On Recognition of Symmetries for Switching Functions in Reed-Muller Forms *

J. T. Butler,† G. W. Dueck,‡ G. Holowinski,|| V. P. Shmerko, § S. N. Yanushkevich ¶

† Department of Electrical & Computer Eng., Naval Postgraduate School,
Code EC/BU Monterey, CA 93943-5121, USA, butler@cse.nps.navy.mil
‡ St. Francis Xavier University, Department of Math. & Computer Science,
   Antigonish, B2G 2W5, Nova Scotia, CANADA, gdueck@stfx.ca
|| Institute of Computer Science & Information Systems, Techn. University, Szczecin, POLAND
§, ¶ Institute of Computer Science & Information Systems, Techn. University, Szczecin, POLAND,
   shmerk@ii.uw.szczecin.pl /University of Informatics & Radioelectronics, Minsk, BELARUS

Abstract

This paper addresses the recognition of partial symmetry in variables of switching functions represented in positive polarity Reed-Muller (RM) form. We develop a formal representation of partial symmetries in this RM form and present algorithms for their detection. In addition, we show necessary and sufficient conditions to recognize in RM expression partial and total symmetries in variables of the function. Our program RECSym successfully recognizes symmetries in RM expansion in standard benchmark circuits.

Index Terms. Switching functions, symmetry, Reed-Muller expansion,
1 Introduction

The problem of recognition of symmetries in switching functions has been studied since the early history of switching theory. Interest in this important area has continued to the present. In technology independent circuit minimization and technology mapping, one of the actual current problems is to determine if two functions are equivalent under input permutation and input/output complementation (Boolean matching problem) \([19], [26], [27]\). In the case of totally symmetric functions, Boolean matching focuses on complementation only.

There has been significant interest in exploiting symmetry among variables to determine efficient variable orders in binary decision diagrams \([17], [18]\). Although the best orders tend to place symmetric variables together, in rare instances the best ordering requires symmetric variables to be dispersed. For such functions, fast recognition of groups of symmetric variables is very important for decision diagram design.

There are well known methods of circuits design, decomposition, verification and minimization based on symmetric properties \([9], [10], [16], [19], [21]\).

Generally speaking, algorithms for detection of the symmetry conditions in any given function is a prerequisite to many recently developed methods of modern Computer Aided Design (CAD) of integrated circuits. Our interest in the problem of recognizing symmetries in switching functions is motivated by these practical applications in CAD. This paper addresses the recognition of partial and total symmetry of variables in switching functions.

There are several main techniques to investigate symmetries based on different principles, namely,

(i) manipulation of a truth table,

(ii) transformation of the given function into the spectral domain, and

(iii) formal representation of symmetric functions developed in this paper.

The best known algorithms explore properties of symmetries via manipulation of the truth table of given function. For example, in \([6]\) an effective method to detect totally, partially and so called multiform symmetric switching functions based on numerical methods has been proposed. Also, information theory methods have been applied to detect symmetries; the basic idea is to convert truth table into decision tree and detect symmetries via information theoretic measures \([5], [28]\).

The second direction exploits features of spectra to determine the symmetries in variables of a given function. There are many results on detecting symmetries in Hadamard, Haar and other transform bases \([12], [13], [15], [20]\).

Recently, there is a growing interest in AND/EXOR based design styles in CAD (see e.g., \([10], [21]\)). Implementation of AND/EXOR circuits often results in a more economical realization of the circuit (in terms of gates and gate interconnections) and is often more easily tested. This is particularly true for applications like error control, arithmetic circuits, and encrypting schemes.
In our investigation we focus on the recognition of symmetries in RM expressions.

RM spectrum (coefficients) are used to recognize certain properties of the switching function. It should be pointed out that from the position of spectral technique, the RM expansion is a result of RM transform, i.e. particular case of spectral representation.

To the best of our knowledge, Davio and Bioul were the first to suggest a method to detect total symmetry in RM spectrum of switching functions [7].

The main advantage of this approach is that one doesn’t need a formal (algebraic) representation and the detection of symmetries is a process of manipulation spectral coefficients.

A feature of our classification is an algebraic representation of symmetric switching function that has not been received much attention. The crucial point is to obtain formal descriptions of different types of symmetries and to study unique features of given symmetric function in formal way. Contrary to well known classical methods which operate with Sum of Products (SOP) expressions [14], formal RM representation of symmetric switching functions is more difficult.

As an example of important and successful result of formal approach and an illustration of its power, we refer the reader to [2], [8], [14], [25]. For instance, the following statement is widely used in AND/EXOR representation of switching function: a function is totally symmetric if and only if in positive polarity RM form of the function, the coefficients of all products with the same number of literals are the same. However, several attempts to find an algorithm to recognize partially symmetric functions in AND/EXOR forms have so far failed. In our opinion, the main reason why an efficient algorithm has not been developed yet, is the absence of strong mathematical results on AND/EXOR forms for partially symmetric switching functions.

It should be pointed out that there are many related unsolved practical problems. For example, optimal characteristics of decision diagrams for partially symmetric function has not been studied yet. In has been shown in [3] that reduced order decision diagram require $O(n^2)$ nodes for totally symmetric functions; the optimal characteristics of FPRM expressions for different types of symmetries have not been investigated yet either.

The second motivation of our investigation resulted in an examination of algebraic representations of symmetric functions in the RM domain and their formal study. We try to overcome the difficulties in synthesis of formal equations for some types of symmetries widely used in CAD.

A review of previously obtained results shows that there are some approaches to describe symmetric functions in RM form. The mostly known approach by Davio et al. [8] is based on matrix calculation (spectral transform). Suprun in [24] uses a rectangular binary table to synthesize FPRM expression for totally symmetric function. We report more general approach applied to partially symmetric functions.

The aim of this paper is twofold: first, to obtain formal (algebraic) representations of partially symmetric functions in positive polarity RM notation; second, to show
advantages of formal study of symmetries and "translate" them into practical benefits for CAD. In this connection, we prove necessary and sufficient conditions for the recognition of symmetries in RM expansion and we develop recognition algorithms. Our intermediate result has an important independent significance, namely, method for calculation characteristics of partially symmetric functions. Suggested algorithm can be more preferable in some cases compared, for example, with [6] because of their simplicity. These are the main contributions of our paper.

Moreover, we solve some related problems. In particular, we show that due to the formal representation, we determine different technical properties which are useful for realization of algorithms. However, to simplify the problem, we have to limit our investigations to positive polarity RM expansion because of the complexity associated with other polarities.

This paper is organized as follows. In Section 2 we give terminology and briefly describe properties and summarize necessary definitions. Section 3 describes a method of representing partially symmetric functions and the detection of symmetries in positive polarity RM expansions. Also, the method is suitable for totally symmetric functions. In Section 4, we discuss experimental results from our program REC Sym for standard benchmarks.

2 Preliminaries

The goal of this section is to introduce formally the main properties of partially and totally symmetric functions. We give a formal representation of an arbitrary function in positive polarity RM form whose algebraic structure is most convenient for description of these symmetries.

2.1 Partially and totally symmetric functions

Let $f$ be a switching function on a set of variables $X = \{x_1, x_2, \ldots, x_n\}$. $f$ is partially symmetric with respect to $X_i \subseteq X$ if any permutation of variables in $X_i$ leaves $f$ unchanged.

Let $\rho = \{X_1, X_2, \ldots, X_s\}$ denotes a partition of $X$. Function $f$ is $\rho$-symmetric if $f(X_1, X_2, \ldots, X_s) = f(X'_1, X'_2, \ldots, X'_s)$, where $X'_i$ is an arbitrary permutation on $X_i$ [14]. A $\rho$-symmetric function $f$ for which $\rho$ is the partition consisting of one block $\rho = \{X\}$, is a totally symmetric function. That is, this function is unchanged by any permutation of its variables and depends only on the number of variables that are 1.

A switching function may be symmetric in a subset of $k$ variables, $2 \leq k \leq n$, in many different forms, for instance, in variables $\{x_i, x_j\}$ and also in $\{\overline{x_i}, \overline{x_j}\}$. A function exhibiting symmetry in a subset of $k$ variables in all $2^{n-1}$ possible forms as above is said to be multiform symmetric in those $k$ variables [6].

Example 2.1. (i) Function $f = \overline{x_1}x_3x_4 \vee x_1x_2x_3x_4 \vee x_1x_2x_3 \vee \overline{x_2}x_3x_4$ is $\rho$-symmetric in variables $\rho = \{x_1, x_2\}, \{x_3, x_4\}$.

