Optimal assignment of multi-valued and binary nodes in heterogeneous DDs

Suzana Stojković[1] and Radomir Stanković

Abstract – **Decision diagrams (DDs) are a data structure for representation and manipulation of discrete logic functions efficiently in terms of space and time. Binary Decision Diagrams (BDDs) are the most widely used for representation of Boolean functions. Complexity of BDD representations is usually estimated through the number of nodes in BDDs, usually denoted as the size of BDD. Minimization of size of BDDs is a greatly considered problem in the literature. One of the solutions is the usage of multi-valued and heterogeneous DDs for representing Boolean functions. In this paper, an approach for determination of optimal assignment of multi-valued and binary nodes in heterogeneous DDs is proposed. The paper presents experimental results that verify usefulness of the proposed method.**

Keywords – **Boolean functions, Decision diagrams, Heterogeneous decision diagrams, Minimization of size of decision diagram**

I. INTRODUCTION

Decision diagrams (DDs) are graph-based data structures for representation of discrete functions permitting to represent functions of large number of variables efficiently in terms of space and time [11]. For representing Boolean functions, Binary Decision Diagrams (BDDs) are widely used [2]. Compactness of BDD representations is usually estimated through the number of nodes in the BDDs denoted as the size of BDD. The size of BDDs depends on the order of variables in functions represented by BDDs. Minimization of the size of BDD by reordering of variables is a widely discussed problem, see for example [4], [5], [6], [7] and references therein. Majority of the proposed algorithms for BDD minimization are heuristic and express common disadvantages of heuristic algorithms that does not garantee the quality of solutions provided. Another approach to BDD minimization is by using linear transformations of input variables [5], [6]. Third way for minimization of DD representations of Boolean functions is by using multi-valued DDs (MDDs) [10]. MDDs, in general case, represent multi-valued functions $f:\{0,1,...,q-1\}^n \rightarrow \{0,1,...,q-1\}$. Boolean functions are represented by MDDs by encoding subsets of binary variables by a single multiple-valued variables.

Recently, heterogeneous MDDs are used for Boolean function representation [7], [8], [9]. In heterogeneous DDs, nodes at different levels can have different number of outgoing edges. It is shown that heterogeneous DDs are useful for representation of Boolean functions when priority is: minimization of size and cost of DDs expressed as the number of nodes and cost of nodes, minimization of the average path length in DDs, etc.

In this paper, we propose a method to determine an optimal assignment of multi-valued and binary nodes in heterogeneous DDs when the size of DD should be minimized.

[1] The authors are with Dept. of Computer Science, Faculty of Electronic Engeneering, Beogradska 14, 18000 Niš, Serbia and Montenegro, e-mail: suza@elfak.ni.ac.yu.

II. MINIMIZATION OF DD SIZE

A. Minimization of DD by reordering of variables

Size of DD is defined as the number of non-terminal nodes in the DD. The size of a DD depends on the order of variables assigned to the levels of the DD.

Example 1. 1Fig. 1 and shows BDDs for a function $f(x_1,x_2,x_3)=x_1x_3+x_2$ for the order of variables (*a*) (x_1,x_2,x_3), and (*b*) (x_2,x_1,x_3). These BDDs have 4 and 3 non-terminal nodes, respectively. Thus, in this example, permutation of variables x_1 and x_2 permits reduction of a non-terminal node.

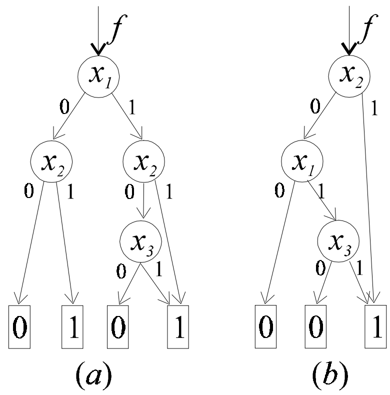

Fig. 1 BDDs for the function f in Example 1.

Exact algorithms for optimal order of variables reuce to brute search methods. Heuristic algorithms are based on dynamic reordering and sifting of variables [3].

B. DD Minimization by using of MDDs

Boolean function of $n*k$ binary variables can be alternatively viewed as a binary-valued function of n 2^k-valued variables. In this approach, a subset of k binary-valued variables is replaced by a single 2^k-valued variable. Due to this encoding, Boolean functions can be represented by MDDs.

Example 2. Fig. 2 shows that a Boolean function

$$f(x_1,x_2,x_3,x_4) = x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1$$

can be represented by BDD with 6 non-terminal nodes and or by a 4-valued MDD with four non-terminal nodes.

