constitute*

Table 4: Set function f'.

Row	x_1	x_2	x_3	x_4	f_1'	f_2'
1	$0,1$	0	0	$0,1$	1	0
2	1	0	$1,2$	2	0	$1,2$
3	0	0	$0,1$	0	$1,2$	0
4	$0,2$	$1,2$	2	2	$0,1$	2
5	0	$0,2$	1	1	2	$1,2$
6	0	$0,1$	0	2	1	1
7	1	$0,2$	0	2	0	0
8	2	0	0	2	1	2

a (serial) separation of f'. To verify this, first we check that g' and h' satisfy the consistency condition. Next, we find input vectors in Σ^4 that are relevant to f'. For each of these vectors consider the (x_2, x_3, x_4) components. For each of these 3-tuples, $\hat{g}'$ is a singleton, as required. It remains to verify (1). The first relevant input vector is $(x_1, x_2, x_3, x_4) = 0000$, and $\hat{f}'(0000) = \{1\}\{0\}$, from Table 4. From Table 5, we have $\hat{g}'(000) = 0$, and from Table 6, we find $\hat{h}'(000) = \{1\}\{0\}$. Here we have equality, and (1) is satisfied. For $(x_1, x_2, x_3, x_4) = 0010$, we find $\hat{f}'(0010) = \{1,2\}\{0\}$, $\hat{g}'(010) = 0$, and $\hat{h}'(000) = \{1\}\{0\}$; thus (1) is again satisfied, though this time the containment is proper. The rest of the entries are verified similarly.

Table 5: Function g'.

x_2	x_3	x_4	g'
0	0	$0,1$	0
0	$1,2$	2	1
0	$0,1$	0	0
$1,2$	2	2	0
$0,2$	1	1	1
$0,1,2$	0	2	2

Theorem 5.3. *Let $f(X)$ be a set function specified by a set F of set vectors, and let (U, V) be a pair of subsets of X satisfying $U \cup V = X$. For every blanket β_g satisfying the conditions stated below, there is a serial separation.*

- $\beta_V \leq \beta_g$, *and*
- $\beta_U * \beta_g \leq \beta_f$,

Table 6: Function h'.

x_1	x_2	g'	h'_1	h'_2
0, 1	0	0	1	0
1	0	1	0	1, 2
0, 2	1, 2	0	0, 1	2
0	0, 2	1	2	1, 2
0	0, 1	2	1	1
1	0, 2	2	0	0
2	0	2	1	2

Proof. Suppose β_g satisfying the conditions of the theorem is given and has q blocks. We encode the blocks of β_g uniquely by assigning to each block B a combination $\alpha(B)$ of p variables, where p is the desired number of outputs of function g. We could always use one output with q values, but it may be desirable to reduce the size of the domain of this variable. In that case we can use more variables, each with fewer values.

For each singleton vector $e \in D^s$, we find the set vectors t in F in which e is involved. If there are no such vectors, $\dot{e}$ is omitted from the table for $\hat{g}$. Otherwise, the set vectors t define a block B' of β_V. Next, we find any block B of β_g that contains B', and associate to e the code word $\alpha(B)$ assigned to B. In this manner we define $\hat{g}$.

Next we construct the table for $\hat{h}$. For each $a \in E^r$ and $b \in E^p$ we find a block of $\beta_U {*} \beta_g$, if such a block exists. By the condition of the theorem, each such block is contained in a block B'' of β_f. Then we assign $\bigsqcap_{t \in B''} t^{\uparrow}$ to the input vector ab of $\hat{h}$.

The proof that this construction is correct follows very closely the proof of correctness of a similar construction for the Boolean function case, and we refer the interested reader to [3].