(ii) $f = x_1x_2 \vee x_3$ is a partially symmetric function in variables $\{x_1, x_2\}$.

(iii) $f = x_1\overline{x_2}\overline{x_4} \vee x_1x_3x_4 \vee \overline{x_1}x_2x_5 \vee \overline{x_1}\overline{x_2}x_5$ is symmetric with respect to sets of
variables \( \{x_3, x_4\} \) and \( \{x_2, x_5\} \).

(iv) \( f = x_1 \oplus x_2 \), and \( x_1 x_2 \lor x_2 x_3 \lor x_1 x_3 \) are totally symmetric functions.

(v) \( f = x_1 \bar{x}_2 \lor \bar{x}_1 x_2 \) is multiform symmetric in \( \{x_1, x_2\}, \{\bar{x}_1, \bar{x}_2\} \) and \( \{x_1, \bar{x}_2\}, \{\bar{x}_1, x_2\} \).

A useful concept is the carrier vector (extended carrier vector) of Davio notation \cite{7}.

**Definition 2.1.** The carrier vector \( Y \) of a symmetric switching function \( f \) is the truth column vector of \( f \) with entries removed that are identical because of symmetry.

The carrier vector is a reduced ordering truth column vector of a symmetric switching function. It contains all of the information necessary to completely specify a symmetric function. For a totally symmetric function on \( n \) variables, the carrier vector has length \( n + 1 \). We can specify a partially symmetric function as a vector of values \( Y = [y^{(0)} y^{(1)} \ldots y^{(\theta-1)}] \), where \( \theta = (k + 1)2^{n-k} \) that is agree with \cite{14}.

In other words, the number of distinct assignments is the number of logic values that need to be specified to completely specify partially symmetric function \( f \). That is, this is a specification of values to variables outside the set of partially symmetric variables, together with a specification of how many of the partially symmetric variables are 1 completely specifies \( f \).

\footnote{It was proved in \cite{14} that there are \( 2^n \), \( \theta = (k + 1)2^{n-k} \), different partially symmetric functions of \( n \) variables with respect to \( k \) variables.}

The concept of distinct assignments of values to variables is recalled in the example below.

**Example 2.2.** (i) A totally symmetric function of 3 variables is represented by the column truth vector \( X = [abbcbcd] \) where \( a, b, c, d \in \{0, 1\} \), i.e. \( f(000) = a \), \( f(001) = b \), \( f(010) = b \), \( f(011) = c \), \( f(100) = b \), \( f(101) = c \), \( f(110) = c \), \( f(111) = d \). The distinct assignments among them are \( f(000) = a \), \( f(001) = b \), \( f(011) = c \), \( f(111) = d \), i.e. the elements 0,1,2,4 of vector \( X \). These assignments, enumerated by 0,1,2,3, form the carrier vector \( Y = [abcd] \). The distinct sets of assignments are \{000\}, \{001, 010, 100\}, \{011, 101, 110\}, \{111\}.

(ii) A partially symmetric function of 3 variables with respect to \( \{x_1, x_3\} \) is given by the truth column vector \( X = [abcdbdef] \). The distinct elements are 0,1,2,3,5,7, i.e. \( f(000) = a \), \( f(001) = b \), \( f(010) = c \), \( f(011) = d \), \( f(100) = b \), \( f(101) = c \), \( f(110) = d \), \( f(111) = f \). So, it can be represented by the carrier vector \( Y = [abcde] \) whose elements are 0,1,2,3,4,5. The distinct assignments are grouped to the following sets: \{000\}, \{001, 100\}, \{010\}, \{011, 110\}, \{101\}, \{111\}.

2.2 Taylor expansion of a switching function

In this paper, to represent the positive polarity RM form, we use the Taylor expansion, as an analogue originally proposed by Akers \cite{1} and later developed by Davio \cite{7}, Thayse \cite{25}, Bochmann and Posthoff \cite{2}.

The main characteristics of the Taylor expansion are: (i) the calculation of RM coefficients through Boolean differences \footnote{The calculation of RM coefficients through Boolean differences is described in detail in \cite{2}.}
and (ii) a mechanism of manipulation with assignments of values to variables. We explore in our study the last feature, assuming that RM coefficients can be calculated via any known methods, e.g. [8], [10], [11], [12], [21], [28].

Remark 2.1. An FPRM expansion can be viewed as the result of a succession of expansions, each time using either the positive Davio \( f = f_0 \oplus x f_2 \) or negative Davio \( f = \overline{x f_2} \oplus f_1 \) decomposition, where, for each variable of function \( f \), \( f_0(f_1) \) is \( f \) with variable \( x \) replaced by 0(1), and \( f_2 = f_0 \oplus f_1 \).

In the FPRM expression each variable appears always complemented or always uncomplemented. For more details, the reader is directed to [21].

Example 2.3. The result of positive Davio decomposition for function \( f = \overline{x_1 x_2} \) is \( f = (x_1 \oplus 1)(x_2 \oplus 1) = 1 \oplus x_1 \oplus x_2 \oplus x_1 x_2 \).

We use positive polarity RM expansion (0-polarity) of a given switching function in this paper

\[
f = \sum_{j=0}^{2^n-1} r^{(j)} x_1^{j_1} x_2^{j_2} \ldots x_n^{j_n}, \tag{2.1}
\]

\[
x_t^{j_t} = \begin{cases} 
1, & j_t = 0 \\
x_t, & j_t = 1 
\end{cases} \tag{2.2}
\]

Expression (2.1) can be interpreted as the result of applying the positive Davio decomposition to the given function. Note, that there is one positive polarity

RM expansion for an arbitrary switching function, i.e. it is canonical form that contains positive literals only and requires up to \( 2^n \) product terms.

Remark 2.2. Traditionally, the positive polarity RM expansion is written as follows:

\[
a^{(0)} \oplus a^{[1]} x_1 \oplus \ldots \oplus a^{[n]} x_n \oplus a^{[12]} x_1 x_2 \oplus a^{[13]} x_1 x_3 \oplus \ldots \oplus a^{[n-n-1]} x_n x_{n-1} \oplus \ldots \oplus a^{[12..n]} x_1 x_2 \ldots x_n. \tag{2.2}
\]

In contrast, the logic Taylor expansion (2.1) allows easier manipulate with indexes of RM coefficients and product terms.

This fact is important in our study. We illustrate the difference in a formal style (algebraic structure) and a calculation associated with a traditional Taylor form of positive polarity RM expression (2.1) by a simple example.

Example 2.4. (i) Consider a traditional positive polarity RM representation of an arbitrary function \( f \) of two variables, \( f = a^{(0)} \oplus a^{[1]} x_1 \oplus a^{[2]} x_2 \oplus a^{[12]} x_1 x_2 \).

(ii) Now represent function \( f \) as a Taylor expansion (2.1), in which case, we obtain

\[
f = r^{(0)} x_1^{0} x_2^{0} \oplus r^{(1)} x_1^{0} x_2^{1} \oplus r^{(2)} x_1^{1} x_2^{0} \oplus r^{(3)} x_1^{1} x_2^{1}.
\]

Finally, taking in account (2.2), we can write \( f = r^{(0)} \oplus r^{(1)} x_2 \oplus r^{(2)} x_1 \oplus r^{(3)} x_1 x_2 \).

Note, the order of RM coefficients is different. Hence, (2.1) gives a mechanism for manipulation with indexes of RM coefficients, related products and literals.

In next Section, we will show a formal representation of the synthesis of the positive polarity RM form of symmetric functions.
3 Recognition of partial symmetries in positive polarity RM expression

In this Section, we focus on the following recognition problem: for a partially symmetric switching function in a positive polarity RM form, find

(i) a formal representation
(ii) necessary and sufficient conditions for symmetry in variables, and
(iii) an efficient strategy to determine for which variables it is partially symmetric.

The idea of our approach is as follows. We consider $2^n$ assignments of values to variables that should be structured with the goal to obtain the distinct sets of assignments. It is the first stage of our study. In order to form the carrier vector of the function we propose an ordering operator. This allows a formal representation of a positive polarity RM expression for partially symmetric functions.

3.1 Basic properties of partially symmetric functions

The main question of any symmetry recognition algorithm is to describe the assignment of values to variables for which the function is symmetric or non-symmetric. There are many methods to solve this problem. For example, one of the widely used algorithm for detection symmetries is based on numerical methods [6].

Below we introduce the method to define main characteristics of the partially symmetric function with respect to $k$ given variables.

