MODULAR DECOMPOSITION OF THE NOR-TSUM MULTIPLE-VALUED PLA

T. KALGANOVA†, N. LIPNITSKAYA†, G. HOLOWINSKI‡

†Belarusian State University of Informatics and Radiotechnics, Laboratory of Image Processing and Pattern Recognition. Brovky St., 6, 220600, Minsk, Republic of Belarus, Phone: (0375172) 49-19-81, Fax: (0375172) 495-106, E-mail: jack@expert.belpak.minsk.by or pottosina@risq.belcaf.minsk.by

‡Institute of Computer Science & Information Systems, Technical University, ul. Zolnierska 49, Szczecin, Poland, Fax. (+4891) 48-76439, E-mail: holowins@dedal.man.szczecin.pl

ABSTRACT. A method for designing PLA-based combinational circuits by modular decomposition is presented. Main subjects are 1) Specific properties of TSUM operator, 2) MIN-TSUM and NOR-TSUM expansions with respect to the bound set, X1 of variables, 3) Realization of functions by multiple-valued PLA-based combinational circuits, 4) Comparison with other methods. Experimental investigations show that the size of suggested combinational circuit is the same as the size of multiple-valued PLA implementing a multiple-valued logic function with large number of variables.

KEY WORDS. Multiple-valued logic, modular decompositions, logic design, multiple-valued programmable logic array (MV PLA)

1. Introduction

Multiple-valued logic is one of the active area of research. It is known that multiple-valued logic allows interconnections to be reduced into a chip, as well as a reduction in the numbers of inputs and outputs. Different MV PLA structures to implement the multiple-valued logic (MVL) functions have been proposed [1, 2, 3, 4]. These MV PLA's realize the logic functions using the MIN (r-valued minimum), MAX (r-valued maximum), MODSUM (modular sum), linear SUM and TSUM (truncated sum) arrays. The number of inputs, outputs and product terms of suggested MV PLA's have physical restrictions. These structures do not permits the MVL functions with the large number of variables to be implemented because of the number of inputs is smaller than the number of input variables. The solution of this problem is to use the functional decomposition technique. The decomposition of a logic function can lead to a systematic design which incorporates readily designed simpler functions. Ideally, a systematic design would posses such desirable properties as regularity and testability.

One approach to decomposition of multiple-valued logic functions (MVL functions) is to extend disjunctive decomposition for the binary case [5]. The main drawback of this approach is follows. Since only a very small number of functions will have a specific decomposition property [5], if the function does not have the desired property, the analysis does not yield a design. In other hand, the specific property of MVL functions is that an arbitrary logic function can be given by different set of multiple-valued operators. These determines the basic multiple-valued gates from which a complex digital circuit implemented given MVL function is synthesized.

In this paper a NOR-TSUM MV PLA, proposed by Pelayo and etc., is assumed to be the component used to implement the function. A decomposition approach for the modular design of combinational multiple-valued circuits based on the NOR-TSUM PLA is described. The primary objective of this approach is to develop procedures which are applicable to the design of PLA-based combinational circuits and to extent the class of decomposable functions.

The rest of the paper is structured as follows. Section 2 provides definitions and notations, the distinctive features of TSUM operator are investigated in Section 3, Section 4 is devoted to MIN-TSUM and NOR-TSUM decompositions of the MVL function; the MV PLA - based technique is explained in Section 5, experimental results are discussed in Section 6, the Section 7 draws a conclusion.