Example 5.4. *Consider function f' from Table 4. Suppose $U = \{x_1, x_2\}$ and $V = \{x_2, x_3, x_4\}$. Also suppose that we are given blanket $\beta_{g'} = \{\overline{1,3,4};\ \overline{2,5};\ \overline{6,7,8}\}$. Note that for this function we have*

$$\beta_U = \{\overline{1,3,5,6};\ \overline{4,6};\ \overline{4,5};\ \overline{1,2,7};\ \overline{7};\ \overline{8};\ \overline{4}\},$$

$$\beta_V = \{\overline{1,3};\ \overline{1};\ \overline{6,7,8};\ \overline{3};\ \overline{5};\ \overline{2};\ \overline{6};\ \overline{4};\ \overline{7}\},$$

and

$$\beta_{f'} = \{\overline{7};\ \overline{2};\ \overline{2,4};\ \overline{1,3};\ \overline{6};\ \overline{4,8};\ \overline{3};\ \overline{5}\}.$$

We can check that $\beta_{g'}$ satisfies the conditions of the theorem. The blocks of $\beta_{g'}$ can be coded by a single 3-valued variable. The coding is shown below.

$$\beta_{g'} = \{\overset{0}{\overline{1,3,4}};\ \overset{1}{\overline{2,5}};\ \overset{2}{\overline{6,7,8}}\}.$$

From Table 4 we find all the possible values relevant to g'; these are shown in the first three columns of Table 7. The fourth column lists the block of β_V corresponding to each input combination. The fifth column lists blocks of $\beta_{g'}$ containing the blocks of β_V; in this case there is no choice, since $\beta_{g'}$ is a partition. The sixth column lists the corresponding output values. One verifies that this is the same function as that of Table 5.

Table 7: Finding function $\hat{g}'$.

x_2	x_3	x_4	β_V	$\beta_{g'}$	$\hat{g}'$
0	0	0	$\overline{1,3}$	$\overline{1,3,4}$	0
0	0	1	$\overline{1}$	$\overline{1,3,4}$	0
0	0	2	$\overline{6,7,8}$	$\overline{6,7,8}$	2
0	1	0	$\overline{3}$	$\overline{1,3,4}$	0
0	1	1	$\overline{5}$	$\overline{2,5}$	1
0	1	2	$\overline{2}$	$\overline{2,5}$	1
0	2	2	$\overline{2}$	$\overline{2,5}$	1
1	0	2	$\overline{6}$	$\overline{6,7,8}$	2
1	2	2	$\overline{4}$	$\overline{1,3,4}$	0
2	0	2	$\overline{7}$	$\overline{6,7,8}$	2
2	1	1	$\overline{5}$	$\overline{2,5}$	1
2	2	2	$\overline{4}$	$\overline{1,3,4}$	0

Example 5.5. *The construction of function h'' is illustrated in Table 8. For the input vector $(x_1, x_2, x_3, x_4) = 0000$ relevant to f', the value of $\hat{g}'$ from Table 7 is 0; hence 000 is a relevant input vector for h''. The block of β_U corresponding to $(x_1, x_2) = 00$ is $\overline{1,3,5,6}$, and the block of $\beta_{g'}$ corresponding to 0 is $\overline{1,3,4}$. The block of $\beta_U * \beta_{g'}$ is $\overline{1,3}$, which is contained only in block $\overline{1,3}$ of $\beta_{f'}$. Hence the value of $\hat{h}''$ is $\{1\}\{0\}$, as given in Table 4, Rows 1 and 3.*

For $(x_1, x_2, x_3, x_4) = 1022$, the value of $\hat{g}'$ is 1; hence 101 is relevant to h''. The corresponding block of $\beta_U * \beta_{g'}$ is $\overline{2}$, which is contained in two blocks of β'_f. Block $\overline{2,4}$ corresponds to output $\{0\}\{2\}$, and block $\overline{2}$ corresponds to output $\{0\}\{1,2\}$. We choose the last block, since it maximizes the don't cares.