The characteristics of partially symmetric functions include:

- Distinct sets of the assignments, and
- Carrier vector.

3.1.1 Distinct sets of assignments

Let $f$ be a switching function of $n$ variables that is partially symmetric with respect to $k < n$ variables $\{x_{t_1}, ..., x_{t_k}\}$ where $j_1, ..., j_k \in \{1, 2, ..., n\}$. Consider assignments of values $j_1, j_1, ..., j_k, j_n$ to variables $x_1, x_1, ..., x_k, x_n$, where $j_1, ..., j_k \in \{1, 2, ..., n\}$.

The length of carrier vector for a function which is partially symmetric in $k$ variables and has no symmetries with respect to remaining $n-k$ variables, is $(k+1)2^{n-k}$ accordingly. It is not surprise that the number of distinct positive polarity RM coefficients for such function takes the same values, namely, $(k+1)2^{n-k}$.

We are interested in describing distinct sets of assignments taking into account the partial symmetry.

We start with formal definition of distinct sets of assignments that allows us to describe $(k+1)2^{n-k}$ these sets within all of the $2^n$ possible assignments of values to variables.

**Definition 3.1.** Let $f$ be a function partially symmetric in $k$ variables. The distinct set is a set of the assignments

$$j_1, j_1, ..., j_k, j_n$$

(3.1)

of values to $n$ variables

$$x_1, x_1, ..., x_k, x_n$$
such that to satisfy the linear equation

\[ j_{t_1} + \ldots + j_{t_k} = J, \quad (3.2) \]

for \( J \in 0, 1, \ldots, k. \)

Some comments for this formal description are useful.

The number of distinct sets of assignments is the number of logic values that need to be specified to completely specify partially symmetric function \( f \). That is, this is a specification of values to \( n - k \) variables outside the set of partially symmetric variables, together with a specification of how many of the partially symmetric variables are 1.

Within a distinct set, corresponding to a \( J \), there are \( C_k^J \) assignments of \( J \) 1’s to \( k \) partially symmetric variables, while the remaining \( n - k \) variables are fixed. Also, these fixed values are assigned in \( 2^{n-k} \) ways. Since there are \( k + 1 \) different \( J \) \((J = 0, 1, \ldots, k)\), there are \((k + 1)2^{n-k}\)-th distinct set of assignments.

It follows from the above that the number of distinct sets of assignments is equal to the number of distinct assignments.

Naturally, let \( J = 0 \) in linear equation (3.2), i.e. \( j_t_1 + \ldots + j_{t_k} = 0 \), that produce one solution \( 0 \ldots 0 \). There are \( 2^{n-k} \) assignments \( j_{t_1}0\ldots0\ldots j_{t_k}0\ldots0 \) of values to variables outside the set of partially symmetric variables. So, the case \( J = 0 \) raises \( 2^{n-k} \) distinct 1-element sets.

In case \( J = 1 \), i.e. \( j_{t_1} + \ldots + j_{t_k} = 1 \), we have \( C_k^1 \) solutions, namely, \( j_{t_1}0\ldots01\ldots j_{t_k}0\ldots0 \ldots j_{t_1}0\ldots0 \ldots j_{t_k} \). Each of these \( k \) solutions raises \( 2^{n-k} \) specifications of values to remaining \( n - k \) variables. So, there are \( 2^{n-k} \) distinct \( k \)-elements sets. By analogy, for \( J = 2 \) equation \( t_1 + \ldots + j_{t_k} = 2 \) has \( C_k^2 = (k(k-1))/2 \) different solutions. There are next \( 2^{n-k} \) distinct set each includes \((k(k-1))/2 \) elements.

Finally, the equation \( j_{t_1} + \ldots + j_{t_k} = k \) has one solution \( j_{t_1} = \ldots = j_{t_k} = 1 \). It implies next \( 2^{n-k} \) distinct sets counted as \( 2(n-k) \) specifications of values to \( n - k \) variables outside the set of partially symmetric variables.

So, the number of distinct sets is

\[ 2^{n-k} + \ldots + 2^{n-k} = (k + 1)2^{n-k}. \]

We demonstrate the technique of calculation based on Definition 3.1 with example as follows.

Example 3.1. Let \( f \) be a 4-variable function partially symmetric with respect to variables \( \{x_1, x_3, x_4\} \), i.e. \( k = 3 \). Let us divide the set of assignments \( j_1, j_2, j_3, j_4 \) into \( 2^{n-k}(k+1) \) distinct subsets.

We start with \( J = 0 \). The assignments to satisfy the equation \( j_1 + j_3 + j_4 = 0 \) be 0000 that corresponds to \( 2^{n-k} = 2 \) distinct sets of assignments each consists of one assignment: \{0000\} and \{0100\}. These assignments raise the product terms \( x_{1}^{0}x_{2}^{0}x_{3}^{1}x_{4}^{0} = 1 \) \((j_2 = 0)\) (remain, that \( x_{j}^{j} = 1 \) for \( j = 0 \) and \( x_{i}^{j} = x_{i} \) for \( j = 1 \); here ”1” means that there is no product term corresponded this assignment) and \( x_{1}^{0}x_{2}^{1}x_{3}^{0}x_{4}^{0} = x_{2} \) \((j_2 = 1)\).

Further, for \( J = 1 \) the equation \( j_1 + j_3 + j_4 = 1 \) gives \( C_3^1 = 3 \) solutions. It raises two distinct 3-components sets: when \( j_2 = 0 \), we obtain the set \{0001, 0010, 1000\} that correspond to the product terms
\[x_0^1 x_0^2 x_3 x_4^1 = x_4, \quad x_1^0 x_2^0 x_3^1 x_4^1 = x_3 \quad \text{and} \quad x_1^0 x_2^0 x_3^0 x_4^0 = x_1; \] when \( j_2 = 1 \), the set be \{0101,0110,1100\} that corresponds to the product sets \( x_0^1 x_2 x_3 x_4^1 = x_2 x_4, \quad x_0^0 x_2 x_3 x_4^0 = x_2 x_3 \).

For \( J = 2 \) the equation gives three solutions too. Then, we obtain two distinct sets: when \( j_2 = 0 \), these are \{0011,1001,1010\}; when \( j_2 = 1 \), these are \{0111,1101,1110\}. Finally, when \( J = 3 \), we obtain two distinct 1-component sets \{1011\} and \{1111\} and two product sets: \( x_1^1 x_2^1 x_3^1 x_4^1 = x_1 x_3 x_4 \) and \( x_1^1 x_2^1 x_3^1 x_4^1 = x_1 x_2 x_3 x_4 \).

### 3.1.2 Carrier vector

Definition 3.1 is true for arbitrary order of distinct sets and assignments of values to variables. On the other hand, the order of the elements of the carrier vector \( Y = [y^{(0)} y^{(1)} \ldots y^{(\theta-1)}] \) is fixed (see Example 2.2). Hence, to form a carrier vector, we have to reorder the distinct sets of assignments. The idea is to build the ordered string of distinct assignments \([Set_0, Set_1, \ldots, Set_{\theta-1}]\), and then the carrier vector \([f_{Min\{Set_0\}} f_{Min\{Set_1\}} \ldots f_{Min\{Set_{\theta-1}\}}]\) under condition

\[Min\{Set_{t-1}\} < Min\{Set_t\}, \quad (3.3)\]

t = 1, \ldots, \theta - 1.

We clarify this fact via examples below.

**Example 3.2.** (Continuation) Let us build carrier vector \( Y = [y^{(0)} y^{(1)} \ldots y^{(\theta-1)}] \).

The number of distinct sets of assignments is equal to the length of carrier vector \( \theta = (k + 1)2^{n-k} = 8 \).

Denote a distinct non-ordered set \( S_t' \) of assignments and corresponding product terms as

\[S_t' = [j_1 j_2 j_3 j_4 x_1^{j_1} x_2^{j_2} x_3^{j_3} x_4^{j_4}], t = p,\]

where \( p = 0, 1, \ldots, k, \quad i = 0, 1, \ldots, \theta - 1 \) and form these sets.

For \( J = 0 \) the solution of the equation \( j_1 + j_3 + j_4 = 0 \) is 0000. Consider two cases.

For \( j_2 = 0 \), we obtain

\[S_0' = [j_1 0 j_3 j_4 x_1^{j_1} 0 x_3^{j_3} x_4^{j_4}], J = 0 = [0000]\]

and the value of the function be \( f_{0000} \), denote it by \( a \in \{0, 1\} \).

