On Efficient Dynamic Reordering of Variables for Binary Decision Diagrams*

(Preliminary Version)

Paul Molitor †

Abstract

We present some new ideas for dynamic reordering of variables for ordered binary decision diagrams (BDD). The new method uses the degree of symmetry between two variables and the conditional informational contribution of variables in order to efficiently find variable orders minimizing the size of the BDDs.

Remark

As already mentioned in the preface of this anniversary volume, this paper is dedicated to Prof. Dr. Karl-Heinz Rauchhaus whose 60th birthday took place some weeks ago. As I would never have believed that this round birthday is celebrated already this year – Karl-Heinz does not look his age – we have not yet finished our work. This is the reason the paper closes with remarks on future work necessary to reach our goal, namely the development of an efficient dynamic variable reordering heuristic which does not construct temporary BDDs. This future work will be joint work with Nicole Göckel and Bernd Becker, both from Freiburg University, and Laura Litan, Halle University.

*This work was supported in part by DFG grant Mo 645/2-1.

†email: `molitor@informatik.uni-halle.de`

1 Introduction

Binary Decision Diagrams (BDD) as a data structure for representation of Boolean functions were first introduced by Lee (1959) [11] and further popularized by Akers (1978) [1] and Moret (1982) [16]. In the restricted form of reduced ordered BDDs (ROBDDs), where the variables are visited in the same order on any path from a root to a leaf in the diagram, they gained widespread application because ROBDDs are a canonical representation for a fixed variable order and allow further efficient manipulations [3]. Some fields of application are logic verification, test generation, fault simulation, and logic synthesis [13, 4].

Most of the algorithms using ROBDDs have running time polynomial in the size of the ROBDDs. In general, the sizes strongly depend on the variable order used. Thus, there is a need for efficient algorithms computing efficient variable orders. As determining the optimal variable order is NP-complete [3], we have to attack the problem by various heuristic approaches.

The existing heuristic methods for finding good variable orders can be classified into two categories: initial heuristics which derive an order by inspection of a logic circuit [13, 7, 8] and dynamic reordering heuristics which try to improve on a given order [9, 19, 6, 2, 5]. Sifting introduced by Rudell [19] has emerged so far as the most successful algorithm for dynamic reordering of variables. This algorithm is based on finding the optimum position of a variable assuming all other variables remain fixed. The position of a variable in the order is determined by moving the variable to all possible positions while keeping the other variables fixed. However, this approach to dynamical reordering has two drawbacks.

As already observed by Panda and Somenzi [17], one limitation of sifting is that it uses the absolute position of a variable as the primary objective and only considers the relative positions of groups of variables indirectly. Recently, it has been shown by Möller et.al. [15] and Panda et.al. [18] that symmetry properties can be used to efficiently construct good variable orders for ROBDDs using modified gradual improvement heuristics. The crucial point is to locate the partial symmetric variables side by side and to treat them as fixed block. This results in 'symmetric sifting' which sifts the variables of a symmetric group simultaneously. Regular sifting usually puts symmetric variables together in the order, but the symmetric groups tend to be in suboptimal positions. The suboptimal solutions result from the fact that regular sifting is unable to recognize that the variables of a symmetric group have a strong attraction to each other and should be shifted together. When a variable of a symmetric group is sifted by regular sifting, it is likely to return to its initial position due to the attraction of the other variables of the group.

The other drawback is the fact that both regular sifting and symmetric sifting construct as many as n^2 different ROBDDs as, in every step, the costs of the corresponding ROBDDs, i.e., the total number of inner nodes, have to be computed. However, ROBDD construction can often be rather time consuming.

In this paper we present some new ideas in order to attack the second drawback. We try to develop a dynamic reordering heuristic which has not to construct any temporary ROBDD. We introduce some information-theoretical concepts, which also consider symmetries, that can lead to new, effective and practical ordering algorithms.

The paper is structured as follows. In the next section we briefly review some definitions. Section 3 presents the new approach for dynamic variable reordering. The paper closes with some remarks concerning future work necessary to reach our goal.

2 Preliminaries

This brief review is taken in part from the paper of Panda and Somenzi [17] and Möller et.al. [14].