2. Definition and notations

Let us examine some r-valued operators by which we can represent any r-valued logic function. Let x and y be two r-valued variables. Then,

(a) \( x \land y = \text{MIN}(x, y) \);
(b) \( x \oplus y = \text{TSUM}(x+y, r-1) \);

\[
= r-1, \quad \text{if} \ x = s,
\]

where \( \text{MIN} \) and \( \text{TSUM} \) are the minimum and truncated sum operations, respectively.
A product of the literals \( x_1^{\sigma_1}x_2^{\sigma_2}\ldots x_n^{\sigma_n} \) is referred to as a \textit{product term} (also called \textit{term} or \textit{product} for short). In a product term, all the operations are AND operations. A product term that includes literals for all variables \( x_1, x_2, \ldots, x_n \) is called a full term. An \( r \)-valued minimum of term and \( r \)-valued constant, \( f(\sigma x_1^{\sigma_1}x_2^{\sigma_2}\ldots x_n^{\sigma_n}) \) is denoted as an \( r \)-valued minimum term (also called \textit{r-valued term}). An \( r \)-valued term that contains literals for all variables \( x_1, x_2, \ldots, x_n \) is called a \textit{full r-valued term}. A truncated sum of \( r \)-valued terms is called a truncated sum-of-minimums (TSOM) expression:

\[
\tau_{r-1}^{-1} f(X) = \sum_{\sigma = 0}^{r^{-1}} f(\sigma x_1^{\sigma_1}x_2^{\sigma_2}\ldots x_n^{\sigma_n})
\]

where the symbol \( \sum \) stands for TSUM operator, \( f(\sigma) \) is the value of function evaluated with \( X = \sigma \), \( \sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n) \), \( 0 \neq f(\sigma) \in E' = \{0, 1, \ldots, r-1\} \), \( x_i \in E' \), \( \sigma \subseteq E' \).

In other hand, this expression may be written as follows [1]:

\[
\tau_{r-1}^{-1} f(X) = \sum_{\sigma = 0}^{r^{-1}} f(\sigma) \Lambda(\bigcup_{i=1}^{n} \sigma_i)
\]

where the symbol \( \bigcup \) stands for binary NOR operator.

Expression (2) states that any \( r \)-valued function may be expanded as a truncated sum of binary NOR terms weighted by \( r \)-valued constants \( f(\sigma) \).

\textit{Definition 1.} Let \( X \) be the set of \( r \)-valued variables, \( \{X_1, X_2\} \) is a partition of \( X \) when \( X_1 \cap X_2 = \emptyset \) and \( X_1 \cup X_2 = X \).

To represent \( r \)-valued functions with minimal expressions, the following notations are adopted:

- \( E' \): the set of constants is an \( r \)-valued Rosser algebra, \( E' = \{0, 1, \ldots, r-1\} \)
- \( X \): the set of \( n \) \( r \)-valued variables \( \{x_1, x_2, \ldots, x_n\} \), where \( n \) is the number of elements of \( X \).
- \( X_j \): the bound set of variables;
- \( X_f \): the free set of variables;
- \( d(X) \): the number of elements of the set \( X \).
- \( \sigma \): \( (\sigma_1, \sigma_2, \ldots, \sigma_n) \), where \( \sigma_i \in E' = \{0, 1, \ldots, r-1\} \), \( i = 1, 2, \ldots, n \).
- \( \tau \): \( (\tau_1\tau_2\ldots\tau_n) \in \sigma \), where \( \tau_i \in E' = \{0, 1, \ldots, r-1\} \), \( i = 1, 2, \ldots, n_1 \), \( d(X_j) = n_1 \).
- \( \eta \): \( (\eta_1\eta_2\ldots\eta_n) \in \sigma \), where \( \eta_i \in E' = \{0, 1, \ldots, r-1\} \), \( i = 1, 2, \ldots, n_2 \), \( d(X_2) = n_2 \).

\textit{Definition 2.} Let \( f(X) \) be an \( r \)-valued function and \( \{X_1, X_2\} \) is a partition of \( X \). Then the projection of \( f(X) \) over \( X_j = \tau \), \( f(\tau, X_j) \) is the value of \( f(X) \) evaluated with \( X_j = \tau \).