For \( j_2 = 1 \), we have

\[S_1' = [j_1 1 j_3 j_4 x_1^{j_1} 1 x_3^{j_3} x_4^{j_4}], J = 0 = [0100] x_2, J = 0 \]

and \( f_{0100} = d \) (0 or 1) for these assignments.

Calculations in the following steps produce the string of the non-ordered distinct sets of the assignments \([S_0' S_1' S_2' S_3' S_4' S_5' S_6' S_7']\), where each corresponds to a distinct value of the function:

\[S_2' = [j_1 0 j_3 j_4 x_1^{j_1} 0 x_3^{j_3} x_4^{j_4}], J = 1 =\]

\[
\begin{bmatrix}
00001 \\
00101 \\
10000
\end{bmatrix} x_1 \quad \text{and} \quad \begin{cases}
f_{0001} = b \\
f_{0010} = a \\
f_{1000} = c
\end{cases}
\]

\[S_3' = [j_1 1 j_3 j_4 x_1^{j_1} 1 x_3^{j_3} x_4^{j_4}], J = 1 =\]

\[
\begin{bmatrix}
01011 \\
01100 \\
11000
\end{bmatrix} x_1 x_2 \quad \text{and} \quad \begin{cases}
f_{0101} = b \\
f_{0110} = e \\
f_{1100} = c
\end{cases}
\]

\[S_4' = [j_1 0 j_3 j_4 x_1^{j_1} 1 x_3^{j_3} x_4^{j_4}], J = 2 =\]

\[
\begin{bmatrix}
00111 \\
10100 \\
10011
\end{bmatrix} x_1 x_2 \quad \text{and} \quad \begin{cases}
f_{0011} = b \\
f_{1010} = c \\
f_{1001} = c
\end{cases}
\]
$S_6 = [j_1j_3j_4[x_1^{j_1}x_2^{j_2}x_3^{j_3}x_4^{j_4}]]_{j=2} = \\
\begin{bmatrix}
1011 & x_2x_3x_4 \\
1110 & x_1x_2x_3 \\
1101 & x_1x_2x_4
\end{bmatrix}_{j=2} \quad \text{and} \quad f_{1011} = f \\
f_{1110} = g \\
f_{1111} = h.

S_6 = [j_10j_3j_4[x_1^{j_1}0x_3^{j_3}x_4^{j_4}]]_{j=3} = \\
\begin{bmatrix}
1011 & x_1x_3x_4
\end{bmatrix}_{j=3} \quad \text{and} \quad f_{1011} = g

S_7 = [j_1j_3j_4[x_1^{j_1}1x_3^{j_3}x_4^{j_4}]]_{j=3} = \\
\begin{bmatrix}
1111 & x_1x_2x_3x_4
\end{bmatrix}_{j=3} \quad \text{and} \quad f_{1111} = h.

Now, form the carrier vector from this string. The first element of the string is the first assignment $Min\{S_0\}$ that corresponds to the first element $a \in \{0,1\}$ of carrier vector $Y$. Further, $Min\{S_6\}=0001$ corresponds to the second element $b$ of $Y$. By analogy, $Min\{S_4\} = Min\{0011, 1010, 1001\} = 0011$ and $f_{0011} = c$. Reordering of the numbers under condition (3.3) gives us the ordered string of distinct sets $[S_0^*S_2^*S_3^*S_4^*S_5^*S_6^*S_7^*] = [Set_0Set_1Set_2Set_3Set_4Set_5Set_6Set_7]$ and the carrier vector $Y = [f_{Min\{Set_0\}} f_{Min\{Set_1\}} f_{Min\{Set_2\}} f_{Min\{Set_3\}} f_{Min\{Set_4\}} f_{Min\{Set_5\}} f_{Min\{Set_6\}} f_{Min\{Set_7\}}] = [abcdefgh] \blacksquare

Now apply this technique to totally symmetric functions.

**Example 3.3.** Consider the construction of the string of distinct assignments and carrier vector $Y$ for a 4 variable totally symmetric function.

The number of distinct sets is equal the length of carrier vector $n+1 = 5$. The distinct sets are calculated as follows. For $J = 0$ the solution of the equation $j_1 + j_2 + j_3 + j_4 = 0$ is 0000, i.e. $Set_0 = [00001]^J_{j=0}$ and $f_{0000} = a$. Calculations for the next steps are done analogously:

$Set_1 = \begin{bmatrix}
0001 & x_4 \\
0101 & x_3 \\
1000 & x_1
\end{bmatrix}_{J=1}

Set_2 = \begin{bmatrix}
0011 & x_3x_4 \\
0110 & x_2x_4 \\
1010 & x_1x_3 \\
1100 & x_1x_2
\end{bmatrix}_{J=2}

Set_3 = \begin{bmatrix}
0111 & x_2x_3x_4 \\
1011 & x_1x_3x_4 \\
1101 & x_1x_2x_4 \\
1110 & x_1x_2x_3
\end{bmatrix}_{J=3}

Set_4 = \begin{bmatrix}
1111 & x_1x_2x_3x_4
\end{bmatrix}_{J=4}

Hence, the string of the distinct sets can be written as $[Set_0Set_1Set_2Set_3Set_4]$, and no ordering is needed. Finally, we obtain the carrier vector $Y = [Min\{Set_0\} Min\{Set_1\} Min\{Set_2\} Min\{Set_3\} Min\{Set_4\}] = [abcde] \blacksquare

3.1.3 Algorithm to define the distinct sets of assignments

The algorithm, *Index Generator* to determine the distinct sets of assignments is one of the most important modules of our recognition program RECSym. The input data be the truth column vector of the given switching function $f$ and the output data be distinct sets of assignments each corresponds to a coefficient of the positive polarity RM expression of $f$.

**Definition 3.2.** Let $S = [s_0s_1...s_n]$ be a *symmetry vector* of a function $f$, where
\( s_i = s_j = 1 \) iff \( f \) is unchanged by an interchange of \( x_i \) and \( x_j \).

**Example 3.4.** The symmetry vector \( S = [1111] \) specifies a totally symmetric function. \( S = [1011] \) specifies a function that is partially symmetric in \( \{x_1, x_3, x_4\} \).

**Definition 3.3.** Let 
\[
PS = \{x_{j_1}, x_{j_2}, \ldots, x_{j_k}\}
\]
be the set of partially symmetric variables in variable set \( X = \{x_1, x_2, \ldots, x_n\} \). For any assignment \( A \) of values to \( X \), define \( N_{PS(A)}^0 \) and \( N_{PS(A)}^1 \) as the number of 0’s and 1’s assigned to variables in \( PS \). Also define \( N_{PS(A)}^- \) and \( N_{PS(A)}^+ \) as the number of 0’s and 1’s assigned to remaining variables in \( A \).

Note that the defined notations satisfy the equations
\[
N_{PS(A)}^0 + N_{PS(A)}^1 = k,
\]
\[
N_{PS(A)}^- + N_{PS(A)}^+ = n - k.
\]

**Example 3.5.** (Continuation) Given \( S = [1011] \) and the current assignment \( j_1j_2j_3j_4 \), we obtain the following values: \( N_{PS(A)}^0 = 1, N_{PS(A)}^- = 0, N_{PS(A)}^1 = 2 \) and \( N_{PS(A)}^+ = 1 \).

The assignments \( j_1 \ldots j_n \) of a distinct set are equivalent, in the numbers \( N_{PS(A)}^0, N_{PS(A)}^-, N_{PS(A)}^1 \) and \( N_{PS(A)}^+ \).

**Example 3.6.** (Continuation) Given \( S = [1011] \), the assignment 0001 is equivalent to 0010 and 1000; for all three assignments, \( N_{PS(A)}^0 = 1, N_{PS(A)}^- = 0, N_{PS(A)}^1 = 2 \) and \( N_{PS(A)}^+ = 1 \).
The sketch of an algorithm to derive the distinct assignments and sets include the following steps:

- for assignment \( j_1 j_2 ... j_n \) of \( j \)-th element of a given truth table vector, \( j \in \{0, 2^n - 1\} \) compute the values \( N0_{PS(A)}, N1_{PS(A)}, N0_{PS(A)}, N1_{PS(A)} \)
- if values \( N0_{PS(A)}, N1_{PS(A)}, N0_{PS(A)}, N1_{PS(A)} \) for this assignment are equal to values \( N0_{PS(A)}, N1_{PS(A)}, N0_{PS(A)}, N1_{PS(A)} \) of one of components from the previously obtained distinct sets, and also the bits \( j_{t_1} ..., j_{t_k} \) are covered by 1’s from the symmetry vector \( S \), then this assignment is included in the distinct group.
- else the assignment forms a new distinct set.