For $(x_1, x_2, x_3, x_4) = 0122$, the value of $\hat{g}'$ is 0; hence 010 is relevant to h''. The corresponding block of $\beta_U * \beta_{g'}$ is $\overline{4}$, which is contained in two blocks of β'_f. Block $\overline{2,4}$ corresponds to output $\{0\}\{2\}$, and block $\overline{4,8}$ corresponds to output $\{1\}\{2\}$. Arbitrarily, we choose the first block.

The rest of Table 8 is computed similarly. One verifies that this is not *the same function as that of Table 6.*

Table 8: Finding function $\hat{h}''$.

x_1	x_2	g	β_U	$\beta_{g'}$	$\beta_U{*}\beta_{g'}$	$\hat{h}''_1$	$\hat{h}''_2$
0	0	0	$\overline{1,3,5,6}$	$\overline{1,3,4}$	$\overline{1,3}$	1	0
0	0	1	$\overline{1,3,5,6}$	$\overline{2,5}$	$\overline{5}$	2	1, 2
0	0	2	$\overline{1,3,5,6}$	$\overline{6,7,8}$	$\overline{6}$	1	1
0	1	0	$\overline{4,6}$	$\overline{1,3,4}$	$\overline{4}$	0	2
0	1	2	$\overline{4,6}$	$\overline{6,7,8}$	$\overline{6}$	1	1
0	2	0	$\overline{4,5}$	$\overline{1,3,4}$	$\overline{4}$	0	2
0	2	1	$\overline{4,5}$	$\overline{2,5}$	$\overline{5}$	2	1, 2
1	0	0	$\overline{1,2,7}$	$\overline{1,3,4}$	$\overline{1}$	1	0
1	0	1	$\overline{1,2,7}$	$\overline{2,5}$	$\overline{2}$	0	1, 2
1	0	2	$\overline{1,2,7}$	$\overline{6,7,8}$	$\overline{7}$	0	0
1	2	2	$\overline{7}$	$\overline{6,7,8}$	$\overline{7}$	0	0
2	0	2	$\overline{8}$	$\overline{6,7,8}$	$\overline{8}$	1	2
2	1	0	$\overline{4}$	$\overline{1,3,4}$	$\overline{4}$	0	2
2	2	0	$\overline{4}$	$\overline{1,3,4}$	$\overline{4}$	0	2

6. Disjunctive Serial Separations

In the case of disjunctive separations we can prove the converse of Theorem 5.3.

Theorem 6.1. *For every disjunctive serial separation (g, h) of f there is a blanket β_g satisfying the conditions of Theorem 5.3.*

Proof. The proof follows closely the proof given in [3] for the case of Boolean functions, and the interested reader is referred to that paper for more details.

Suppose first that a disjunctive serial separation (g, h) exists. Then, for every singleton vector e^V, such that $e \in \Sigma^n$ is relevant to f, $\hat{g}(\dot{e}^V)$ is a singleton vector in Σ^p. Define blanket β_g on F as follows. Let $a \in \Sigma^p$ be any singleton vector, and let $\beta_g = ne\{F_{\hat{g}=a}\}$, where

$$F_{\hat{g}=a} = \{t \in F \mid \exists b \in \Sigma^s \text{ such that } t^V_\downarrow \supseteq b, \text{ and } \hat{g}(\dot{b}) = a\}.$$

One verifies that β_g satisfies the conditions of the theorem; for details see [3].

It is shown in [3] that the construction used in the proof of Theorem 6.1 does not hold for nondisjunctive separations.

Example 6.2. *Consider function f from Table 2. Suppose that we are given the separation consisting of functions g and h of Tables 9 and 10. We construct blanket β_g as follows. The value* 0 *is produced by g only if the inputs are* 22*. This can happen* *only if the input vector to f ends in 22, which occurs in rows 2 and 4 of Table 2. Thus $\{2,4\}$ is a block of β_g. Similarly, the blocks corresponding to the values 1 and 2 for g are $\{2,3,4\}$ and $\{1,3,4\}$, respectively. Hence*

$$\beta_g = \{\overline{2,4};\ \overline{2,3,4};\ \overline{1,3,4}\}.$$

One verifies that β_g satisfies the conditions of the Theorem.