\textit{Definition 3.} An \( r \)-valued function \( f(X) \) is said to have a \textit{generalized decomposition} with a bound set \( X_j \) and a free set \( X_2 \) if there exist \( r \)-valued functions \( h_1, h_2, \ldots, h_k \) and \( g \) such that

\[f(X) = g(h_1(X_1), h_2(X_1), \ldots, h_k(X_1), X_2)\]

where \( \{X_1, X_2\} \) is a partition of \( X \).

\[
\begin{align*}
\text{Fig. 1 Circuit diagram of } f(X) = g(h_1(X_1), h_2(X_1), \ldots, h_k(X_1), X_2). \\
(a) & \quad h_i \text{ is coded as } r \text{-valued variables, } \\
(b) & \quad h_i = f(\tau, X_i) \text{ is examined as the projections of } f(X) \text{ over } X_i = \tau.
\end{align*}
\]
3. The specific properties of TSUM operator

In this section we derive the specific properties of TSUM operator. Based on these features the MIN-TSUM and NOR-TSUM expansions with respect to a group of variables (Theorem 1 and Theorem 2) are obtained which will be used in the next section for designing the PLA-based combinational circuits.

Let \( x, y \) and \( z \) be \( r \)-valued logic variables. The following operations hold for multiple-valued algebra:

1. Associative laws: \((x \oplus y) \oplus z = x \oplus (y \oplus z)\)
2. Commutative laws: \(x \oplus y = y \oplus x\)
3. Identities:
   - 3A. \((r-1) \oplus x = r-1\)
   - 3B. \(0 \oplus x = x\)
   - 3C. \(x_0 \oplus x_1 \oplus \ldots \oplus x_{r-1} = r-1\)
   - 3D. \(x \land x' = 2 \land x^2 \oplus \ldots \oplus (r-1) \land x^{r-1} = x\)
4. Distributive laws:
   - 4A. \((x \land y) \land z = (x \land z) \land (y \land z)\)
   - 4B. \((x \land y) \lor z = (x \lor z) \land (y \lor z)\)
   - 4C. \((x \land y) \lor z = (x \land z') \lor (y \land z')\)

All specific properties of TSUM operator are proved in Appendix 1.

4. Decomposition of NOR-TSUM and MIN-TSUM expansions

In this section we will describe the new representations of NOR-TSUM and MIN-TSUM expansions over the bound set, \( X_1 \) of variables.

The \( Q_d(X) \) term is a binary function defined by the Boolean product of literals and taking values in the \( \{0, r-1\} \) set:

\[
Q_d(X) = x_1 \sigma_1 \cdot x_2 \sigma_2 \cdot \ldots \cdot x_n \sigma_n
\]

Theorem A. Let \( f(X) \) be an \( r \)-valued logic function and \( \{X_1, X_2\} \) be a partition of \( X \). Let \( n_1 = d(X_1) \) and \( n_2 = d(X_2) \). Then the MIN-TSUM expansion of \( f(X) \) over \( X_1 \) is given by

\[
f(X) = \sum_{\tau=0}^{r_1-1} \sum_{\eta=0}^{r_2-1} f(\sigma) \land Q_d(X_1) \land Q_d(X_2)
\]

where \( f(\sigma) = f(\tau, \eta) \) is the value of \( f(X) \) evaluated with \( X_1 = \tau \) and \( X_2 = \eta \), i.e. simple the value of \( f(X) \) for \( \sigma \), \( \sigma = (\sigma_1, \sigma_2, \ldots, \sigma_r) \), be the input combination of \( X \), \( \sigma = (\tau, \eta) \), \( Q_d(X) = Q_d(X_1) \land Q_d(X_2) \).

