Proceedings of the IAPR International Conference on Pattern Recognition and Information Processing, PRIP'99, Minsk, Belarus, 1999, pp. 197-206

FIXED POLARITY REED-MULLER MINIMIZATION OF INCOMPLETELY SPECIFIED BOOLEAN FUNCTIONS BASED ON INFORMATION ESTIMATIONS ON DECISION TREES*

DENIS POPEL

Belarusian State University of Informatics and Radioelectronics,
Laboratory of Image Processing and Pattern Recognition.
Brovky Str., 6, 220600, Minsk, Republic of Belarus,
Tel.: (+172) 491-981, Fax: (+172) 310-914, e-mail: popel@bseu.minsk.by, popel_denis@yahoo.com

Abstract. This paper presents algorithm to find minimal Fixed Polarity Reed-Muller expressions, two-level fixed polarity AND-EXOR canonical representations, for incompletely specified Boolean functions that based on information measures on decision trees. We study the *Free Reed-Muller Tree* as acceptable representation and manipulation structure to find minimal Fixed Polarity Reed-Muller expressions. In contrast to previously published methods, the algorithm can handle incompletely specified Boolean functions up to one thousand variables for reasonable time.

Key words: *Boolean functions minimization, information estimations, Free Reed-Muller Tree (FRMT)*

INTRODUCTION

AND-EXOR representations are functional outside the area of logic design. They can be of use in *image processing, coding* and *recognition problems* (see, for example, [14]). They have been also utilized as efficient data structures for logical forms manipulation in Computer-Aided Design (CAD) systems, being a base of automatic theorem provers and logic programming languages like Prolog. The above reasons have recently caused an increased interest in solution classical but yet unsolved problems of finding minimal AND-EXOR representations. In this paper we attract attention to the problem of searching for minimal Fixed Polarity Reed-Muller (FPRM) expressions, two-level fixed polarity AND-EXOR canonical representations, for incompletely specified Boolean functions.

* This work was partially supported from Fund of Fundamental Researches (BELARUS). The partially support from the Institute of Computer Science, Technical University of Szczecin (POLAND), is acknowledged.

Two contrasting approaches are used for minimization of incompletely specified Boolean functions.

The first one is based on *over-defining* the unspecified values of the function. Exhaustive search for minimal expression proposed by D.Green [7] in which all possible values assigning to the points where the function is not specified. L.McKenzie et al. paper [10] presents a branch and bound search where complexity is double exponential in the number of variables.

The algorithm to find minimal AND-EXOR representation of incompletely specified Boolean functions was presented by M.Perkowski and M.Chrznowska-Jaske [12]. The further improvement of the algorithm was proposed by C.Chang and B.Falkowski [1] for fixed polarity multiple-output functions, hence this method required very large computation expenses when the Boolean functions are weakly specified. The sparse interpolation method for minimization of incompletely specified Boolean functions was considered by Z.Zilic and Z.Vranesic [21].

The another approach for minimization of incompletely specified Boolean functions is based on principle "*don't care about don't care*". A.Zakrevskij has proposed algorithms to find a minimal Positive Polarity Reed-Muller expressions for weakly specified function [19] and for system of weakly specified Boolean functions [20]. The algorithm for one Boolean function is based on tree search along the extended matrix formed from the truth table. The algorithms for a system of weakly specified Boolean functions are grounded on a theory of linear vector spaces, operate with a basis of nil-space.

Research group headed by V.Shmerko and S.Yanushkevich has developed some classes of so called *Staircase algorithms*: for minimization of incompletely specified multiple-valued logic functions [5,8,9,16] that generalized algorithm proposed by A.Zakrevskij.

We present an algorithm for finding minimal FPRM expressions for incompletely specified Boolean functions that based on converting given function in the form of truth table to special data structure – *decision tree*. The core of proposed algorithm is principle "*don't care about don't care*". Decision trees and in general case decision diagrams can extensively used in many application of decision making systems, such as *pattern recognition*, *decision support systems* and etc. for such reasons:

- make possible to represent hierarchical logical and probability model of investigated process;
- allow to use characteristics of mixed nature: qualitative as well as quantitative.