Table 9: Function g.

x_2	x_3	g
0	0, 2	2
1	0, 2	1
2	0	1
2	2	0

Table 10: Function h.

x_1	g	h_1	h_2
0	0, 2	0, 1	0
1	0, 1	0	1
1	2	0, 1	1, 2
2	0	0, 1	1, 2
2	1, 2	1	2

Now we give another example of the construction used in Theorem 5.3 to illustrate the choices involved in finding function g.

Example 6.3. *Suppose we are given blanket $\beta_{g''} = \{\overline{1,2,4};\ \overline{2,3,4};\ \overline{1,3,4}\}$, for function f from Table 2. Note that this blanket satisfies the conditions of the theorem. We can code the blocks of $\beta_{g''}$ by a single 3-valued variable, by two binary variables, etc.; we choose the first option. The coding is shown below.*

$$\beta_{g''} = \{\overset{0}{\overline{1,2,4}};\ \overset{1}{\overline{2,3,4}};\ \overset{2}{\overline{1,3,4}}\}.$$

From Table 2 we find all the possible values relevant to g''; these are shown in the first two columns of Table 11. The third column lists the block of β_V corresponding to each input combination. The fourth column lists the blocks of $\beta_{g''}$ containing each block of β_V, and the fifth column lists the corresponding output values. The last column shows the values of $\hat{g}''$ that we have arbitrarily chosen from the fifth column.

Table 11: Finding function $\hat{g}''$.

x_2	x_3	β_V	$\beta_{g''}$	options	$\hat{g}''$
0	0	$\overline{1,3,4}$	$\overline{1,3,4}$	2	2
0	2	$\overline{1,4}$	$\overline{1,2,4},\ \overline{1,3,4}$	0, 2	0
1	0	$\overline{2,3}$	$\overline{2,3,4}$	1	1
1	2	$\overline{2}$	$\overline{1,2,4},\ \overline{2,3,4}$	0, 1	0
2	0	$\overline{2,3,4}$	$\overline{2,3,4}$	1	1
2	2	$\overline{2,4}$	$\overline{1,2,4},\ \overline{2,3,4}$	0, 1	0

From the proof of Theorem 5.3 and from the examples, it is clear that for each β_g there may be are many separations, since often there are choices to be made. But it may appear from the construction (Theorem 6.1) of β_g from a separation, that the blanket corresponding to a disjunctive decomposition is unique. This is not necessarily the case. In fact, the next theorem states that set systems could be used instead of blankets.

Theorem 6.4. *Let $f(X)$ be a set function specified by a set F of set vectors, and let (U, V) be a pair of subsets of X such that $U \cup V = X$. For every set system satisfying the conditions stated below, there is a serial separation. For every disjunctive serial separation of f there is a set system σ satisfying the conditions stated below.*

- $\sigma_V \leq \sigma_g$, *and*
- $\sigma_U \circ \sigma_g \leq \sigma_f$,

where β_U, β_V, and β_f are defined as above, and $\sigma_U = max(\beta_U)$, $\sigma_V = max(\beta_V)$, and $\sigma_f = max(\beta_f)$.

Proof. The verification of this theorem is routine.

7. Serial Decompositions

Suppose we have a serial separation (g, h) of a set function f with n inputs, where g has s inputs and p outputs, and h has $r+p$ inputs and m outputs. Such a separation is not quite a decomposition, because in a decomposition we require both g and h to have strictly fewer inputs than f. Thus we insist that $s < n$ and $r + p < n$. Since we are considering multiple-valued functions, there is always the possibility of widening the range of outputs of set function g so that it would have fewer outputs. In fact, we can make $p = 1$. However, in the following discussions we assume the range of output values of g is E.

Definition 7.1. *A serial decomposition of f with respect to (U, V) is a serial separation (g, h), in which $s < n$, and $r + p < n$.*

Note that if a serial separation (g, h) based on blanket β_g is a serial decomposition, then $s < n$ and the number q of blocks of β_g satisfies

$$r + \lceil \log_k q \rceil < n,$$

where $k = |E|$.