(Proof) Transform expression (1) using conjunction and truncated sum properties. By definition of \( Q_d(X) \) term, expression (1) can be represented as:

\[
f(X) = \sum_{\sigma=0}^{r^1-1} f(\sigma) \land Q_d(X)
\]

Because \( \{X_1, X_2\} \) is a partition of \( X \), the function \( Q_d(X) \) can be expressed as \( Q_d(X) = Q_d(X_1) \land Q_d(X_2) \), where \( \sigma = (\tau, \eta) \) is the input combination of \( X \) and \( \tau \) and \( \eta \) are the input combinations of \( X_1 \) and \( X_2 \), respectively. Therefore,

\[
f(X) = \sum_{\sigma=0}^{r^1-1} f(\sigma) \land Q_d(X_1) \land Q_d(X_2)
\]

Both conjunction and truncated sum are commutative (9), then

\[
f(X) = \sum_{\sigma=0}^{r^1-1} Q_d(X_1) \land f(\sigma) \land Q_d(X_2)
\]

Note, that \( Q_d(X_1) \) and \( Q_d(X_2) \) are the binary functions and \( f(\sigma) \) takes the value of 0, 1, ..., \( (r-1) \). Because \( \sigma = (\tau, \eta) \) and both conjunction and truncated sum are submitted to all distributive laws (14, (16), so

\[
f(X) = \sum_{\tau=0}^{r_1-1} \sum_{\eta=0}^{r_2-1} f(\tau, \eta) \land Q_d(X_1) \land Q_d(X_2)
\]
Hence, the theorem is proved.  \textit{Q.E.D.} \\
Note that the projection of \( f(X) \) over \( X_i \) for MIN-TSUM expansion is determined as follows:

\[
f(r_{n_i-1}) = \sum_{\eta=0}^{n_i} \sum_{j=1}^{r_{n_i-1}} f(\tau \eta) \Lambda(\bigcup_{x_i=1}^{n_i} \tau_j) \Lambda(\bigcup_{x_i=0}^{n_i} \eta_j).
\]

\textbf{Example A}. Consider a 3-valued 3 variable logic function \( f(x_1, x_2, x_3) \) \((r=3, n=3)\), shown in Table 1. Then the MIN-TSUM expansion with respect to \( x_1, x_2 \) variables for this function is given as

\[
\begin{array}{cccccccccc}
x_1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\
x_2 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 2 \\
x_3 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 2 & 1 \\
\end{array}
\]

\textbf{Table 1}. Decomposition table for \( f(x_1, x_2, x_3) \)

\[
f(X) = x_1 x_2 (2 - x_3 0 \oplus 2 - x_3 1 \oplus 1 - x_3 2) x_1 x_2 (1 - x_3 0 \oplus 1 - x_3 1 \oplus 1 - x_3 2) x_1 x_2 (0 - x_3 0 \oplus 0 - x_3 1 \oplus 2 - x_3 2) x_1 x_2 (2 - x_3 0 \oplus 1 - x_3 1 \oplus 0 - x_3 2) x_1 x_2 (0 - x_3 0 \oplus 0 - x_3 1 \oplus 1 - x_3 2) x_1 x_2 (1 - x_3 0 \oplus 0 - x_3 1 \oplus 1 - x_3 2) x_1 x_2 (2 - x_3 0 \oplus 0 - x_3 1 \oplus 1 - x_3 2) x_1 x_2 (0 - x_3 0 \oplus 2 - x_3 1 \oplus 0 - x_3 2).
\]

\textit{(End of example)}

\textbf{Theorem B}. Let \( f(X) \) be an \( r \)-valued logic function and \( \{X_i, X_j\} \) be a partition of \( X \). Let \( n_i = d(X_i) \) and \( n_j = d(X_j) \).

Then the expansion of \( f(X) \) over \( X_j \) for NOR-TSUM expression is given by

\[
f(X) = \sum_{\tau=0}^{r_{n_j-1}} \sum_{i=1}^{r_{n_i-1}} f(\tau \eta) \Lambda(\bigcup_{x_i=1}^{n_i} \tau_j) \Lambda(\bigcup_{x_i=0}^{n_i} \eta_j).
\]

where the symbol \( \lor \) stands for binary NOR operator.