Traditionally decision trees and diagrams are observed in logic design as logical model without any qualitative characteristics. This paper presents algorithms based on information estimations that allow to extent usual knowledge of decision trees and diagrams [2,3,13,18].

We show that proposed algorithm for FPRM minimization is fast in contrast to previously published works. We succeeded in minimizing incompletely specified functions up to one thousand variables in acceptable time. Experimental results are given to show the efficiency of information measures.

The paper is organized as follows. Section 1 presents terminology. Section 2 outlines background of investigation and presents strategy of FPRM minimization based on conversion truth table to decision tree. Section 3 outlines the minimization algorithm InfoFPRM that based on information measures. Section 4 presents extended experimental results and Section 5 concludes the paper.

1. TERMINOLOGY

$f = f(x_1, x_2, \ldots, x_n)$	Boolean function of n variables $x_1, x_2, \ldots, x_n$
$X = \{x_1, x_2, \ldots, x_n\}$	Set of variables of Boolean function f
$u_j = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$	*Combination* of variables values $x_1, x_2, \ldots, x_n$
$U = \{u_1, u_2, \ldots, u_K\}$	Set of combinations u_j - *domain* of Boolean function f
$F = \{0, 1\}$	Set of function values - *range* of Boolean function f
$D = \{U, F\}$	Truth table
$\Phi = \{\Pi, V\}$	Decision tree with nodes set Π and vertex set V

2. BACKGROUND

The basic task to be solved in this paper can be formulated as follows. Given an incompletely specified Boolean function f of n variables in the form of truth table $D = \{U, F\}$, where U - domain of logic function f, F - range of logic function f. Can one find an minimal FRPM expression for given incompletely specified Boolean function through conversion truth table to decision tree, which correspond to FPRM expression.

Table 1. A truth table for incompletely specified Boolean function *f*.

U	x_1	x_2	x_3	x_4	x_5	f
u_1	0	0	0	0	1	1
u_2	0	1	0	0	0	0
u_3	0	1	0	1	1	0
u_4	1	1	0	1	1	1
u_5	1	0	1	1	1	1
u_6	1	1	1	1	1	1

Example 1. Boolean function f which is specified for $K = 6$ combinations of variables values is given in Table 1.

Let give some essential definitions and prepositions which are important for the understanding the paper.

Definition 1. *Decision Tree* (*DT*) over X is directed acyclic graph $\Phi = \{\Pi, V\}$ with nodes set Π and vertex set V. Each node $\pi \in \Pi$ is labeled with possible expansion on variable $x_j \in X$ called *arbitrary variable*. Each node has exactly one incoming vertex and two outcoming vertexes, which correspond decomposition step of Boolean function on two subfunctions - f_{low} and f_{high}. Terminal vertex is labeled with leaves *value* and has no successors, nonterminal vertex is labeled with v and has exactly two successors denoted by *low* and *high*.

According to T.Sasao classification [15] we consider *Reed-Muller Tree*, such decision tree have nodes which correspond two types of expansions – positive Davio and negative Davio. The rules of positive Davio and negative Davio expansions for arbitrary variable x_j are presented in Table 2.

Example 2. Reed-Muller Tree for the function from Table 1 is given in Fig. 1.

Definition 2. *Free Reed-Muller Tree* (FRMT) is decision tree where each variable is encountered at most once in the decision tree from the root node to terminal vertex.

Free Binary Decision Trees and Diagrams have been studied by J.Gergov and C.Meinel [6].

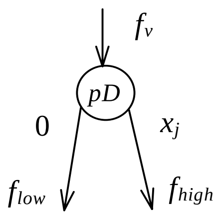
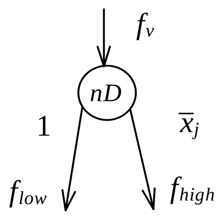
Example 3. Fig. 2 presents a FRMT for the function given in Table 1.