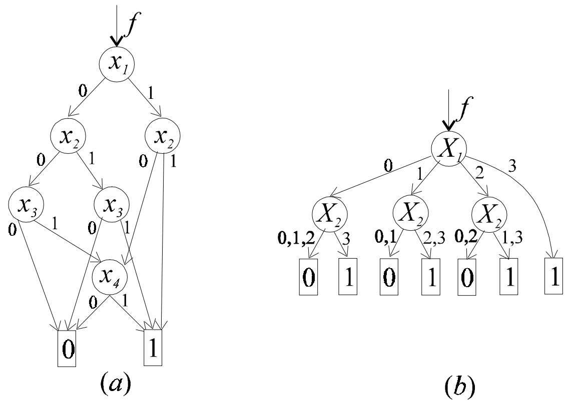

Fig. 2 BDD and 4-valued MDD for the function *f* in Example 2.

In estimation of the const of a DD, it is usually assumed that the cost of a node is proportional to the number of outgoing edges. In this setting, the cost of a multi-valued node (the node with *q* outgoing edges) is greater then the cost of a binary-valued node. Due to that, the usage of MDDs is not always efficient. If we assume that the cost of a non-terminal node is *K*number of edges*, where *K* is the size of the DD, the cost of the BDD in Fig. 2 (*a*) is a *12K*, but the cost of MDD in Fig. 2 (*b*) is *16K*. It follows that usage of MDD, in that case, is not justified. Another disadvantage of representation of Boolean functions by MDDs is the requirement for the relationship between the number of binary variables and the number of outgoing edges of multiple-valued nodes that must be satisfied. In particular, MDDs with 2^k-valued variables can be applied for representation of Boolean functions with $n*k$ binary variables.

III. Heterogeneous DDs

When a Boolean function is represented by a MDD, the set of input variables is partitioned into subsets of the same cardinality. In heterogeneous DDs, the set of input variables is partitioned in subsets of different cardinality. Therefore, in heterogeneous DDs non-terminal nodes at different levels can have different number of outgoing edges.

Example 3. Fig. 3 shows three heterogeneous DDs representing the function f in Example when the set of input variables (x_1,x_2,x_3,x_4) is partitioned into subsets (X_1,X_2,X_3), where:

(a) $X_1 = (x_1)$, $X_2 = (x_2)$ and $X_3 = (x_3,x_4)$.
(b) $X_1 = (x_1)$, $X_2 = (x_2,x_3)$ and $X_3 = (x_4)$.
(c) $X_1 = (x_1,x_2)$, $X_2 = (x_3)$ and $X_3 = (x_4)$.

Size of the first DD is 6 and number of its 4-valued non-terminal nodes is 3 (its total cost is *18K*). Size of the second DD is 4 and number of its non-terminal 4-valued non-terminal nodes is 2 (its total cost is *8K*). Size of third DD is also 4, but the number of 4-valued non-terminal nodes is 1 (its total cost is *6K*). It shows that heterogeneous DD shown in Fig. 3 (*c*) is the most efficient for representing the considered Boolean function.

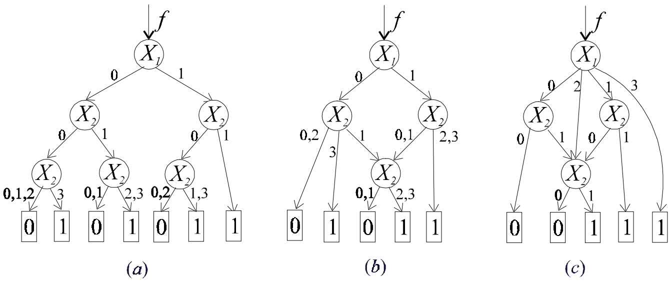

Fig. 3 Three different heterogeneous DDs for f in Example 3.

From the considerations in Example, the following question arises. Which assignment of multi-valued and binary-valued nodes produces a heterogeneous DD with the smallest cost for a given Boolean function *f*? When a subset of binary-valued variables $(x_j,\ldots x_{j+k-1})$ is replaced by a 2^k-valued variable, nodes at the levels *j*+1,...,*j*+*k*-1 are deleted from DD. For determination of subsets of variables which should be replaced by multi-valued variables, two criteria are imposed:

- Replacement is reasonable when the number of new nodes (that must be created at the level *j*) is smaller than the number of deleted nodes.
- Heterogeneous DD will be of the minimum size if levels with the maximal number of non-terminal nodes will be deleted.