2.1 Cofactors and Shannon Decompositon

Let f be a completely specified Boolean function over a set of input variables $X = \{x_1, \ldots, x_n\}$. For a constant $b \in \{0, 1\}$ and a variable $x_i \in X$ the *(Shannon) cofactor* $f|_{x_i=b}$ of f with respect to $x_i = b$ is defined by

$$f|_{x_i=b}(x_1, \ldots, x_n) := f(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n).$$

Instead of $f|_{x_i=0}$ and $f|_{x_i=1}$ we also write $f_{x_i'}$ and f_{x_i}, respectively. The cofactor of f with respect to a set of variables and constants is defined inductively.

It is easy to verify that the equation

$$f = x_i' \cdot f_{x_i'} + x_i \cdot f_{x_i}$$

holds, where x_i' is the negation of literal x_i. The equation is called *Shannon decomposition* in literature. It is the basis of the representation of Boolean functions by Binary Decision Diagrams.

2.2 Binary Decision Diagrams

Binary Decision Diagrams (BDDs) are an efficient data structure for the representation of logic functions. A BDD representing a set of Boolean functions $\{f_1, \ldots, f_m\}$ is a directed acyclic graph (DAG) with m roots – one for each function – and two leaves. One leaf represents the constant 1 and the other represents the constant 0. An internal node N of the DAG is labeled with a variable x_i and has two children, T and E. The function represented by N, denoted by f_N is given by $f_N = x_i \cdot f_T + x_i' \cdot f_F$.

It is customary to impose the restriction that the variables are ordered along all paths of the DAG in the same way, that no isomorphic subgraphs exist, and that there is no internal node N with $f_T = f_E$. Under these restrictions, BDDs provide a canonical representation of logic functions. Especially, f_T from above equals the positive cofactor f_{x_i} and f_F equals the negative cofactor $f_{x_i'}$. Note that canonicity makes BDDs suitable for logic verification whenever the BDD of a Boolean function can be constructed in reasonable time.

It is also customary to attach a complementary attribute to arcs in the DAG. Proper use of complementation attributes insures that complementary functions are represented by the same DAG and that canonicity is preserved.

2.3 Symmetric variables

Symmetry plays a crucial role in our new approach for dynamic variable reordering. We consider symmetric Boolean functions as originally defined by Shannon [20, 21] considering symmetry with complementation, too.

Definition 1 *A Boolean function $f(x_1, \ldots, x_n)$ is symmetric in x_i and x_j if the interchange of x_i and x_j leaves the function identically the same; f is symmetric in x_i and x_j' (alternatively x_i' and x_j), if the interchange of x_i and x_j' leaves the function identically the same.*

It is well known that

Theorem 1 (Naive symmetry check) *f is symmetric in x_i and x_j if and only if $f_{x_i,x'_j} = f_{x'_i,x_j}$.*

The naive symmetry check is very popular, but a handicap of it is that temporary ROBDDs have to be created if x_j does not directly follow x_i or vice versa in the variable order of the initial ROBDD. The creation of these ROBDDs may be very time consuming. Möller et al. [14] have presented methods to accelerate symmetry detection by detecting as many asymmetric pairs of variables as possible by structural properties of the initial ROBDD to be able to avoid the naive symmetry check for those pairs. These methods reduce the running time for symmetry detection by a factor of about 15 compared to the naive method.

3 The crucial idea of the new approach

The crucial idea of our new variable ordering heuristic is to define an attraction measure $\mathcal{A}(f, x_i, x_j)$ (or repulsion measure $\mathcal{R}(f, x_i, x_j)$) between any two variables x_i and x_j of a Boolean function f such that

$$\sum_{i=0}^{n-1} \sum_{j=i+1}^{n} \frac{\mathcal{A}(f, x_{\pi(i)}, x_{\pi(j)})}{\varphi(j-i)} \qquad (1)$$

is large if and only if the variable order

$$(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(i)}, x_{\pi(i+1)}, \ldots, x_{\pi(n)})$$

defined by permutation $\pi : \{0, 1, \ldots, n\} \to \{0, \ldots, n\}$ is efficient with respect to the size of the corresponding ROBDD. By function φ we can take care that variables with high repulsion are placed at very different positions.

Note that we have introduced a new variable x_0. We need it in order to be able to attract a variable to the beginning of the variable order. Thus the permutations π we consider all have the property $\pi(0) = 0$.

4 Some attraction measures

In this section we present and discuss some attraction measures which probably meet the condition formulated in Section 3.