Let represent Fixed Polarity Reed-Muller (FPRM) expression as follows:

$$f(x_1, ..., x_n) = c_0 \oplus c_1 \cdot \overset{p}{x}_1 \oplus ... \oplus c_n \cdot \overset{p}{x}_n \oplus c_{n+1} \cdot \overset{p}{x}_1 \cdot \overset{p}{x}_2 \oplus ... \oplus c_{2^n-1} \cdot \overset{p}{x}_1 \cdot \overset{p}{x}_2 \cdot ... \cdot \overset{p}{x}_n,$$

where $c_i = \{0, 1\}$ are the coefficients of FPRM expression and $\overset{p}{x}_i = \{x_i, \overline{x_i}\}$, where p are called polarity of variable x_i. FPRM expression is canonical representation of Boolean function f if the polarity of each variable is fixed. The choice of polarity largely influences the size of the resulting FPRM expression, as is shown by the following example (the size of FPRM expressions are estimated by number of *terms* C_T and *literals* C_L).

Table 2. Positive Davio and negative Davio expansions.

	positive Davio expansion	***negative Davio expansion***
Short description of expansion	*pD*	*nD*
Expansion rule	$f_v = f_{low} \oplus x_j f_{high}$, $f_{low} = f(x_1, ..., x_{j-1}, 0, x_{j+1}, ..., x_n)$ $f_{high} = f(x_1, ..., x_{j-1}, 0, x_{j+1}, ..., x_n) \oplus f(x_1, ..., x_{j-1}, 1, x_{j+1}, ..., x_n)$	$f_v = f_{low} \oplus \overline{x_j} f_{high}$, $f_{low} = f(x_1, ..., x_{j-1}, 1, x_{j+1}, ..., x_n)$ $f_{high} = f(x_1, ..., x_{j-1}, 0, x_{j+1}, ..., x_n) \oplus f(x_1, ..., x_{j-1}, 1, x_{j+1}, ..., x_n)$
Graphical presentation of expansion		

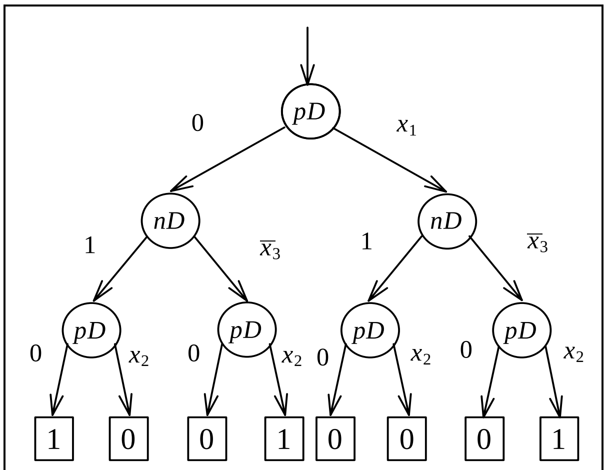


Fig.1. An example of Reed-Muller Tree.

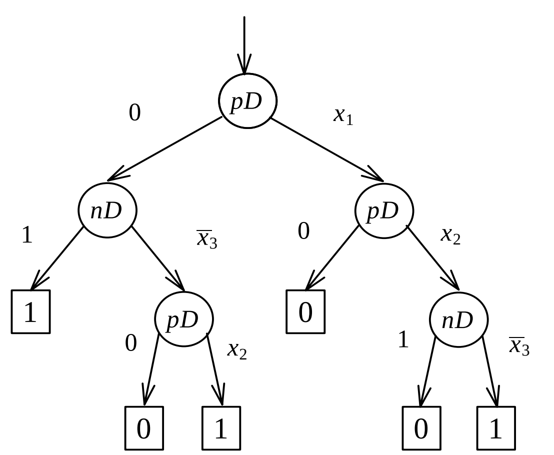
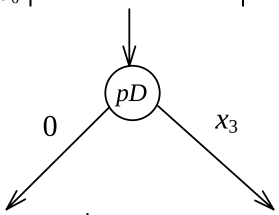


Fig.2. An example of Free Reed-Muller Tree.