As in the Boolean case [3], we now develop a useful necessary condition for the existence of a decomposition. Let β be a blanket on a set, and $\beta' = \{B_i'\}$, a blanket on the same set. The *quotient* of a block B of β by β' is a set system on the set B defined by

$$B/\beta' = max\{B \cap B_i'\}.$$

The *block-cover cost* $c(B, \beta')$ is the minimum number of blocks of B/β' required to cover B, in the sense that the union of these blocks is B. The *cover cost*, $c(\beta, \beta')$, is now defined as the largest block-cover cost, that is,

$$c(\beta, \beta') = max\{c(B, \beta') | B \in \beta\},$$

where *max* here denotes the largest integer in a set.

Theorem 7.2. *Let $f(X)$ be a set function, and let U be a subset of X, where X has n elements, and U has $r < n$ elements. If for some V, a proper subset of X with $U \cup V = X$, and some blanket β_g, a serial decomposition of f based on β_g exists, then*

$$n - r > \lceil \log_k c(\beta_U, \beta_f) \rceil,$$

where k is the cardinality of the underlying set E.

Proof. Suppose a decomposition (g, h) based on $\beta_g = \{B_i\}$ exists. By our earlier note, if β_g has q blocks, then

$$n - r > \lceil \log_k q \rceil.$$

Let $c(\beta_U, \beta_f) = c$. Then our result follows if we can prove $q \geq c$. Take any block B of β_U. Since $\beta_U * \beta_g \leq \beta_f$, we have $B \cap B_i$ is contained in a block of β_f, for every block B_i of β_g. Let B_i' be a block of β_f such that $B \cap B_i \subseteq B_i'$. Since β_g is a blanket, $\{B_i\}$ covers B, which in turn implies that $\{B_i'\}$ covers B. Thus, by definition, $c(B, \beta_f) \leq q$, the number of blocks in β_g. Since this is true for all blocks of β_U, we get $c \leq q$.

Example 7.3. *Let us find all the sets U for which a decomposition of the set function f' of Table 4 is not ruled out by the condition of Theorem 7.2. For convenience, we shall use l_0 as a shorthand for $\lceil \log_k c(\beta_U, \beta_{f'}) \rceil$. Recall that*

$$\beta_{f'} = \{\overline{7};\ \overline{2};\ \overline{2,4};\ \overline{1,3};\ \overline{6};\ \overline{4,8};\ \overline{3};\ \overline{5}\}.$$

For one-variable sets, that is, for $U = \{x_i\}$, we have $n - r = 3$. The condition of the theorem is satisfied, because the number of blocks in β_f is 8; consequently, the cost $c(B, \beta_f)$ cannot be greater than 8 for any block B of β_U. Therefore, $l_0 \leq \lceil \log_3 8 \rceil = 2 < n - r$.

For $U = \{x_1, x_2\}$, we have:

$$\beta_U = \{\overset{B_1}{\overline{1,3,5,6}};\ \overset{B_2}{\overline{4,6}};\ \overset{B_3}{\overline{4,5}};\ \overset{B_4}{\overline{1,2,7}};\ \overset{B_5}{\overline{7}};\ \overset{B_6}{\overline{8}};\ \overset{B_7}{\overline{4}}\}.$$

For the first block, $B_1/\beta_{f'} = max\{\overline{1,3};\ \overline{6};\ \overline{3};\ \overline{5}\} = \{\overline{1,3};\ \overline{5};\ \overline{6}\}$. Similarly, $B_2/\beta_{f'} = \{\overline{4};\ \overline{6}\}$, $B_3/\beta_{f'} = \{\overline{4};\ \overline{5}\}$, $B_4/\beta_{f'} = \{\overline{1};\ \overline{2};\ \overline{7}\}$, $B_5/\beta_{f'} = \{\overline{7}\}$, $B_6/\beta_{f'} = \{\overline{8}\}$, and $B_7/\beta_{f'} = \{\overline{4}\}$. The corresponding costs are $c(B_1, \beta_{f'}) = 3$, $c(B_2, \beta_{f'}) = 2$, etc. Hence, $l_0 = 1 < n - r = 2$, and the condition is satisfied.