A pseudo-code of the algorithm is represented in Fig. 1. Note, that the status matrix stores values of \( N0_{PS(A)}, N1_{PS(A)}, N0_{PS(A)}, N1_{PS(A)} \) and the minimal assignment \( \text{Min}\{\text{Set}_i\} \) of each distinct set. The array \( \text{Set}_i \) stores the whole \( i \)-th distinct set.

**Example 3.7.** (Comments to the algorithm) Input: truth column vector of a 4-variable function that is partially symmetric with respect to \( \{x_1, x_3, x_4\} \), and the symmetry vector \( S = [1011] \). Output: distinct sets of assignments corresponding to values of the coefficients of the positive polarity RM expression.

Let \( i = 0 \). For \( j = 0 \) (the current indexes \( j_1 j_2 j_3 j_4 = 0000 \)), we compute values \( N0_{PS(A)} = 1, N1_{PS(A)} = 0, N0_{PS(A)} = 3, \) and \( N1_{PS(A)} = 0 \). The status matrix is \( A_0 = [N0_{PS(A)}, N1_{PS(A)}, N0_{PS(A)}, N1_{PS(A)}], index = [1030, 0000], set_0 = \{0000\} \).

Next, \( i = i + 1 \). Now \( i = 1 \) and let \( j = 1, j_1 j_2 j_3 j_4 = 0010 \). We get values \( N0_{PS(A)} = 1, N1_{PS(A)} = 0, N0_{PS(A)} = 2, N1_{PS(A)} = 1 \). Here, there does not exists \( l \in \{0, 1\} \) such that \( [N0_{PS(A)}, N1_{PS(A)}, N0_{PS(A)}, N1_{PS(A)}, index \lor_S S_l'] = [N0_{PS(A)}, N1_{PS(A)}, N0_{PS(A)}, N1_{PS(A)}], j \lor_S S_l' \), because the values \( N0_{PS(A)}, N1_{PS(A)} \) of the current status matrix are not equal to the values in \( A_0 \). So, we extend the status matrix as follows \( [1030, 0000; 1021, 0001] \), \( set_1 = \{0001\} \).

Assign \( i = i + 1 \), i.e. \( i = 2 \) now and let \( j = 2, j_1 j_2 j_3 j_4 = 0010 \). We obtain values \( N0_{PS(A)} = 1, N1_{PS(A)} = 0, N0_{PS(A)} = 2, N1_{PS(A)} = 1 \). Here there exists \( l \in \{0, 1\} \) such that \( [N0_{PS(A)}, N1_{PS(A)}, N0_{PS(A)}, N1_{PS(A)}, index \lor_S S_l'] = [N0_{PS(A)}, N1_{PS(A)}, N0_{PS(A)}, N1_{PS(A)}], j \lor_S S_l' = [1030, 0000 \lor 1021, 0001 \lor 1011] \). So, we do not extend the status matrix, \( set_1 = set_2 \) add \( j_{bin} = \{0001, 0010\} \).

Following the algorithm in this way, we obtain the status matrix: \( [1030, 0000; 1021, 0001; 1012, 0011; 0130, 0100; 0121, 0101; 0112, 0111; 1003, 1011; 0103, 1111] \). The rows from 0 to 7 of the status matrix corresponds to the distinct sets, each characterized by \( N0_{PS(A)}, N1_{PS(A)}, N0_{PS(A)}, N1_{PS(A)} \) and the first component \( \text{Min}\{\text{Set}_i\} \). These sets are represented by arrays \( \text{Set}_i: \text{Set}_0 = \{0000\}, \text{Set}_1 = \{0001, 0010, 1000\}, \text{Set}_2 = \{0011, 1001, 1010\}, \text{Set}_3 = \{0100\}, \text{Set}_4 = \{0101, 0110, 1100\}, \text{Set}_5 = \{0111, 1101, 1110\}, \text{Set}_6 = \{1011\}, \text{Set}_7 = \{1111\} \).
3.1.4 Ordering operator

Following the considered algorithm, let us formulate the ordering process in formal way.

Given: the total number \((k + 1)2^{n-k}\) of distinct sets of assignments of a partially symmetric function.

Find: (i) the order of distinct sets and (ii) the components of each distinct set.

**Definition 3.4. Ordering operator**

\[ \mathcal{R}_{i}\{j_1...j_t,...,j_k...j_n\} \]

is the procedure for forming the \(i\)-th set \(\mathcal{R}_{i}\) of distinct assignments corresponding to the \(i\)-th element \(y^{(i)}\) of the carrier vector \(Y = [y^{(0)}y^{(1)}...y^{(\theta-1)}]\), \(i = 0, 1, \ldots \theta - 1\).

This operator produces, in order, a string of the distinct sets. It is clear that the length of the ordering distinct string is \((k + 1)2^{n-k}\).

3.1.5 Formal representation of partially symmetric function

Below we describe a procedure to determine the formal representation of positive polarity RM expansion for partially symmetric functions. This is based on the fact that a function of \(n\) variables with symmetry in \(k\) variables can be represented with \((k + 1)2^{n-k}\) distinct RM coefficients.

Suppose that the ordering operator \(\mathcal{R}_{i}\{j_1...j_t,...,j_k...j_n\}\) is defined for a partially symmetric function.

**Theorem 3.1.** Let \(f\) be a switching function of \(n\) variables that is partially symmetric with respect to \(k < n\) variables \(\{x_{t_1},...,x_{t_k}\}\) and let \(j_1...j_t,...,j_k...j_n\) be a distinct set of the assignments of values to \(n\) variables \(x_1...x_{t_1}...x_{t_k}...x_n\) to satisfy the linear equation (3.2). This distinct set corresponds to the distinct group of product terms \(x_{t}^{j_1}...x_{t_k}^{j_k}...x_{n}^{j_n}\), each is assigned to the same coefficient in the expression (3.4).

Then, the exclusive OR of all of the product terms described above is a positive polarity RM expansion of \(f\)

\[ f = \sum_{i=0}^{\theta-1} r^{(i)} \sum_{\mathcal{R}_{i}} x_{t_1}^{j_1}...x_{t_k}^{j_k}...x_{n}^{j_n}, \quad (3.4) \]

where \(\theta = (k + 1)2^{n-k}\); \(r^{(i)} \in \{0, 1\}\) is \(i\)-th coefficient; \(\mathcal{R}_{i}\) is the ordering operator for assignments \(j_1...j_t,...,j_k...j_n\) and \(x_{t}^{j_1}\) is defined from (2.2).

**Proof.** The proof follows immediately from Definition 3.4 of the ordering operator \(\mathcal{R}_{i}\); the second sum is modulo 2 sum of product terms within \(i\)-th distinct group, and the first sum is related to the distinct sets.

We explain the technique of calculation based on expression (3.4) by the following example.

**Example 3.8. (Continuation)** Applying Theorem 3.1, write the positive polarity RM form for partially symmetric function from Example 3.7.

Following equation (3.4) given \(k = 3\), \(n = 4\) and \(\theta = (k + 1)2^{n-k} = (3 + 1)2^{4-3} = 8\), we obtain

\[ f = \sum_{i=0}^{7} r^{(i)} \sum_{\mathcal{R}_{i}} x_{1}^{j_1}x_{2}^{j_2}x_{3}^{j_3}x_{4}^{j_4} \]

When \(i = 0\) and \(j_1 + j_2 = 0\), the result of the ordering operator is \(\mathcal{R}_{0}\{j_10j_3j_4\}\).
\{0000\}_{j=0}$. It corresponds to product term \(r^{(0)}x_0x_2x_3x_4 = r^{(0)}1 = r^{(0)}\).

By analogy
\[
i = 1 : \ \mathcal{R}_1\{j_1j_3j_1\}_{j=1} = \{0001, 0010, 1000\} \Rightarrow r^{(1)}(x_1 \oplus x_3 \oplus x_1);
\]
\[
i = 2 : \ \mathcal{R}_2\{j_1j_3j_1\}_{j=2} = \{0011, 1010, 1001\} \Rightarrow r^{(2)}(x_3x_4 + x_2x_3 \oplus x_1x_4);
\]
\[
i = 3 : \ \mathcal{R}_3\{j_1j_3j_1\}_{j=0} = \{0101\} \Rightarrow r^{(3)}x_2;
\]
\[
i = 4 : \ \mathcal{R}_4\{j_1j_3j_1\}_{j=1} = \{0101, 0110, 1100\} \Rightarrow r^{(4)}(x_2x_4 \oplus x_2x_3 \oplus x_1x_2).
\]

Other possible cases yield the results represented in Table 1.