Table 3. Truth table decomposition rules.

positive Davio expansion

U	x_1	x_2	x_3	x_4	x_5	f
u_1	0	0	0	0	1	1
u_2	0	1	0	0	0	0
u_3	0	1	0	1	1	0
u_4	1	1	0	1	1	1
u_5	1	0	1	1	1	1
u_6	1	1	1	1	1	1

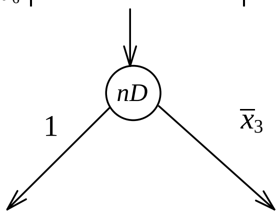


U_0	x_1	x_2	x_4	x_5	f_{low}
u_1	0	0	0	1	1
u_2	0	1	0	0	0
u_3	0	1	1	1	0
u_4	1	1	1	1	1

U_1	x_1	x_2	x_4	x_5	f_{high}
u_5	1	0	1	1	0
u_6	1	1	1	1	0

negative Davio expansion

U	x_1	x_2	x_3	x_4	x_5	f
u_1	0	0	0	0	1	1
u_2	0	1	0	0	0	0
u_3	0	1	0	1	1	0
u_4	1	1	0	1	1	1
u_5	1	0	1	1	1	1
u_6	1	1	1	1	1	1



U_0	x_1	x_2	x_4	x_5	f_{low}
u_5	1	0	1	1	1
u_6	1	1	1	1	1

U_1	x_1	x_2	x_4	x_5	f_{high}
u_1	0	0	0	1	0
u_2	0	1	0	0	1
u_3	0	1	1	1	1
u_4	1	1	1	1	0

Example 4. Two possible FPRM expressions for the function given by Table 1 be:

$f = \overline{x}_1 \cdot x_4 \oplus x_5$ ($C_T = 2$, $C_L = 3$) and $f = 1 \oplus \overline{x}_3 \cdot x_2 \oplus \overline{x}_3 \cdot x_1 \cdot x_2$ ($C_T = 3$, $C_L = 5$)

Where exist closer relation between considered forms - FRMT and FPRM expression. We now investigate this relation that directly outlines methods for construction of minimal FPRM expression.

Construction of FRMT is carried directly by the truth table decomposition according positive Davio and negative Davio expansions (see Table 3), which are recursively applied for given function in the form of truth table. In the Table 4 we have closer look at the correspondence of notations in FRMT and FPRM.

Using FRMT it is possible to represent Boolean functions efficiently than only have other types of decision trees, for example Positive Davio Tree representation. But as well-known that decision tree/diagram representation are very sensitive to the variable ordering (see, for example, [11]). The size of decision tree/diagram and consequently the size of the corresponding expression is very sensitive to the variable ordering and the choice of the type of expansion for tree nodes (see, for instance, [4]).

Table 4. Close relation between Free Reed-Muller Tree and Fixed Polarity Reed-Muller expression.

Free Reed-Muller Tree		*Fixed Polarity Reed-Muller expression*
< the root π_0 of the tree>	$\Leftrightarrow$	< given Boolean function f >
< the node π_i of the tree>	$\Leftrightarrow$	< the variable x_i of Boolean function f >
< the vertex v of the tree>	$\Leftrightarrow$	< subfunction f_v >
< type of expansion of the node π_i $\{pD, nD\}$>	$\Leftrightarrow$	< polarity p of variable x_i >
< the path in the tree>	$\Leftrightarrow$	< the term of Boolean function f >
< the leaf *value* of the tree>	$\Leftrightarrow$	< the coefficient c_l >

Thus, in proposed algorithm we consider the following tasks, that will be solved by information approach:

1. *How can we determine a variable ordering for an FRMT representing a given Boolean function f such that the number of terms in the corresponding FPRM expression is minimized?*
2. *How can we determine a type of expansion for arbitrary variable during FRMT construction such that the number of terms in the corresponding FPRM expression is minimized?*

3. Minimization Algorithm

We propose an algorithm to find minimal Fixed Polarity Reed-Muller expression via Free Reed-Muller Tree construction. The algorithm called *InfoFPRM* (*Information minimizer of Fixed Polarity Reed-Muller expressions*) is described in Fig.3.