\textit{(Proof)} By theorem a any \( r \)-valued function can be represented as follows:

\[
f(X) = \sum_{\tau=0}^{r_{n_j-1}} \sum_{i=1}^{r_{n_i-1}} Q_d(X_i) \sum_{\tau=0}^{r_{n_j-1}} \sum_{i=1}^{r_{n_i-1}} f(\tau \eta) \Lambda(\bigcup_{x_i=1}^{n_i} \tau_j) \Lambda(\bigcup_{x_i=0}^{n_i} \eta_j) = \sum_{i=1}^{r_{n_j-1}} \sum_{i=1}^{r_{n_i-1}} Q_d(X_i) \sum_{i=1}^{r_{n_i-1}} f(\tau \eta) \Lambda(\bigcup_{x_i=1}^{n_i} \tau_j) \Lambda(\bigcup_{x_i=0}^{n_i} \eta_j).
\]

Each term in previous expression may be converted as follows [1]

\[
Q_d(X_i) = x_1 x_2 x_3 \ldots x_{n_i} \tau_i \biglor \bar{x}_1 \biglor \bar{x}_2 \biglor \bar{x}_3 \ldots \biglor \bar{x}_{n_i} \tau_i = \biglor_{i=1}^{n_i} x_i \tau_i.
\]

Taking into account the expression (7) we can make the substitution in (6). Hence we have a theorem.  \textit{Q.E.D.} \\

Note that the projection of \( f(X) \) over \( X_i \) for NOR-TSUM expansion is determined as follows:

\[
f(r_{n_i-1}) = \sum_{\eta=0}^{n_i} \sum_{i=1}^{r_{n_i-1}} f(\tau \eta) \Lambda(\bigcup_{x_i=1}^{n_i} \tau_j) \Lambda(\bigcup_{x_i=0}^{n_i} \eta_j).
\]

\textbf{Example B}. The NOR-TSUM expansion with respect to \( x_1, x_2 \) variables for function described in example a is offered as

\[
f(X) = x_1 x_2 \Lambda(0 - x_3 0 \oplus 2 - x_3 1 \oplus 0 - x_3 2) x_1 x_2 \Lambda(1 - x_3 0 \oplus 0 - x_3 1 \oplus 2 - x_3 2) x_1 x_2 \Lambda(0 - x_3 0 \oplus 1 - x_3 1 \oplus 0 - x_3 2) x_1 x_2 \Lambda(1 - x_3 0 \oplus 0 - x_3 1 \oplus 1 - x_3 2) x_1 x_2 \Lambda(0 - x_3 0 \oplus 2 - x_3 1 \oplus 0 - x_3 2) x_1 x_2 \Lambda(0 - x_3 0 \oplus 2 - x_3 1 \oplus 0 - x_3 2)
\]

\textit{(End of example)}

\section{Design method for MV PLA-based combinational circuits}

In this section we present a design method for MV PLA based combinational circuits using decomposition of NOR-TSUM and MIN-TSUM expansions.

This novel method incorporates the concept of systematically reexpressing the given MVL function. In our approach, we do not attempt to rewrite the function in terms of functions which are then used as new variables in the logic equation. Rather, we identify the minimum number of projections of \( f(X) \) over \( X_i \) and reexpressed the given function by obtained partition of X. Each of function projection is then utilized as \( r \)-valued constant in the logic equation. The function is reexpressed recursively until it is represented entirely by the function projections. In this case the partition of \( X \) composes \( p \) subsets of variables \( \{X_i, X_2, \ldots, X_p\} \), where \( p \) is the number of function reexpressions. In this paper we consider only partition of \( X \) containing two subsets. Therefore, function \( f(X) \) is reexpressed only once (Theorem 1 and Theorem 2).
The \( r \)-valued combinational circuit composes of two level. On the first level the projections of \( f(X) \) over \( X_1 \) are implemented by MV PLA. Let us consider the MV PLA proposed in [1], and called NOR-TSUM PLA. In this case the number of inputs corresponds to the number of variables in the free set, \( X_2 \). The number of outputs is determined by the number of different projections in the NOR-TSUM expansion. On the second level of combinational circuit the function projections over \( X_1 \) are identified as \( r \)-valued constants.