<table>
<thead>
<tr>
<th>(n)</th>
<th>RM expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>(x_1x_2 \quad r^{(0)} \oplus r^{(1)}x_3 \oplus r^{(2)}(x_1 \oplus x_2) \oplus r^{(3)}x_2x_3 \oplus x_1x_3 \oplus r^{(4)}x_2x_3 \oplus r^{(5)}x_1x_2x_3)</td>
</tr>
<tr>
<td>3</td>
<td>(x_1x_3 \quad r^{(0)} \oplus r^{(1)}(x_3 \oplus x_1) \oplus r^{(2)}x_3 \oplus r^{(3)}x_2x_3 \oplus x_1x_2 \oplus r^{(4)}x_2x_3 \oplus r^{(5)}x_1x_2x_3)</td>
</tr>
<tr>
<td>3</td>
<td>(x_2x_3 \quad r^{(0)} \oplus r^{(1)}x_2x_3 \oplus r^{(2)}x_2 \oplus r^{(3)}x_2x_3 \oplus x_1x_2 \oplus r^{(4)}x_2x_3 \oplus r^{(5)}x_1x_2x_3)</td>
</tr>
<tr>
<td>4</td>
<td>(x_1x_2x_3 \quad r^{(0)} \oplus r^{(1)}x_1 \oplus r^{(2)}x_1 \oplus r^{(3)}x_2 \oplus x_1 \oplus r^{(4)}x_1 \oplus r^{(5)}x_2x_3 \oplus r^{(6)}x_1x_2 \oplus r^{(7)}x_1x_3 \oplus r^{(8)}x_1x_2x_3 \oplus r^{(9)}x_1x_2x_3 \oplus r^{(10)}x_1x_2x_3 \oplus r^{(11)}x_1x_2x_3x_4)</td>
</tr>
<tr>
<td>4</td>
<td>(x_1x_2x_3 \quad r^{(0)} \oplus r^{(1)}x_1 \oplus r^{(2)}x_1 \oplus r^{(3)}x_2 \oplus x_1 \oplus r^{(4)}x_1 \oplus r^{(5)}x_2x_3 \oplus r^{(6)}x_1x_2 \oplus r^{(7)}x_1x_3 \oplus r^{(8)}x_1x_2x_3 \oplus r^{(9)}x_1x_2x_3 \oplus r^{(10)}x_1x_2x_3 \oplus r^{(11)}x_1x_2x_3x_4)</td>
</tr>
<tr>
<td>4</td>
<td>(x_1x_3x_4 \quad r^{(0)} \oplus r^{(1)}x_1 \oplus r^{(2)}(x_1 \oplus x_2) \oplus x_1x_2 \oplus r^{(3)}x_2 \oplus x_1x_2x_4 \oplus r^{(4)}x_2x_3 \oplus r^{(5)}x_1x_2 \oplus r^{(6)}x_1x_2 \oplus r^{(7)}x_1x_2x_3 \oplus x_1x_2x_3 \oplus x_1x_2x_3 \oplus r^{(8)}x_1x_2x_3 \oplus r^{(9)}x_1x_2x_3 \oplus r^{(10)}x_1x_2x_3 \oplus r^{(11)}x_1x_2x_3x_4)</td>
</tr>
</tbody>
</table>

Table 1: Positive polarity RM expansion of switching functions with 3 and 4 variables that are partially symmetric with respect to 2 and 3 variables

![Table 1](image1)

3.1.6 Formal representation of totally symmetric functions

In this section, we consider a particular case of the Theorem 3.1 for totally symmetric functions. The well known fact that there are \(n+1\) distinct coefficients in positive polarity RM expression of totally symmetric function, follows immediately from this theorem given \(k = n\). We formulate this result in the form as follows.

The ordering operator \(\mathcal{R}_k\{j_1j_2\ldots j_n\}\) is of special form for totally symmetric functions as shown below.

**Corollary 3.1.** Positive polarity RM expansion for a totally symmetric switching function \(f\) of \(n\) variables is

\[
f = \sum_{i=0}^{n} r^{(i)} \sum_{j_1 + \ldots + j_n = i} x_1^{j_1} \ldots x_n^{j_n}, \quad (3.5)
\]

where \(r^{(i)} \in \{0, 1\}\) is \(i\)-th coefficient, \(\sum\) denotes exclusive OR, \(j_t = 0\) or \(1\) represents...
the absence or presence of } x_t \text{ in a product } \text{ term accordingly, and } x^{j_i} \text{ is defined from } (2.2).

\textbf{Proof.} The proof follows directly from Theorem 3.1 \hfill \Box

The coefficients } r^{(i)} \text{ can be calculated by the truncated RM transform approach [11] originally proposed by Davio [8].

\textbf{Remark 3.2.} Theorem 3.1 can be written for } (2^n - 1)\text{-polarity RM by complementing all variables.

\textbf{Example 3.10.} The positive polarity RM expansion of totally symmetric functions of 3 variables in accordance with Corollary 3.1 is

\[ f = \sum_{i=0}^{3} r^{(i)} \sum_{j_1+j_2+j_3 = i} x_1^{j_1}x_2^{j_2}x_3^{j_3} \]

Here the solutions for indexes } j_1, j_2, \text{ and } j_3 \text{ are: } j_1j_2j_3 = \{000\} \text{ for } i = 0; j_1j_2j_3 = \{010, 010, 100\} \text{ for } i = 1; j_1j_2j_3 = \{011, 101, 110\} \text{ for } i = 2; \text{ and } j_1j_2j_3 = \{111\} \text{ for } i = 3. \text{ Hence, } f = r^{(0)}x_1^0x_2^0x_3^0 \oplus r^{(1)}(x_1^0x_2^0x_3^1 \oplus x_1^0x_2^1x_3^0 \oplus x_1^1x_2^0x_3^0 \oplus r^{(2)}(x_1^0x_2^1x_3^0 \oplus x_1^1x_2^0x_3^0 \oplus x_1^1x_2^1x_3^0 \oplus r^{[3]}x_1^1x_2^1x_3^1.

Finally, } f = r^{(0)} \oplus r^{(1)}(x_3 \oplus x_2 \oplus x_1) \oplus r^{(2)}(x_2x_3 \oplus x_1x_3 \oplus x_1x_2) \oplus r^{[3]}x_1x_2x_3. \hfill \Box

\textbf{3.2 Strategy to recognize partial and total symmetries}

\textbf{3.2.1 Partially symmetric functions}

Below we consider a strategy to detect partial symmetries in a given switching function based on the Theorem 3.1.

\textbf{Corollary 3.2.} The necessary and sufficient condition for a switching function } f \text{ to be partially symmetric with respect to variables } \{x_{j_1}^{i_1} \ldots x_{j_k}^{i_k}\}, \text{ in RM form, is that there are exactly } C_k^i \text{ or none of products } x_1^{j_1} \ldots x_{j_1}^{i_1} \ldots x_{j_k}^{i_k} \ldots x_n^{i_n} \text{ for every value } i \in \{1, 2, \ldots, k - 1\} \text{ and condition } j_{t_1} + \ldots + j_{t_k} = i.

\textbf{Proof.} The condition is obviously necessary, it follows directly from Theorem 3.1. Its sufficiency is the direct consequence of the unique representation of a function in positive polarity RM form (3.4). \hfill \Box

\textbf{Example 3.11.} Let us check if the function } f = 1 \oplus x_1x_3 \oplus x_1x_4 \oplus x_3x_4 \oplus x_2 \oplus x_1x_3x_4 \text{ is partially symmetric with respect to variables } \{x_1, x_3, x_4\}.

It has to include } C_k^i \text{ of none of products } x_1^{j_1}x_2^{j_2}x_3^{j_3}x_4^{j_4} = x_1^{j_1}x_2^{j_2}x_3^{j_3}x_4^{j_4} \text{ for every value } j_1 + j_3 + j_4 = i \text{ where } i = 1, 2.

(a) Let } j_1 + j_3 + j_4 = 1. \text{ Then } j_1j_2j_3j_4 = \{001, 010, 100\} \text{ when } j_2 = 0 \text{ or } j_1j_2j_3j_4 = \{0101, 0110, 1100\} \text{ when } j_2 = 1. \text{ So, } f \text{ has to include } C_3^1 = 3 \text{ or none of single products: } x_1, x_3, x_4 \text{ or double products } x_1x_2, x_2x_3, x_3x_4. \text{ In fact, the given function includes none of these products.