We used information estimations for variable ordering and selection of expansion type for arbitrary variable, presented in [13]. Information estimations are based on *non-probabilistic* notations of Shannon entropy, so called *functional entropy*, proposed by D.Simovici [17]. The using of the functional entropy provides the calculation profit, for example, we do not need to use division.

Let briefly outline estimations that used in minimization algorithm. The Boolean function given by truth table can be described by the number of combinations of variables values $k|_{x_j=a}$ and $k|_{\substack{f=b\\x_j=a}}$. Where $k|_{x_j=0} = k_1$ means that variable x_j takes value $x_j = 0$ k_1 times; $k|_{\substack{f=0\\x_j=1}}$ means number of combinations for the case $(x_j = 1) \wedge (f = 0)$.

Example 5. A Boolean function f from Table 1 is defined by number of combination of variables values $k|_{x_3=0} = 4$, $k|_{x_3=1} = 2$, $k|_{\substack{f=0\\x_3=0}} = 2$, $k|_{\substack{f=1\\x_3=0}} = 2$, $k|_{\substack{f=0\\x_3=1}} = 0$, and $k|_{\substack{f=0\\x_3=0}} = 2$.

The information measure for positive Davio expansion:

$$\mathcal{H}^{pD}(x_j) = k|_{x_j=0} \log_2 k|_{x_j=0} + k|_{x_j=1} \log_2 k|_{x_j=1} - k|_{\substack{f=0\\x_j=0}} \log_2 k|_{\substack{f=0\\x_j=0}} - k|_{\substack{f=1\\x_j=0}} \log_2 k|_{\substack{f=1\\x_j=0}}. \quad (1)$$

The information measure for negative Davio expansion:

$$\mathcal{H}^{nD}(x_j) = k|_{x_j=0} \log_2 k|_{x_j=0} + k|_{x_j=1} \log_2 k|_{x_j=1} - k|_{\substack{f=0\\x_j=1}} \log_2 k|_{\substack{f=0\\x_j=1}} - k|_{\substack{f=1\\x_j=1}} \log_2 k|_{\substack{f=1\\x_j=1}}. \quad (2)$$

Example 6. For a given Boolean function (Table 1) let calculate information measures for positive Davio and negative Davio expansions for variable x_3.

$\mathcal{H}^{pD}(x_3) = 4 \log_2 4 + 2 \log_2 2 - 2 \log_2 2 - 2 \log_2 2 = 8 + 2 - 2 - 2 = 6$ bit.

$\mathcal{H}^{nD}(x_3) = 4 \log_2 4 + 2 \log_2 2 - 0 \log_2 0 - 2 \log_2 2 = 8 + 2 - 0 - 2 = 8$ bit.

The principle of minimum entropy are used for variable ordering:

```
/* (input) f = f (x1, x2, ..., xn): U → {0, 1}: {0, 1}^n→ {0, 1}*/
/*(output) fprm: Fixed Polarity Reed - Muller expression, Free Reed - Muller Tree */
InfoFPRM( U ) {
    if ( ∀u ∈ U: f = const ) { fprm←const; }
    else {
        for(∀ xi ∈ x1, x2, ..., xn) {
                Calculate information measures for positive Davio expansion (pD node)
                H^pD(xi) = Calculate_pD_Entropy( f, xi );          /* see (1) */
                Calculate information measures for negative Davio expansion (nD node)
                H^nD(xi) = Calculate_nD_Entropy( f, xi );          /* see (2) */
        }
        Choose variable xj for construction of tree node
        xj = ChooseVariable (H^pD, H^nD );                         /* see (3) */
        Choose expansion type of arbitrary variable xj for construction of tree node
        p = ChooseExpansionType(H^pD, H^nD );                      /* see (4) */
                                                                   /*p = 0 - positive Davio expansion */
                                                                   /*p = 1 - negative Davio expansion */
        U0 ← U| xj = p;                      /* the subfunction on xj = p */
        fprm0 ← InfoFPRM ( U0 );             /*recursively construct nodes for subfunction U0 */

        U1← U| xj = 1 - p;                   /* the subfunction on xj = 1 - p */
        f1 ← fprm0 ⊕ f;                      /* compute f1 */
        fprm1← InfoFPRM ( U1 );              /*recursively construct nodes for subfunction U1 */

        fprm ← fprm ⊕ xj^p·fprm;
        return fprm;
    }
}
```