Note that instead of MV PLA we can use any kind of universal logic module such as T-gate (multiplexer), programmable logic array (PAL), look-up table type FPGA (field programmable gate array) and etc.

**Example 3.** The decomposition table of \( f(x_1, x_2, x_3, x_4, x_5) \) is shown in Fig. 2(b). A 3 variable MV PLA is assumed to be the component used to implement the function. Since the MV PLA uses not more than 3 variables, only 3 variables are used to partition function \( f(X) \). No attempt is made to try 4 variables in an attempt to decompose the function.

Selecting \( x_1 \) and \( x_2 \) as the variables of the free set, \( X_2 \), the function \( f(X) \) is rewritten as a 3 variable function and its functional values become the projections of \( f(X) \) over \( X_1 \). Then, there are 5 different projections of \( f(X) \) over \( X_1=\pi \), that is the number of product terms in the second MV PLA. The \( r \)-valued combinational circuit implementing \( f(X) \) function is shown in Fig. 2(b).

### Table 2

<table>
<thead>
<tr>
<th>Function</th>
<th>PLA by method [1]</th>
<th>Combinational circuit by the proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \text{In} )</td>
<td>( \text{Out} )</td>
</tr>
<tr>
<td>( X+Y )</td>
<td>( \text{In} )</td>
<td>( \text{Out} )</td>
</tr>
</tbody>
</table>

Fig. 2 The structure of PLA-based combinational circuit (a) common case, (b) for function \( f(x_1, x_2, x_3, x_4, x_5) \).
6. Experimental results

Table 3 compares the size of synthesized circuits with the size of MV PLA implemented the initial function. This table shows that the present method produces the competitive results with other methods. Because the proposed decomposition approach permits only one \(r\)-valued function to be decomposed the functions from the standardized benchmarks are separately implemented. Here such benchmarks as 2-figure quaternary adder, 2-figure quaternary multiplier, 3 digit shift register are used for experimental investigations.

7. Conclusion

A new technique for realizing PLA-based combinational circuits has been proposed. This technique yields multiple-valued circuit of reduced size. This has been illustrated by considering the realization of functions from standardized benchmarks. Within suggested approach the novel MIN-TSUM and NOR-TSUM expansions with respect to \(X_1\) are derived. The specific properties of TSUM operator are investigated too. It is shown that truncated sum of \(r\)-valued variables does not submit one of the distributive laws. It needs to take into account the consideration at examining the MIN-TSUM and NOR-TSUM expansions.

Future work is carried out in frame of generalization the proposed approach for system of MVL functions. It leads to investigate the performance of new approach on other standardized benchmark problem.

8. Acknowledgment

The authors sincerely thank Dr. V. Shmerko from Technical University (Szczecin, Poland) for his helpful comments and suggestions on this paper.

Appendix 1

1. Associative laws: (8) \((x \oplus y) \oplus z = x \oplus (y \oplus z)\)

\((\text{Proof})\) \((x \oplus y) \oplus z = x \oplus (y \oplus z) = x \oplus (y \oplus z) = x \oplus (y \oplus z)\) because \(r-1+z \geq r-1\) for all values of \(z\).

In other hand, \((x \oplus y) \oplus z = x \oplus (y \oplus z) = x \oplus (y \oplus z)\) because \(r-1+x \geq r-1\) for all values of \(x\).

Since expressions \((a)\) and \((b)\) are equivalent, then TSUM operator is associative. \(Q.E.D.\)

2. Commutative laws: (9) \(x \oplus y = y \oplus x\)

\((\text{Proof})\) This expression is proved using the definition of TSUM operator.