(b) Let } j_1 + j_3 + j_4 = 2, \text{ then } j_1j_2j_3j_4 = \{011, 1001, 1010\} \text{ when } j_2 = 0 \text{ or } j_1j_2j_3j_4 = \{0111, 1101, 1110\} \text{ when } j_2 = 1. \text{ Function } f \text{ has to contain the following triples (} C_3^2 = 3\text{) of products: } x_1x_3, x_1x_4, x_3x_4 \text{ and } x_2x_3x_4, x_1x_2x_4, x_1x_2x_3. \text{ The given function } f \text{ contains products } x_1x_3, x_1x_4, x_3x_4. \text{ So, } f \text{ is partially symmetric with respect to variables } \{x_1, x_3, x_4\}. \hfill \Box

Recognition of partial symmetry can be made by comparing the RM coefficients with the correspondent indexes.
Example 3.12. (Continuation)
The column coefficients vector be \( \mathbf{R} = [r^{(1)} \ldots r^{(15)}] = [1001 1000 0111 0000] \).
Calculate the indexes of coefficients \( r^{(i)} \) when \( j_1 + j_3 + j_4 = 1 \).
Case \( j_2 = 0 \) : \( j_1 j_2 j_3 j_4 = \{0001, 0010, 1000\} \), i.e. \( i = 1, 2, 8 \).
Case \( j_2 = 1 \) : \( j_1 j_2 j_3 j_4 = \{0101, 0110, 1100\} \), i.e. \( j = 5, 6, 12 \).
The first group of coefficients take value 0, as well as the second. Now calculate the indexes of coefficients \( r^{(i)} \) when \( j_1 + j_3 + j_4 = 2 \) : \( i = 3, 9, 10 \) (case \( j_2 = 0 \)) and \( i = 7, 13, 14 \) (case \( j_2 = 0 \)).
The coefficients with indexes 3, 9, 10 and 7, 13, 14 take the value 1 and 0 accordingly.
So, this function is symmetric with respect to variables \( \{x_1, x_3, x_4\} \).

3.2.2 Totally symmetric functions
Following Corollary 3.2, we obtain the well-known result that the necessary and sufficient condition for total symmetry for a switching function in the RM form is that all \( C_n^i \) or none of products \( x_1^{j_1} x_2^{j_2} \ldots x_n^{j_n} \) occur for every value \( i \in \{1, 2, \ldots, n - 1\} \) and \( \sum_{i=1}^{n} j_i = \).

Example 3.13. Recognize if the next functions are totally symmetric:
(a) \( x_1 x_2 \oplus x_2 x_3 \oplus x_1 x_4 \),
(b) \( x_1 \oplus x_2 \oplus x_3 \oplus x_1 x_2 x_3 \),
(c) \( 1 \oplus x_1 x_2 x_3 \oplus x_1 x_3 \).
The function (a) is not totally symmetric because there are 3 product term with 2 literals, we obtain 3 products that is less than \( C_n^2 \). By analogy we can recognize functions (b) and (c) as totally symmetric.

On the other hand, it follows from the statement 3.1 that a necessary and sufficient condition for totally symmetry in a switching function in the positive RM form is that the coefficients \( r^{(i)} \) take the same value if their indexes satisfy the equation \( \sum_{i=1}^{n} j_i = \) for \( i = 1, 2, \ldots, n - 1 \) (we don’t consider the trivial cases \( i = 0 \) and \( n \)). Thus, the basis for another algorithm.

Example 3.14. Recognize total symmetry in the RM expansions given in Example 3.13. The column coefficients vector be \( \mathbf{R} = [r^{(0)} r^{(1)} \ldots r^{(15)}] = [0000 0010 0100 1000] \).
(i) Calculate the indexes of coefficients \( r^{(i)} \). Start from \( i = 1, 2, 4, 8 \). When \( \sum_{i=1}^{4} j_i = \) : \( j_1 j_2 j_3 j_4 = \{0001, 0010, 0100, 1000\} \). These coefficients take the same value 0. Now calculate the indexes of \( r^{(i)} \) when \( \sum_{i=1}^{4} j_i = 2 \) : \( i = 3, 5, 6, 9, 10, 12 \). The coefficients with indexes \( i = 6, 9, 12 \) and \( i = 3, 5, 10 \) take different values. So, this function is not totally symmetric.
(ii) The coefficients \( r^{(i)} = 1 \) when \( \sum_{i=1}^{4} j_i = 1 \) (\( i = 1, 2, 4 \)), and the value of \( r^{(i)} = 0 \) when \( \sum_{i=1}^{4} j_i = 2 \) (\( i = 3, 5, 6 \)). So, this function is totally symmetric.
(iii) The coefficients \( r^{(i)} = 0 \) when \( \sum_{i=1}^{4} j_i = 1 \) but the values of coefficients \( r^{(i)} \) for \( \sum_{i=1}^{4} j_i = 2 \) are different, so the function is not totally symmetric.

4 Experimental results
4.1 Partially symmetric functions
The recognition program RECSym includes an INDEX GENERATOR to generate indices of the RM coefficients that have to be checked for equality, and a COMPARATOR that analyzes the
indices and values of coefficients. The EXOR minimizer is considered as an input data generator for our program which transforms the input switching function to a positive polarity RM expression.

### 4.1.1 Minimizer

We have used a minimizer based on the staircase strategy, originally developed by Zakrevskij [30] for minimizing switching functions in the FPRM form. This strategy can minimize incompletely specified functions, but it is also well-suited for completely specified functions too. Moreover, the minimizer based on this strategy allows us to find exact and near optimal FPRM expressions. Further descriptions of this strategy can be found in [22] and [28].

### 4.1.2 Index generator

To recognize the symmetry of a function in \( k \) variables \( x_1^{j_1} \ldots x_t^{j_t} \ldots x_n^{j_n} \), we need to generate the indices of the coefficients to be compared. The INDEX GENERATOR calculates sets of indexes \( \{j_1 \ldots j_t \ldots j_n\}_i \) for \( i = 0, 1, 2, \ldots, \theta - 1 \) that satisfy the equation \( j_t + \ldots + j_k = i \), while fixing the values of the other indexes and forming the sets of distinct assignments with respect to the algorithm described in Section 3.1.3.

### 4.1.3 Experiments

The proposed algorithm has been implemented as program RECSym in C++ on a Pentium 200MMX processor. To verify the efficiency of our approach, we tested our recognition program RECSym on MCNC benchmark functions with 4-15 variables. Table 3 and Table 4 contain a fragment of our results. The column with label \textbf{In} shows the numbers of variables. The column \textbf{P/L} refers to the number of products (\( P \)) and literals (\( L \)) in the positive RM expression (input data for our recognition system), respectively. The column labeled \textbf{t} refers to the CPU time of the recognition in seconds. Our program have manipulated about one thousand RM coefficients as input data, i.e. product terms.

Consider some results in detail. For test \( f_2 \), our recognizer found partial symmetries in variables for output functions \( f_{21}, f_{22}, f_{23} \) of this 4-output test.

The second of the output functions \( f_{22} \) of this test is partially symmetric.