Fig. 3. Algorithm of FPRM minimization based on information estimations for variable ordering and polarity selection.

$$x_j = \arg\min \left(\mathcal{H}^{pD}(x_j) \cup \mathcal{H}^{nD}(x_j)\right). \qquad (3)$$

The minimum of information measures for positive Davio and negative Davio expansions for a given arbitrary variable is described as follows:

$$p = \min \left(\mathcal{H}^{pD}(x_j), \mathcal{H}^{nD}(x_j)\right). \qquad (4)$$

Example 7. Consider the first minimization step of incompletely specified Boolean function f given in Table 1. Information measures for positive Davio and negative Davio expansions are presented in Table 5. According (3) and (4) variable x_5 with negative Davio expansion is chosen.

4. Experimental Results

The experiments were conducted with the program implementation of InfoFPRM algorithm - on Pentium 100 Mhz (RAM 48 Mb), programming language *C*++ under OS Windows 95. Boolean functions were random generated and processed with n and K varying to one thousand (truth table density is equal 50%).

We applied our method to 100 patterns of random functions and calculate the average of number of terms and literals. The random functions were generated using standard *C*++ library.

Table 5. Information measures (in bits) for positive and negative Davio expansions.

	$\mathcal{H}^{pD}(x_j)$	$\mathcal{H}^{nD}(x_j)$
x_1	7.51	4.75
x_2	8	6
x_3	6	8
x_4	10	5.25
x_5	11.61	**3.61**

The differences of number of terms, literals and time for Staircase and InfoFPRM algorithms are given in Table 6 and Table 7.

The results of minimization by Staircase algorithm [5], programs EXORCISM-MV-2 and ESPRESSO [16], and InfoFPRM algorithm have been compared on incompletely specified Machine Learning benchmarks in Table 6.

Statistical Properties

Taking advantage of our algorithm, we examined statistical properties of minimized functions. We investigated the relation between this method performance and the truth table density, which is the rate of ones in the truth table. We applied our method to the weighted random functions with 20 variables, 20 specified combinations and ranging from 0% to 100% in density. Fig. 4 shows that the behavior of terms and literals number in minimized FPRM expression is the same as functional information:

Table 6. Staircase algorithm [8] on random generated incompletely specified Boolean functions (truth table density is equal 50%).

$K = n$	**100**	**200**	**300**	**400**	**500**
C_T	21	46	59	87	108
C_L	65	167	252	369	480
Time[*]	0.13	1.67	5.72	19.56	49.56

[*]**Power Challenge 4-processors, OS Linux**

Table 7. InfoFPRM algorithm on random generated incompletely specified Boolean functions (truth table density is equal 50%).

$K = n$	**100**	**200**	**300**	**400**	**500**	**600**	**700**	**800**
C_T	19	43	68	86	116	144	173	204
C_L	60	140	233	323	413	517	615	718
Time[&]	0.23	1.65	2.98	6.15	12.2	23.4	35.6	69.4

[&]**Pentium 100Mhz processor, OS Windows 95**

$$\mathcal{I}(f) = K \log_2 K - k|_{f=0} \log_2 k|_{f=0} - k|_{f=1} \log_2 k|_{f=1}. \quad (5)$$

The number of terms is not symmetric and peeks at about 60% the number of literals are symmetric with a center at 50%, which is like functional information. This result suggests that the number of literals is better as a measure of the complexity of Boolean functions than is the number of terms, which fully correspond to S.Minato [10].