3. Identities: (10) \((r-1) \oplus x = r-1\)

\((\text{Proof})\) \((r-1) \oplus x = MIN(r-1+x, r-1) = r-1\), because \(r-1+x \geq r-1\) for all values of \(x\). \(Q.E.D.\)

\((11)\) \(0 \oplus x = x\)

\((\text{Proof})\) \(0 \oplus x = MIN(0+x, r-1) = x\). \(Q.E.D.\)

\((12)\) \(x^0 \oplus x^1 \oplus \ldots x^{r-1} = r-1\)

\((\text{Proof})\) This expression describes all input combination of \(x\), therefore by definition of literal function Eq. (12) takes value \((r-1)\). \(Q.E.D.\)

\((13)\) \(I \Lambda x^I \oplus 2 \Lambda x^2 \oplus \ldots \oplus \Lambda (r-1) \Lambda x^{r-1} = x\), where the symbol \(\Lambda\) stands for \(r\)-valued minimum.
(Proof) \( x=0 \land x'' \lor I \land x' \lor 2 \land x' \lor \ldots \lor (r-I) \land x'' = I \land x' \lor 2 \land x' \lor \ldots \lor (r-I) \land x'' \), because \( 0 \land a' = 0 \).

Q.E.D.

4. Distributive laws.

(14) \((x \land y) \lor z = (x \lor z) / (y \lor z)\), where \(x, y, z \in \{0, 1, ..., r-1\}\)

(Proof)

a) \( x \land y \lor z = \text{MIN} (\text{MIN} (x, y) + z, r-1) = \text{MIN} (x+z, y+z, r-1) \)

b) \( (x \lor z) / (y \lor z) = \text{MIN} (x+z, r-1) \land \text{MIN} (y+z, r-1) = \text{MIN} (\text{MIN}(x+z, r-1), \text{MIN}(y+z, r-1)) = \text{MIN} (x+z, y+z, r-1) \)

Since expressions \((a)\) and \((b)\) are equivalent, then \(TSUM\) operator submit to the distributive law.

Q.E.D.

(15) \((x \lor y) \land z \neq (x \land z) \lor (y \land z)\)

(Proof)

a) \( (x \lor y) \land z = \text{MIN} (x+y, r-1) \land z = \text{MIN}(\text{MIN}(x+y, r-1), z) = \text{MIN} (x+y, z, r-1) \)

b) \( (x \land z) \lor (y \land z) = \text{MIN} (x, z) \lor \text{MIN} (y, z) = \text{MIN}(\text{MIN}(x, z) + \text{MIN}(y, z), r-1) = \text{MIN}(x+y, z+x, 2z, y+z, r-1) \)

Because the expressions \((a)\) and \((b)\) are not equivalent, \(TSUM\) operator does not submit to the distributive laws.

Q.E.D.

(16) \((x \lor y) \land z' = (x \land z') \lor (y \land z')\)

(Proof)

a) \( (x \lor y) \land z' = \text{MIN} (x+y, r-1) \land z' = \text{MIN}(\text{MIN}(x+y, r-1), z') = \text{MIN} (x+y, z', r-1) \)

b) \( (x \land z') \lor (y \land z') = \text{MIN}(x, z') \lor \text{MIN}(y, z') = \text{MIN}(\text{MIN}(x, z') + \text{MIN}(y, z'), r-1) = \text{MIN}(x+y, z'+x, 2z', y+z', r-1) \)

Because \(z' \in \{0, r-1\}\), then

\[
x+z' = \begin{cases} x, & \text{если } z=0, \\ x+r-1, & \text{otherwise.} \end{cases}
\]

But \(x+r-1 \geq r-1\), Eq. \((b)\) takes into account variable \(z'\), therefore \(x+z'\) and \(z'\) do not examine.

The expression \(y+z'\) is analyzed by analogy with previous expression.

(c) \( \text{MIN}(x+y, z'+x, 2z', y+z', r-1) = \text{MIN}(x+y, z, r-1) \)

Since expressions \((a)\) and \((c)\) are equivalent, then \(TSUM\) operator submits to the distributive law.

Q.E.D.

References