IV. Maximal possible size of levels in BDD

Consider an *n*-variable Boolean function *f*. Denote the number of non-terminal nodes at the level *k* in a BDD by *width*(BDD,*k*). Then, *width*(BDD,*k*) is limited by two ways:

1. *width*(BDD,*k*)≤*width*(BDT,*k*), where BDT denotes a Binary Decision Tree from which BDD is derived by the application of the BDD reduction rules [11]. Recall that the number of non-terminal nodes at the *k*-th level in the BDT is equal to 2^{k-1}. It follows that:

$$width(\mathrm{BDD},k) \leq 2^{k-1} \quad (1)$$

i.e.,

$$max\_wl(\mathrm{BDD},k) = 2^{k-1} \quad (2)$$

2. *width*(BDD,*k*) is also limited by the number of possible successors. A successor of a node at the level *k* can be any node from the levels *k*+1, ... , *n*, *n*+1. Number of possible nodes at the level *n*+1 is 2 (these are terminal nodes 0 and 1). Number of possible nodes at the levels *n*, *n*-1,...,1 can be calculated by:

$$width(\text{BDD},k) \leq \left(\sum_{i=k+1}^{n+1} width(\text{BDD},i) \right)^2 - \sum_{i=k+1}^{n+1} width(\text{BDD},i) \quad (3)$$

i. e.,

$$maxw2(\text{BDD},k) = \left(\sum_{i=k+1}^{n+1} maxw2(\text{BDD},i) \right)^2 - \sum_{i=k+1}^{n+1} maxw2(\text{BDD},i)$$

$$= 2^{2^{n-k+1}} - 2^{2^{n-k}} \quad (4)$$

Generally:

$$max\_width(\text{BDD},k) = \mathbf{min(}maxw1(\text{BDD},k)\mathbf{,}maxw2(\text{BDD},k)\mathbf{)}$$

$$= \min(2^{k-1}, 2^{2^{n-k+1}} - 2^{2^{n-k}}) \quad (5)$$

It follows that the level with the maximum possible size in a BDD (denoted by $L_{\max}$) is:

$$L_{\max} = n - k \text{, for } n \in \left[2^k + k + 1, 2^{k+1} + k + 1\right]. \quad (6)$$

Values of $L_{\max}$ for different number of variables are shown in Table 1. In this table, $[i,j]$ shows the values $i=2^k+k+1$ and $j=2^{k+1}+k+1$ determined in (6).

TABLE 1
VALUES OF L_{MAX} FOR DIFFERENT VARIABLES NUMBER

N	[2,3]	[4,6]	[7,11]	[12,20]	[21,37]	[38,70]	[71,135]
$L_{\max}$	n	n-1	n-2	n-3	n-4	n-5	n-6

$max\_width(\text{BDD},k)$ for the levels less than $L_{\max}$ is equal to $maxw1(\text{BDD},k)$ and for the levels great then $L_{\max}$ is equal to $maxw2(\text{BDD},k)$.

V. DETERMINATION OF OPTIMAL ASSIGNMENT OF MULTI-VALUED AND BINARY NODES

In a BDD, multi-valued variables are introduced by attempting to replace the maximum possible number of binary nodes by the minimum number of multi-valued nodes. Due to that, the level $L_{\max}$ has to be deleted obligatory. If 4-valued variables are used, one partition of input variables must be $(x_{L\max-1}, x_{L\max})$. If one multi-valued variable replaces more than two binary variables, many partitions of the set of input variables containing $x_{L\max}$ can be defined. DD will be of the minimum size when the partition $(x_{L\max-k+1},\ldots,x_{L\max})$ is replaced by one 2^k-valued variable, because $max\_width(\text{BDD},k)$ quickly decreases from the level $L_{\max}$ to the terminal level, then from the level $L_{\max}$ to the root level.

If in a given function f the set of binary variables $(x_j,\ldots x_{j+k-1})$ is replaced by a 2^k-valued variable, then $maxw1(\text{DD},j)$, could not be changed (it is equal to 2^{j-1} independently of the cardinalities of other partitions of the set of input variables). In that case, $maxw2(\text{DD},j)$ would be:

$$maxw2(\text{DD},j) = \left(\sum_{i=j+k}^{n+1} maxw2(\text{BDD},i) \right)^{2^k} - \sum_{i=j+k}^{n+1} maxw2(\text{BDD},i)$$

$$= 2^{2^{n-j+1}} - 2^{2^{n-j-k+1}} = \sum_{i=j}^{j+k-1} maxw2(\text{BDD},i)\,. \quad (7)$$

Equation (7) shows that after replacement of binary nodes by multiple-valued nodes, the maximum number of 2^k-valued nodes at the level j (for j great then $L_{\max}$) is equal to the sum of maximum number of binary nodes at all levels which are changed. It follows that by replacing binary variables from levels greater than $L_{\max}$ by multi-valued nodes is not reasonable. It follows that, when multi-valued nodes are introduced, partitions of the set of input variables should be done by starting from the level $L_{\max}$ to the root node.