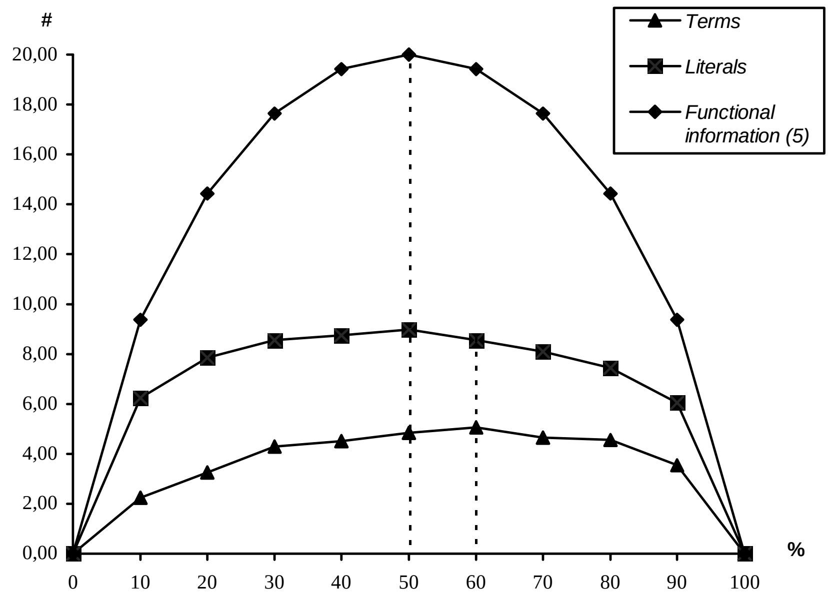


Fig.4. Number of terms and literals in minimized FPRM expression from truth table density.

5. CONCLUSIONS

The paper presents algorithm to find minimal Fixed Polarity Reed-Muller expression for given incompletely specified Boolean function in the form of truth table through construction of Free Reed-Muller Tree. Information measures are used for variable ordering and expansion type selection.

Note that suggested algorithm is designed to deal with any Boolean functions of *n* variables – it means that algorithm allow to minimize completely specified functions. Further investigation will carry out in the field of development effective algorithms for FPRM minimization of system of Boolean functions.

ACKNOWLEDGMENTS

Author would like to thank Prof. V. Shmerko, Dr. S. Yanushkevich and Dr. V. Cheushev for discussions and advises. Author are very grateful to Prof. J. Butler and Prof. G. Dueck who have

Table 8. Comparison of number of terms, literals and time for Staircase algorithm [5], EXORCISM-MV-2, ESPRESSO [16] and InfoFPRM algorithm.

	Input	Output	Staircase algorithm			EXORCISM-MV-2			ESPRESSO			InfoFPRM algorithm		
			C_T	C_L	*Time*[*]	C_T	C_L	*Time*[◊]	C_T	C_L	*Time*[◊]	C_T	C_L	*Time*[&]
add0	8	1	3	8	0.040	15	68	0.23	15	64	0.23	3	7	0.025
add4	8	1	6	12	0.040	2	2	0.50	2	4	0.50	2	2	0.001
ch177f0	8	1	2	2	0.040	2	2	0.50	2	4	0.50	2	2	0.001

***Cyrix 200Mhz processor, OS Linux**
◊ **Pentium 90Mhz processor, OS Linux**
& **Pentium 100Mhz processor, OS Windows 95**

made helpful suggestions and comments during this research.

REFERENCES

[1] Chang C., Falkowski B., Flexible Optimzation of Fixed Polarity Reed-Muller Expressions for Multiple Output Completely and Incompletely Specified Boolean Functions, *Proc. of the Asia and South Pacific Design Automation Conf.*, Chiba, Japan, 1995, pp. 335-340

[2] Cheushev V., Contextual Support in Document Understanding Systems, Chapter 2: Information Model of Document Recognition, PhD Thesis, Supervisor Shmerko V., *University of Informatics and Radioelectronics*, Minsk, Belarus, 1998

[3] Cheushev V., Shmerko V., Simovici D., Yanushkevich S., Functional Entropy and Decision Trees, *Proc. IEEE Int. Symp. on Multiple-Valued Logic*, 1998, pp. 357-362

[4] Drechsler R., Theobald M., Becker B., Fast OFDD Minimization of Fixed Polarity Reed-Muller Expressions, *IEEE Trans. on Computers*, vol. 45, no. 11, 1996, pp. 1294-1299

