Parity OBDDs cannot be handled efficiently enough*

Martin Löbbing, Detlef Sieling, Ingo Wegener
FB Informatik, LS II, Univ. Dortmund
44221 Dortmund, Fed. Rep. of Germany
loebbing, sieling, wegener@ls2.cs.uni-dortmund.de

Abstract Parity Ordered Binary Decision Diagrams (POBDDs) seem to be a promising alternative to OBDDs because they admit a more compact representation of Boolean functions and polynomial time algorithms for all important operations on Boolean functions. However, for the reduction of POBDDs only an algorithm based on Gaussian elimination is known. Since the reduction is used quite frequently, a linear time algorithm would be desirable. In this note it is proved that it is unlikely that such an algorithm exists because a linear time algorithm for POBDD reduction would imply a linear time algorithm for matrix rank computation.

Keywords Parity OBDDs, reduction algorithm, data structures, theory of computation.

1 Introduction

The most successful approach to problems in VLSI design like circuit verification or model checking is the representation of Boolean functions by Ordered Binary Decision Diagrams (OBDDs), which were introduced by Bryant (1986). The reason for the success of OBDDs is that for operations on Boolean functions represented by OBDDs efficient algorithms are known. Examples for such operations are the equivalence test or the synthesis operation, i.e., the computation of an OBDD for $f_1 \otimes f_2$ from OBDDs for f_1 and f_2, where $\otimes$ is a binary Boolean operator. Another nice property of OBDDs is that the minimum size OBDD (also called the reduced OBDD) for each function and each variable ordering is unique up to isomorphism and can be computed efficiently. This operation is called reduction. The reduction is necessary when performing sequences of operations on OBDDs. An example where such a sequence of operations is performed is the computation of an OBDD for a function given by a circuit. This can be done in the following way. First OBDDs representing the input variables are created. Then we run through the circuit in some topological order. For each gate we compute an OBDD for the function computed at the output of this gate from the OBDDs representing the functions computed at the inputs of this gate. If between these applications of the synthesis algorithm no reduction is performed, the size of the computed OBDDs blows up so that this approach is not feasible. For this reason in current OBDD packages the reduction algorithm is integrated into the synthesis algorithm, i.e., the synthesis algorithm only produces reduced OBDDs.

However, there are many functions without OBDDs of reasonable size. Hence, it is an important issue in research to look for generalizations of OBDDs which allow succinct representations of more

*The authors were supported in part by DFG grant We 1066/8.

functions than OBDDs. Examples of such generalizations are IBDDs (Jain, Bitner, Abadir, Abraham and Fussell (1997)), XBDDs (Jeong, Plessier, Hachtel and Somenzi (1991)), graph driven BDDs (Sieling and Wegener (1995), Gergov and Meinel (1994)) and partitioned BDDs (Jain, Bitner, Fussell, Abraham (1992)). But all these generalizations have the disadvantage that the algorithms for some operations are much less efficient than on OBDDs. For IBDDs and XBDDs the equivalence test is co-NP-complete. For graph driven BDDs and partitioned BDDs the replacement of a variable by a constant may cause an exponential blow-up of the size of the representation. Also for other generalizations of OBDDs one can see the following rule of thumb. If the set of functions with compact representations becomes larger, then some operations can be performed only less efficiently. Parity OBDDs (POBDDs) seem to provide a good trade-off between the compactness of the representation and the efficiency of the algorithms for the manipulation. POBDDs were first considered by Gergov and Meinel (1996). Later on, Waack (1997) recognized that there is also a polynomial time reduction algorithm for POBDDs. Gergov and Meinel (1996) and Waack (1997) also presented examples of functions for which POBDDs are exponentially smaller than OBDDs. The only operation for which no polynomial time algorithm for POBDDs is known is the operation SAT-COUNT, i.e., the problem to compute the number of satisfying inputs of a function represented by a POBDD. For OBDDs this operation can be done in linear time, while for POBDDs the problem is $\#P$-complete. (This follows easily from the $\#P$-completeness of the operation SAT-COUNT for Ordered Functional Decision Diagrams (OFDDs) which was proved by Werchner, Harich, Drechsler and Becker (1996).) Since the operation SAT-COUNT is not necessary for many applications, POBDDs seem to be a promising alternative to OBDDs.

But as outlined above the reduction is used quite frequently. For OBDDs the reduction usually is performed by hashing techniques with an expected linear run time (see e.g. Brace, Rudell and Bryant (1990)). There is also a reduction algorithm for OBDDs with a linear run time in the worst case (Sieling and Wegener (1993)). The reduction algorithm for POBDDs introduced by Waack (1997) uses Gaussian elimination and, hence, is not fast. The aim of this note is to give some evidence that we cannot do better. We give a linear time transformation of the computation of the matrix rank over $\mathbb{F}_2$ to the reduction of POBDDs. Problems of linear algebra like matrix multiplication, matrix inversion, computing LUP-decompositions or matrix rank computation are well-studied problems (see e.g. Strassen (1990)). The fastest known algorithms for these problems working on $n \times n$-matrices have a run time of $O(n^{2.36})$ (Coppersmith and Winograd (1987)). However, because of the large constants hidden by the O-notation one can say that the fastest practical algorithm is based on the method of Strassen (1969) for matrix multiplication and has a run time of $O(n^{2.81})$. This is far away from linear time $O(n^2)$. By our proof a linear time algorithm for the reduction of POBDDs would imply a linear time algorithm for the computation of the matrix rank over $\mathbb{F}_2$. Hence, we cannot expect a linear time algorithm for the reduction of POBDDs.