<table>
<thead>
<tr>
<th>Test</th>
<th>In</th>
<th>\textbf{P/L}</th>
<th>\textbf{Symm.}</th>
<th>\textbf{t}</th>
</tr>
</thead>
<tbody>
<tr>
<td>f21</td>
<td>4</td>
<td>4/10</td>
<td>( x_1, x_2, x_4 )</td>
<td>0.00</td>
</tr>
<tr>
<td>f22</td>
<td>4</td>
<td>4/10</td>
<td>( x_1, x_3, x_4 )</td>
<td>0.00</td>
</tr>
<tr>
<td>f23</td>
<td>4</td>
<td>4/10</td>
<td>( x_2 - x_4 )</td>
<td>0.00</td>
</tr>
<tr>
<td>bw13</td>
<td>5</td>
<td>8/26</td>
<td>( x_2, x_4 )</td>
<td>0.00</td>
</tr>
<tr>
<td>bw18</td>
<td>5</td>
<td>8/41</td>
<td>( x_2, x_4 )</td>
<td>0.00</td>
</tr>
<tr>
<td>bw2</td>
<td>5</td>
<td>4/13</td>
<td>( x_4, x_5 )</td>
<td>0.00</td>
</tr>
<tr>
<td>bw24</td>
<td>5</td>
<td>20/51</td>
<td>( x_1, x_2 )</td>
<td>0.00</td>
</tr>
<tr>
<td>bw25</td>
<td>5</td>
<td>18/47</td>
<td>( x_2, x_4 )</td>
<td>0.00</td>
</tr>
<tr>
<td>bw27</td>
<td>5</td>
<td>16/41</td>
<td>( x_2, x_4 )</td>
<td>0.00</td>
</tr>
<tr>
<td>bw3</td>
<td>5</td>
<td>16/42</td>
<td>( x_1, x_4 )</td>
<td>0.00</td>
</tr>
<tr>
<td>5x01</td>
<td>7</td>
<td>16/68</td>
<td>( x_3, x_4 )</td>
<td>0.00</td>
</tr>
<tr>
<td>5x6</td>
<td>7</td>
<td>5/9</td>
<td>( x_5 - x_7 )</td>
<td>0.00</td>
</tr>
<tr>
<td>5x7</td>
<td>7</td>
<td>3/4</td>
<td>( x_1, x_5 - x_7 )</td>
<td>0.00</td>
</tr>
<tr>
<td>f53</td>
<td>8</td>
<td>11/32</td>
<td>( x_1, x_2 )</td>
<td>0.00</td>
</tr>
<tr>
<td>f55</td>
<td>8</td>
<td>5/9</td>
<td>( x_1 - x_4 )</td>
<td>0.00</td>
</tr>
<tr>
<td>f56</td>
<td>8</td>
<td>3/4</td>
<td>( x_1 - x_3 )</td>
<td>0.00</td>
</tr>
<tr>
<td>sac21</td>
<td>10</td>
<td>376/1832</td>
<td>( x_6, x_{10} )</td>
<td>0.03</td>
</tr>
<tr>
<td>sac22</td>
<td>10</td>
<td>512/2592</td>
<td>( x_6, x_{10} )</td>
<td>0.04</td>
</tr>
<tr>
<td>sac24</td>
<td>10</td>
<td>936/4536</td>
<td>( x_6, x_{10} )</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 3: Partially symmetric functions with respect to one group of variables: results of recognition of in MCNC benchmarks.
\[
\rho = \{x_1, x_3, x_5\} \text{ and } \{x_2, x_4\}; \quad 5 \times 10 \text{ is } \\
\rho\text{-symmetric with respect to variables} \\
\rho = \{x_1, x_7\}, \{x_2, x_3, x_4\} \text{ and } \{x_5, x_6\}; \\
z_{41} - z_{44} \text{ are also } \rho\text{-symmetric, as well as } f_{57}.
\]

### 4.2 Totally symmetric functions

#### 4.2.1 Index generator

The INDEX GENERATOR, of course, is applicable to totally symmetric functions. Note, that the symmetry vector includes \(k\) 1’s: \(S = [11...11]\) in this case.

#### 4.2.2 Experiments

The results of the experimental study are presented in Table 5.

<table>
<thead>
<tr>
<th>Test</th>
<th>In</th>
<th>(P/L)</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>rd531</td>
<td>5</td>
<td>5/20</td>
<td>0.00</td>
</tr>
<tr>
<td>rd532</td>
<td>5</td>
<td>5/5</td>
<td>0.00</td>
</tr>
<tr>
<td>rd533</td>
<td>5</td>
<td>10/20</td>
<td>0.00</td>
</tr>
<tr>
<td>rd731</td>
<td>7</td>
<td>21/42</td>
<td>0.00</td>
</tr>
<tr>
<td>rd732</td>
<td>7</td>
<td>7/7</td>
<td>0.00</td>
</tr>
<tr>
<td>rd733</td>
<td>7</td>
<td>35/140</td>
<td>0.00</td>
</tr>
<tr>
<td>rd841</td>
<td>8</td>
<td>28/56</td>
<td>0.01</td>
</tr>
<tr>
<td>rd842</td>
<td>8</td>
<td>8/8</td>
<td>0.01</td>
</tr>
<tr>
<td>rd844</td>
<td>8</td>
<td>70/280</td>
<td>0.01</td>
</tr>
<tr>
<td>0sym</td>
<td>9</td>
<td>230/756</td>
<td>0.06</td>
</tr>
<tr>
<td>sym10</td>
<td>10</td>
<td>266/1300</td>
<td>0.14</td>
</tr>
</tbody>
</table>

### Table 4: The functions symmetric with respect to sets of variables: results of recognition of in MCNC benchmarks

<table>
<thead>
<tr>
<th>Test</th>
<th>In</th>
<th>(P/L)</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>bw11</td>
<td>5</td>
<td>16/10</td>
<td>0.00</td>
</tr>
<tr>
<td>5x10</td>
<td>7</td>
<td>7/34</td>
<td>0.01</td>
</tr>
<tr>
<td>5x5</td>
<td>7</td>
<td>7/36</td>
<td>0.00</td>
</tr>
<tr>
<td>z41</td>
<td>7</td>
<td>15/56</td>
<td>0.00</td>
</tr>
<tr>
<td>z42</td>
<td>7</td>
<td>9/22</td>
<td>0.00</td>
</tr>
<tr>
<td>z43</td>
<td>7</td>
<td>5/8</td>
<td>0.00</td>
</tr>
<tr>
<td>z44</td>
<td>7</td>
<td>3/3</td>
<td>0.00</td>
</tr>
<tr>
<td>f54</td>
<td>8</td>
<td>7/16</td>
<td>0.00</td>
</tr>
<tr>
<td>f57</td>
<td>8</td>
<td>2/2</td>
<td>0.00</td>
</tr>
<tr>
<td>newtag</td>
<td>8</td>
<td>21/89</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 4 represents the results of recognition the sets of partial symmetries by our program. Consider, for example, function \(f_{57}\) (Table 4). This function is shown to be partially symmetric with respect to two sets of variables \(\{x_1 - x_6\}\) and \(\{x_7, x_8\}\). Some of the function are \(\rho\)-symmetric. For instance, \(bw11\) is \(\rho\)-symmetric with respect to variables \(x_1, x_3, x_5; x_2, x_4; x_3, x_5; x_3, x_6; x_3, x_5\).

Consider, for example, function \(rd531\). The result of minimization is a positive RM expression with 5 products and 20 literals: 

\[
x_{2} x_{3} x_{4} x_{5} \oplus x_{1} x_{3} x_{4} x_{5} \oplus x_{1} x_{2} x_{4} x_{5} \oplus x_{1} x_{2} x_{3} x_{5} \oplus x_{1} x_{2} x_{3} x_{5}
\]

with respect to \(x_1, x_3, x_4\). Output \(f_{23}\) is symmetric with respect to the following variables: \(x_2, x_3, x_4\); function \(f_{24}\) is symmetric with respect to variables \(x_1, x_2, x_3\) (Table 3). Note that the first output variable \(f_{21}\) is totally symmetric (Table 5).
Our program recognized it as a totally symmetric function.

Functions \(rd532\), \(rd732\) and \(rd842\) are recognized as totally symmetric too, because these function are linear RM expansion \(\sum_i x_i\) for \(i = 5, 7\) and 8 respectively.

Consider benchmark \(9sym\) that is a completely specified 9-input single-output function, the output of which is 1 only when the weight of an input vector is one of \(\{3,4,5,6\}\). \(9sym\) is totally symmetric function and our program recognizes that it has 210 terms and 756 literals of positive RM expansion \((t = 0.14\) sec. of CPU time is required\).

5 Concluding remarks

In this paper, we have extended the feasible recognition of symmetries in the RM domain, namely, to partially and totally symmetric switching functions. We have shown the advantages of our formal approach for representation of different types of symmetries. The main theoretical results include

1. Positive polarity RM expansion for partially symmetric function, i.e. we deal with more general case of symmetry.

2. Necessary and sufficient conditions to recognize mentioned above symmetries.

We have realized advantages of formal approach in our program RECSym. Program RECSym successfully recognizes partial and total symmetries in positive polarity RM expansion of about 50 circuits.

We have observed some interesting effects in our study. For example, most of the benchmark functions used in our experiments (with 4-15 variables) was identified as partially symmetric functions. However, the main limitation of our program and theoretical results is to the positive polarity RM expression. However, there may be advantages to allowing other polarities (i.e. FPRM), in which one or more variables appear complemented. Recognition of symmetries in the FPRM expansion with arbitrary polarity is an area for future research. In addition, it will be interesting to extend our results to specific types of symmetric functions, self-dual and anti-self-dual functions, as well as symmetric functions that are also symmetric in logic values (e.g. multivalued functions as described in [4]).

References