VI. EXPERIMENTAL RESULTS

Theoretical considerations, presented in sections IV and V were verified by experimental results. We performed two groups of experiments. In the first, we represented 100 randomly generated functions of 10 variables by different heterogeneous DDs. It is assumed that a function can take the value 0 or 1 with equal probability. **TABLE 3** shows average sizes and average numbers of multi-valued nodes (MVNN) in DDs with the specified assignment of nodes. In the second group, we performed experiments on representing functions of four variables. TABLE 2 shows numbers of functions of defined size for different types of heterogeneous DDs.

In the first, in both groups of experiments, heterogeneous DDs with one multi-valued variable were generated. Rows 2-10 of **TABLE 3** contain data about heterogeneous DDs of functions of 10 variables with one 4-valued node. In that group, DD of minimal size is the DD in which multi-valued variable replaces pair of input variables (x_7,x_8), i. e. $(x_{L\max-1},x_{L\max})$ because $L_{\max}=8$ for $n=10$. The same proposition is verified by comparison of sizes of DDs with one 8-valued non-terminal node. Data about those DDs are shown in rows 11-18. Minimal size in that group has the DD where the 8-valued variable replaces input variables (x_6,x_7,x_8).

In BDDs with four levels, $L_{\max}=3$. It follows that the optimal partition of input variables when 4-valued nodes are used in heterogeneous DDs is: $X_1=(x_1)$, $X_2=(x_2,x_3)$, $X_3=(x_4)$; and when 8-valued nodes are used: $X_1=(x_1,x_2,x_3)$, $X_2=(x_4)$. TABLE 2 shows that for these partitions the size of DDs is minimum.

Then, we have generated heterogeneous DDs with maximal number of multi-valued non-terminal nodes. We performed experiments by using 4-valued and 8-valued non-terminal nodes. The conclusion is that partitioning should be done from the level $L_{\max}$ to the root level. The optimal partitions of the set of 10 input variables when 4-valued 8-valued variables are used are, respectively:

$X_1=(x_1,x_2)$, $X_2=(x_3,x_4)$, $X_3=(x_5,x_6)$, $X_4=(x_7,x_8)$, $X_5=(x_9)$, $X_6=(x_{10})$;

and

$X_1=(x_1)$, $X_2=(x_2)$, $X_3=(x_3,x_4,x_5)$, $X_4=(x_6,x_7,x_8)$, $X_5=(x_9)$, $X_6=(x_{10})$.

Rows 19 and 20 show that the homogeneous 4-valued MDD and heterogeneous DD with induced optimal assignment of multi-valued and binary nodes have equal sizes. Heterogeneous DD is a better solution in that case, because the cost of binary nodes is smaller. Proposition that changing binary variables from levels greater than $L_{\max}$ by multi-valued variables has no effect on the size of the DD is confirmed by comparison of sizes of BDD and heterogeneous DD with one

TABLE 2

NUMBERS OF FUNCTIONS OF 4 VARIABLES REQUIRING DDS OF THE SPECIFIED SIZES

nodes arrangement \ size	0	1	2	3	4	5	6	7	8	9
2-2-2-2	2	8	48	236	960	3248	8928	17666	23280	11160
2-2-4	2	18	120	758	3304	11102	26208	24024		
2-4-2	2	18	192	1118	12784	51422				
4-2-2	2	18	264	2054	8056	18542	25440	11160		
2-8	2	256	1016	64262						
8-2	2	256	12608	52670						

4-valued variable (by replacing variables x_9 and x_{10}) (see rows 1 and 2 in **TABLE 3**).

At the end, sizes of determinated optimal heterogeneous DD with 8-valued variables was compared with sizes of heterogeneous DDs with maximum possible number of 8-valued variables, i.e., with DDs having three 8-valued variables and one binary variable (see rows 21-25 in **TABLE 3**). It was verified that theoretically determined optimal solution is really optimal.