2 Definition and Properties of POBDDs

In this section we recall the definition and some properties of POBDDs due to Waack (1997).

A POBDD representing some Boolean function $f(x_1, \ldots, x_n)$ is a directed acyclic graph with one source and one sink. The nonsink nodes, also called interior nodes, are labeled by one of the variables $x_1, \ldots, x_n$ and may have an arbitrary number of outgoing edges. Each outgoing edge is labeled by 0 or 1. If a node v is reached from u via an edge labeled by $c \in \{0, 1\}$, we say that v is a c-successor of u. Furthermore, we assume that a variable ordering is fixed and that on each path from the source to the sink the variables are arranged according to this ordering. In the following we assume w.l.o.g. that this ordering is $x_1, \ldots, x_n$.

A path from the source to the sink is called a computation path for the input $a = (a_1, \ldots, a_n)$ if on this path at each interior node v labeled by x_i the outgoing edge is labeled by a_i. For each input there may be several computation paths. The function represented by a POBDD takes the value 1 for some input a iff the number of computation paths for a is odd.

For each node v of a POBDD we define the function f_v associated with v as the function that we obtain by considering v as the source of a POBDD. In particular, the function associated with the sink is the constant function 1. In the following the i-th level of a POBDD denotes the set of all nodes labeled by x_i. The $(n+1)$-th level is the set only consisting of the sink. The size $|\mathcal{B}|$ of a POBDD $\mathcal{B}$ is the number of its nodes including the sink. The number of edges of $\mathcal{B}$ and, therefore, the space needed to store $\mathcal{B}$ is bounded by $O(|\mathcal{B}|^2)$.

Different from OBDDs minimum size POBDDs for a function and a fixed variable ordering are not necessarily unique up to isomorphism. However, Waack (1997) presented a characterization of minimum size POBDDs for a fixed function and a fixed variable ordering. This characterization describes the minimum number of nodes on each level. A POBDD with the minimum number of nodes on each level is called a reduced POBDD. In order to present the characterization of reduced POBDDs we introduce the following notation. First we remark that each Boolean function f over the variables $x_1, \ldots, x_n$ can be seen as a 2^n-dimensional vector containing the truth table of f. Now let $\mathcal{B}$ be a POBDD for f, and let $k \in \{1, \ldots, n+1\}$. We define $B_k^{\mathcal{B}}$ as the vector space generated by the functions associated with the nodes of the levels $k, \ldots, n+1$ of $\mathcal{B}$, i.e.,

$$B_k^{\mathcal{B}} = \mathrm{span}\{f_v | v \text{ is a node on level } k, \ldots, n+1\}.$$

Similarly, we define B_k^f as the vector space generated by the subfunctions of f depending on $x_i, \ldots, x_n$, where $i \in \{k, \ldots, n+1\}$, i.e.,

$$B_k^f = \mathrm{span}\left\{ \bigcup_{i=k}^{n+1} \{f_{|x_1=c_1,\ldots,x_{i-1}=c_{i-1}} | c_1, \ldots, c_{i-1} \in \{0,1\}\} \right\}.$$

The following theorem is the characterization of POBDDs due to Waack (1997).

Theorem 1 Let $\mathcal{B}$ be a POBDD for the function $f(x_1, \ldots, x_n)$ with the variable ordering $x_1, \ldots, x_n$. Then for each $k \in \{1, \ldots, n+1\}$ it holds that $B_k^f \subseteq B_k^{\mathcal{B}}$. If $\mathcal{B}$ is reduced, then $B_k^f = B_k^{\mathcal{B}}$ and $|\mathcal{B}| = \dim B_1^f$. Furthermore, the number of nodes labeled by x_k in $\mathcal{B}$ is equal to $\dim B_k^f - \dim B_{k+1}^f$.

Waack (1997) also presented a polynomial time algorithm to compute a reduced POBDD for each function f and each variable ordering from each POBDD for f and this variable ordering. Roughly for each level k the reduction algorithm computes a basis of the vector space B_k^f / B_{k+1}^f using Gaussian elimination. The run time of the algorithm is $O(n|\mathcal{B}|^\omega)$, where $O(|\mathcal{B}|^\omega)$ is the run time for Gaussian elimination, i.e., theoretically we have $\omega \approx 2.36$ and practically $\omega \approx 2.81$. Since a basis of a vector space is not unique, also a reduced POBDD is not unique up to isomorphism.