4.1 Informational contribution of a variable

The informational contribution of a variable x_i has been defined by Jain et.al. [10] as

$$ic(x_i) := \frac{m\left(f_{x_i} \oplus f_{x'_i}\right)}{m(f)},$$

where $m(g)$ denotes the total number of true minterms of the Boolean function g. The greater the informational contribution of x_i, the larger is the total number of paths from the root to a leaf of the ROBDD sensitive to variable x_i.

Jain et.al. conjecture that it is a good idea to select the variable that has the greatest informational contribution as the root of the ROBDD, i.e., ordering the variables according to decreasing informational contribution. The conjecture of Jain et.al. has not been proven correct by experiments. We conjecture that one has to order according to increasing informational contribution, because variables with poor informational contribution do not widen the ROBDD very much. This raises hope that the breadth of the ROBDD increases only very slowly. This idea results in the definition of a first attraction measure

$$\mathcal{A}_0(f, x_i, x_j) = \begin{cases} 1 - ic(x_i) & \text{if } j = 0 \\ 1 - ic(x_j) & \text{if } i = 0 \\ 0 & \text{else.} \end{cases}$$

Note that, if variable x_i has poor informational contribution, it is strongly attracted to variable x_0, i.e, to the beginning of the variable order.

4.2 Conditional informational contribution of a variable

Another conjecture is that variable x_i should be placed close to x_j if the conditional informational contribution $ic(x_j \mid x_i)$ of x_j knowing the value of x_i is small. This idea can be integrated in the attraction measure $\mathcal{A}_0$ defined so far and we obtain:

$$\mathcal{A}_1(f, x_i, x_j) = \begin{cases} 1 - ic(x_i) & \text{if } j = 0 \\ 1 - ic(x_j) & \text{if } i = 0 \\ 1 - \frac{1}{2} \cdot \left(\frac{m\left(f_{x_i x_j} \oplus f_{x_i x_j'}\right)}{m(f)} + \frac{m\left(f_{x_i' x_j} \oplus f_{x_i' x_j'}\right)}{m(f)} \right) & \text{else.} \end{cases}$$

Note that, if the conditional informational contribution x_j knowing x_i is logic 1 is 0, then the attraction of x_j to x_i is at least $\frac{1}{2}$. If the conditional informational contribution of x_j is poor (great) in both cases, the attraction of x_j to x_i tends to 1 (0).

4.3 Degree of symmetry between two variables

In order to take into account that partial symmetric variables tend to form blocks in optimal variable orders we need an attraction measure expressing this fact. The definition

$$\mathcal{A}_2(f, x_i, x_j) = 1 - \min \left\{ \frac{m\left(f_{x_i x_j'} \oplus f_{x_i' x_j}\right)}{m(f)}, \frac{m\left(f_{x_i x_j} \oplus f_{x_i' x_j'}\right)}{m(f)} \right\}$$

will probably do the job. Note that if f is partial symmetric in x_i and x_j or partial symmetric in x_i and x_j', the attraction $\mathcal{A}_2(f, x_i, x_j)$ is maximal, i.e., equals 1. If f is almost partial symmetric in x_i and x_j or x_i and x_j', the attraction will be about 1.

4.4 Overall attraction measure

The overall attraction measure can now be defined by

$$\mathcal{A}(f, x_i, x_j) = \max \left\{ c_1 \cdot \mathcal{A}_1(f, x_i, x_j), c_2 \cdot \mathcal{A}_2(f, x_i, x_j) \right\}$$

for some constants c_1 and c_2.

5 Future work

Of course there are a lot of open problems which have to be solved in order to reach the goal.

First we must show that the attraction measure proposed in Section 4.4 has the property formulated in Section 3, i.e., that sum (1) is large if and only if π is an efficient variable order with respect to the size of the corresponding ROBDD.

After having proven this by experiments, we have to attack two further problems in order to really obtain an efficient heuristic:

- *We have to find methods which only use structural properties of the initial* ROBDD *in order to well approximate the attraction measure.*

  Computing the cofactor of a Boolean function f with respect to variable x_i requires temporary ROBDDs if x_i is not the root variable of f. Thus in order to accelerate the computation of the attraction measure we have to develop quantitative methods which only uses structural properties of the initial ROBDD. We will try to generalize for our purpose the qualitative methods for detection of symmetries presented in [14].

- *We have to generalize the method proposed to sets of functions.*

  The attraction measures above are defined so far only for single valued Boolean functions. However in practice, we deal with multioutput Boolean functions which are represented by ROBDDs with multiple roots, i.e., we have to find one variable order for all the single outputs. Is there any way to generalize the attraction measure presented so far in order to be able to efficiently apply them to sets of Boolean functions?