TABLE 3

AVERAGE SIZES OF DIFFERENT HETEROGENEOUS DDS REPRESENTING FUNCTIONS OF 10 BINARY VARIABLES

No.	Nodes arrangement	DD size	MVNN
1.	2-2-2-2-2-2-2-2-2-2	235.49	
2.	2-2-2-2-2-2-2-2-4	235.49	14
3.	2-2-2-2-2-2-2-4-2	228.09	99.26
4.	2-2-2-2-2-2-4-2-2	140.96	63.96
5.	2-2-2-2-2-4-2-2-2	171.66	32
6.	2-2-2-2-4-2-2-2-2	203.49	16
7.	2-2-2-4-2-2-2-2-2	219.49	8
8.	2-2-4-2-2-2-2-2-2	227.49	4
9.	2-4-2-2-2-2-2-2-2	231.49	2
10.	4-2-2-2-2-2-2-2-2	233.49	1
11.	2-2-2-2-2-2-2-8	226.84	100.01
12.	2-2-2-2-2-2-8-2	128.97	63.97
13.	2-2-2-2-2-8-2-2	77	32
14.	2-2-2-2-8-2-2-2	139.66	16
15.	2-2-2-8-2-2-2-2	187.49	8
16.	2-2-8-2-2-2-2-2	211.49	4
17.	2-8-2-2-2-2-2-2	223.49	2
18.	8-2-2-2-2-2-2-2	229.49	1
19.	4-4-4-4-4	98.96	98.96
20.	4-4-4-4-2-2	98.96	84.96
21.	2-8-8-8	119.01	118.01
22.	8-2-8-8	125.01	117.01
23.	8-8-2-8	172.84	109.01
24.	8-8-8-2	74.97	72.97
25.	2-2-8-8-2-2	53	36

VII. CONCLUSION

In this paper, we presented principles for determination of the optimal assignments of multi-valued and binary nodes in heterogeneous DDs. This principle proposes that partition of the set of binary-valued input variables into subsets of variables (that will be replaced by multi-valued variables) has to be done by starting from the level with the maximum possible number of nodes to the root level. A formula for computation of the level with maximal possible number of non-terminal nodes in BDD is determined.

Experimental results that verify all the presented theoretical considerations are provided.

REFERENCES

[1] K. S. Brace, R. L. Rudell, R. E. Bryant, "Efficient implementation of a BDD package", In Design Automation Conference, pp. 417-421, San Francisko, Juny 1991.

[2] R. E. Bryant, "Graph-based algorithms for Boolean functions manipulation", IEEE Trans. on Computers, Vol. C-35, No. 8, 677-691, August 1986.

[3] R. E. Bryant, "Binary Decision Diagrams and Beyond: Enabling Tehnologies for Formal Verification", International Conference on Computer-Aided Design ICCAD '95, pp. 236-243, November, 1995.

[4] S.W. Jeong, T.S. Kim, and F. Somenzi, "An efficient method for optimal BDD ordering computation" In *International Conference on VLSI and CAD (ICVC'93)*, Taejon, Korea, November 1993.,

[5] W. Gunther, R. Drechler, "BDD Minimization by Linear Transformations", International Workshor on Boolean Problems, Freiberg, 1998.

[6] W. Gunther, R. Drechler, "Linear Transformations and Exact Minimization of BDDs", IEEE Great Likes Symposium on VLSI, pp. 325-330, Lafayette, 1998.

[7] S. Nagayama, T. Sasao, "Minimization of Memory Size for Hetrogeneous MDDs", ASP-DAC 2004 (Asia and South Pacific Design Automation Conference 2004) , pp. 872-875, Yokohama, Jan. 27-30, 2004.

[8] S. Nagayama, T. Sasao, "Compact Representation of Logic Functions using Hetrogeneous MDDs", 33rd International Symposium on Multiple-Valued Logic, pp. 247-255, Tokyo, May 16-19, 2003.

[9] S. Nagayama, T. Sasao, "Compact Representation of Logic Functions using Hetrogeneous MDDs", IEICE Transactions on Fundamentals of Electronics, Vol. E86-A, No. 12, pp. 3168-3175, Dec. 2003.

[10] T. Sasao, J. T. Butler, "A design Method for Look-up Table Type FPGA by Pseudo-Kronecker Expansion", Proceedings of 24-th International Symposium on Multiple-Valued logic, pp. 97-106, Boston, MA, May 25-27, 1994.

[11] T. Sasao, M. Fujita, Representatiosn of Discrete Functions, Kluwer Academic Publishers, 1995.