For $U = \{x_2, x_3\}$,

$$\beta_U = \{\overset{B_1}{\overline{1,3,6,7,8}};\ \overset{B_2}{\overline{2,3,5}};\ \overset{B_3}{\overline{2}};\ \overset{B_4}{\overline{6}};\ \overset{B_5}{\overline{4}};\ \overset{B_6}{\overline{7}};\ \overset{B_7}{\overline{5}}\}.$$

Here U fails to satisfy the condition, because $B_1/\beta_{f'} = \{\overline{1,3};\ \overline{6};\ \overline{7};\ \overline{8}\}$. Hence $l_0 = 2 \not< 2 = n - r$.

Altogether, the condition is satisfied for the following subsets of X:

$$\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_1, x_4\}, \text{ and } \{x_3, x_4\}.$$

After the verification of the necessary condition from Theorem 5.3, we find all these subsets lead to serial decompositions. In fact, except for $\{x_1, x_2\}$, the other seven subsets all lead to disjunctive decompositions, while $\{x_1, x_2\}$ leads to nondisjunctive decompositions only.

8. Concluding Remarks

We have concentrated here only on the problem of finding a decomposition of a function f given the sets U and V. Finding these sets is far from trivial [3]. Other ideas [3] used in Boolean function decomposition can be extended also to multiple-valued functions. However, this is outside of the scope of this paper.

Acknowledgments

The authors thank L. Jóźwiak of the Eidhoven University of Technology, T. Łuba of the Warsaw University of Technology, and the members of the Maveric Research Group at the University of Waterloo for useful discussions concerning this work.

References

1. R. L. Ashenhurst, The Decomposition of Switching Functions, *Proc. of International Symp. Theory of Switching Functions*, 1959.

2. R. K. Brayton, et al., *Logic Minimization Algorithms for VLSI Synthesis,* Kluwer Academic Publishers, Boston, MA, 1984.

3. J. A. Brzozowski and T. Łuba, Decomposition of Boolean Functions Specified by Cubes, Research Report CS–97–01, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, January 1997.

4. H. A. Curtis, *A New Approach to the Design of Switching Circuits,* D. Van Nostrand Co. Inc., Princeton, NJ, 1962.

5. S. Grygiel, M. Perkowski, M. Marek-Sadowska, T. Łuba, and L. Jóźwiak, Cube Diagram Bundles: A New Representation of Strongly Unspecified Multiple-Valued Functions and Relations, *Proc. IEEE International Symposium on Multiple-Valued Logic,* Antigonish, Nova Scotia, Canada, pp. 287–292, May 1997.

6. J. Hartmanis and R. E. Stearns, *Algebraic Structure Theory of Sequential Machines,* Prentice-Hall, Englewood Cliffs, NJ, 1966.

7. T. Łuba, Decomposition of Multiple-Valued Functions, *Proc. 25th International Symposium on Multiple-Valued Logic,* Bloomington, Indiana, pp. 256–261, May 1995.

8. T. Łuba and H. Selvaraj, A General Approach to Boolean Function Decomposition and its Application in FPGA-Based Synthesis, *VLSI Design*, Vol. 3, Nos. 3–4, pp. 289–300, 1995.

9. M. Perkowski et al., Decomposition of Multiple-Valued Relations, *Proc. IEEE International Symposium on Multiple-Valued Logic,* Antigonish, Nova Scotia, Canada, pp. 13–18, May 1997.

10. J. P. Roth and R. M. Karp, Minimization over Boolean Graphs, *IBM Journal of Research and Development,* Vol. 6, pp. 227–238, April 1962.