To understand from the theoretical point of view why some variable orders are efficient, we have to prove (least) upper bounds for the sizes of the ROBDDs of Boolean functions with specific attractions. Here, we will try to generalize the results proven in [12].

References

[1] S.B. Akers. Binary Decision Diagrams. *IEEE Trans. on CAD*, 27(6):509–516, 1978.

[2] B. Bollig, M. Löbbing, and I. Wegener. Simulated annealing to improve variable orderings for OBDDs. In *Int'l Workshop on Logic Synthesis*, Granlibakken (CA), May 1995.

[3] R.E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Trans. on CAD*, 35(8):677–691, 1986.

[4] R.E. Bryant. Symbolic Boolean Manipulation with Ordered Binary-Decision diagrams. *ACM, Comp. Surveys*, 24:293–318, 1992.

[5] R. Drechsler, B. Becker, and N. Göckel. A genetic algorithm for variable ordering of obdds. In *Int'l Workshop on Logic Synthesis*, Granlibakken (CA), May 1995.

[6] E. Felt, G. York, R. Brayton, and A. Sangiovanni-Vincentelli. Dynamic Variable Reordering for BDD Minimization. In *European Conf. on Design Automation*, pages 130–135, February 1993.

[7] M. Fujita, H. Fujisawa, and N.Kawato. Evaluation and Improvements of Boolean Comparison Methods Based on Binary Decision Diagrams. In *IEEE Int'l Conf. on CAD*, pages 2–5, November 1988.

[8] M. Fujita, Y. Matsunaga, and T. Kakuda. On Variable Ordering of Binary Decision Diagrams for the Application of Multi-level Logic Synthesis. In *European Conf. on Design Automation*, pages 50–54, February 1991.

[9] N. Ishiura, H. Sawada, and S. Yajima. Minimization of Binary Decision Diagrams Based on Exchanges of Variables. In *IEEE Int'l Conf. on CAD*, pages 472–475, November 1991.

[10] J. Jain, J. Bitner, D. Moundanos, J.A. Abraham, and Fussell D.S. A New Scheme to Compute Variable Orders for Binary Decision Diagrams. In *Great Lakes Symp. VLSI*, pages 105–108, 1994.

[11] C.Y. Lee. Representation of switching circuits by binary decision diagrams. *Bell System Technical Journal*, 38:985–999, 1959.

[12] L. Litan, P. Molitor, and D. Möller. Least upper bounds on the sizes of symmetric variable order based OBDDs. In *Great Lakes Symp. VLSI*, pages 126–129, Iowa, March 1996.

[13] S. Malik, A.R. Wang, R.K. Brayton, and A. Sangiovanni-Vincentelli. Logic Verification using Binary Decision Diagrams in a Logic Synthesis Environment. In *IEEE Int'l Conf. on CAD*, pages 6–9, November 1988.

[14] D. Möller, J. Mohnke, and M. Weber. Detection of Symmetry of Boolean Functions Represented by ROBDDs. In *IEEE Int'l Conf. on CAD*, pages 680–684, November 1993.

[15] D. Möller, P. Molitor, and R. Drechsler. *Novel Approaches in Logic and Architecture Synthesis*, chapter *Symmetry based Variable Ordering for ROBDDs*, pages 70–81. Chapman & Hall, 1995.

[16] B.M.E. Moret. Decision Trees and Diagrams. *ACM, Comp. Surveys*, 14(4):593–623, 1982.

[17] S. Panda and F. Somenzi. Who Are the Variables in Your Neighborhood. In *IEEE Int'l Conf. on CAD*, pages 74–77, 1995.

[18] S. Panda, F. Somenzi, and B.F. Plessier. Symmetry Detection and Dynamic Variable Ordering of Decision Diagrams. In *IEEE Int'l Conf. on CAD*, pages 628–631, 1994.

[19] R. Rudell. Dynamic Variable Ordering for Ordered Binary Decision Diagrams. In *IEEE Int'l Conf. on CAD*, pages 42–47, November 1993.

[20] C.E. Shannon. A symbolic analysis of relay and switching circuits. *AIEE Transactions*, 57:713–723, 1938.

[21] C.E. Shannon. The Synthesis of Two-Terminal Switching Circuits. *The Bell System Technical Journal*, 28:59–98, January 1949.