[5] Falkowski B., Holowinski G., Malecki K., Effective Minimization of Logic Functions in Reed-Muller Domain, *Proc. Int. Conf. on Application on Computer Systems - ACS'97*, Szczecin, Poland, 1997, pp. 248-255

[6] Gergov J., Meinel C., Efficient Boolean Manipulation with OBDDs can be extended to FBDDs, *IEEE Trans. on Comp.*, vol. C-43, no. 10, 1994, pp. 1197-1209

[7] Green D., Reed-Muller Expansions of Incompletely Specified Functions, *IEE Proc*., Pt.E., vol. 134, no. 5, 1987, pp. 228-236

[8] Holowinski G., Parallel Version of Generalized Zakrevskij's Algorithm for Minimization of Incompletely Specified Multi-Valued Logic Functions, *Proc. Int. Conf. on Patter Recognition and Information Processing*, Minsk, Belarus, 1997, pp. 332-339

[9] Jaroszewicz S., Shmerko V., Janushkevich S., Exact Irredundant Minimal Searing for a Reed-Muller Expansion for an Incompletely Specified MVL Functions, *Proc. Int. Conf. on Applications of Computer Systems - ACS'96*, Poland, 1996, pp. 65-74

[10] McKenzie L., Almaini A., Miller J., Thomson P., Optimization of Reed-Muller Logic Functions, *Int. J. Electronics*, vol. 75, no. 3, 1993, pp. 451-466

[11] Minato S., Binary Decision Diagrams and Application for VLSI CAD, *Cluwer Academic Publishes*, 1996

[12] Perkowski M. Chrznowska-Jeske M., An Exact Algorithm to Minimize Mixed-Radix Exclusive Sums of Products for Incompletely Specified Boolean Functions, *Proc. of the ISCAS Int. Symp. on Circuits and Systems*, New Orlean, USA, 1990, pp. 1652-1655

[13] Popel D., Information Approach to Constraction of Optimal Functional Decision Diagrams, *Proc. Third Int. Conf. on New Information Technologies in Education - NITE'98*, Minsk, Belarus, 1998, pp. 58-64

[14] Reddy B., Pai A., Reed-Muller Transform Image Coding, *Computer Vision, Graphics and Image Processing*, vol. 42, 1988, pp. 48-61

[15] Sasao T., Representation of Logic Functions Using EXOR Operators, *Proc. of the IFIP WG 10.5 Workshop on Application of the Reed-Muller Canonical Expansions in Circuit Design*, Japan, 1995, pp. 213-220

[16] Shmerko V., Holowinski G., Song N., Dill K., Ganguly K.,Safranek R., Perkowski M., High-Quality Minimization of Multi-Valued-Input, Binary-Output EXOR Sum and Multi-Valued Galois Sum of Product Expressions for Incompletely Specified Logic Functions, *Proc. Int. Conf. on Application of Computer Systems, Szczecin - ACS'97, Poland*, 1997, pp.266-286

[17] Simovici D., and C.Reischer, On Functional Entropy, *Proc. Int. Symp. on MVL*, 1993, pp. 100-104

[18] Simovici D., V. Shmerko, V. Cheushev, and S. Yanushkevich, Information Estimation for Logic Functions, *Proc. Int. Conf. On Application of Computer Systems*, Szczecin, Poland, 1997, pp. 287-300

[19] Zakrevskij A., and Toropov N., Optimizing Polynomial Implementation of Incompletely Specified Boolean Functions, *Proc. of the Int. Conf. on Computer - Aided Design of Discrete Devices,* Minsk, Belarus, 1995, pp. 85-92

[20] Zakrevskij A., Minimum Polynomial Implementation of Systems of Incompletely Specified Boolean Functions, *Proc. of the IFIP WG 10.5 Workshop on Application of the Reed-Muller Canonical Expansions in Circuit Design,* Japan, 1995, pp. 250-256

[21] Zilic Z., Vranesic Z., New Interpolation Algorithm for Multiple-Valued Reed-Muller Forms, *Proc. of the 26th Int. Symp. on Multiple-Valued Logic,* Santiago de Compostela, Spain, 1996, pp. 16-23