Finally, we shortly describe the algorithm of Gergov and Meinel (1996) and Waack (1997) for the problem $\oplus$-synthesis, i.e., the computation of a POBDD $\mathcal{B}$ for $f_1 \oplus f_2$ from POBDDs $\mathcal{B}_1$ and $\mathcal{B}_2$ for f_1 and f_2 (and the same variable ordering $x_1, \ldots, x_n$). Let x_i and x_j be the labels of the sources of $\mathcal{B}_1$

and $\mathcal{B}_2$, resp. If $i < j$ we create a new source of $\mathcal{B}_2$ which is labeled by x_i and has two outgoing edges leading to the original source of $\mathcal{B}_2$. One of the edges is labeled by 0 and the other one by 1. The function represented by $\mathcal{B}_2$ does not change by this modification. The case $i > j$ is handled similarly. Now the sources of $\mathcal{B}_1$ and $\mathcal{B}_2$ have the same label. We obtain the POBDD $\mathcal{B}$ by merging the sources of $\mathcal{B}_1$ and $\mathcal{B}_2$ and by merging the sinks of $\mathcal{B}_1$ and $\mathcal{B}_2$. The resulting POBDD represents $f_1 \oplus f_2$. We remark that $\mathcal{B}$ is not necessarily reduced.

3 The Linear Time Transformation of Matrix Rank Computation to POBDD Reduction

First we recall the notion of linear time transformations (Preparata and Shamos (1985)). We use the term transformation instead of reduction in order to avoid confusion with the reduction of POBDDs. Let A and B be problems. We say that A is linear time transformable to B iff A can be solved in the following way: First from the instance x of A an instance of B is computed in linear time. This instance is solved by an oracle for B. The solution has to be of linear size. Finally, by a linear time computation a solution of the given instance of A is obtained. It is obvious, that for each $\alpha \geq 1$ an algorithm for B with the run time $O(n^\alpha)$ implies an algorithm for A with the run time $O(n^\alpha)$.

We are going to prove the following theorem.

Theorem 2 If there is a linear time algorithm for the reduction of POBDDs, then there is also a linear time algorithm for the computation of the matrix rank over $\mathbb{F}_2$.

We give two different proofs for this theorem. By the first proof we show how to construct a POBDD $\mathcal{B}$ for a given Boolean $n \times n$-matrix M, so that after reduction of $\mathcal{B}$ the rank of M can be computed easily. One may argue against this proof that the reduction algorithm for POBDDs is usually applied only on POBDDs $\mathcal{B}$ that are results of the synthesis of reduced POBDDs $\mathcal{B}_1$ and $\mathcal{B}_2$. In other words, one may say that the lower bound suggested by this proof only holds for instances that do not occur in applications. For this reason we give a second linear time transformation. From a given Boolean $n \times n$-matrix M we construct two reduced POBDDs $\mathcal{B}_1$ and $\mathcal{B}_2$. These POBDDs are combined into a single POBDD $\mathcal{B}$ by applying the $\oplus$-synthesis as outlined in the last section. Again after the reduction of $\mathcal{B}$ the rank of M can be computed easily.

For the first proof let a Boolean $n \times n$-matrix M be given. In linear time $O(n^2)$ we construct a POBDD $\mathcal{B}$ with $3n$ nodes $u_1, \ldots, u_n, v_1, \ldots, v_n, w_1, \ldots, w_n$ and $O(n^2)$ edges. The function f computed by this POBDD is defined over the variables $x_1, \ldots, x_{2n+1}$, which are arranged in this order in the POBDD. The source of the POBDD is u_1. For $i = 1, \ldots, n$ the node u_i is labeled by x_i and has as the only 0-successor the node v_i. For $i = 1, \ldots, n-1$ the only 1-successor of u_i is the node u_{i+1}. The node u_n does not have any 1-successor. The nodes $v_1, \ldots, v_n$ are labeled by x_{n+1} and do not have any 0-successor. The node v_i has the 1-successor w_j iff $M(i,j) = 1$. Finally, each node w_i is labeled by x_{n+i+1} and has as the only 1-successor the sink. An example of a Boolean matrix and the constructed POBDD is shown in Figure 1. In the figure edges labeled by 0 are drawn in dashed lines and edges labeled by 1 are drawn in solid lines.

(Insert Figure 1 here.)

Now let $\mathcal{B}'$ be a POBDD obtained by the reduction of $\mathcal{B}$. We claim that the rank of M is equal to the number of nodes of $\mathcal{B}'$ labeled by x_{n+1}. The functions $f_{w_1}, \ldots, f_{w_n}$, which are associated with the nodes $w_1, \ldots, w_n$ in $\mathcal{B}$, are the functions $x_{n+2}, \ldots, x_{2n+1}$. In particular, all these functions are linearly independent. Let $j(1), \ldots, j(l)$ be the indices of the columns of M which are not constant 0. Then we have $B^f_{n+2} = \mathrm{span}\{f_{w_{j(1)}}, \ldots, f_{w_{j(l)}}\}$ and $\dim B^f_{n+2} = l$. The function f_{v_i} associated with v_i is equal to $x_{n+1} \wedge \langle M_i, (x_{n+2}, \ldots, x_{2n+1})\rangle$, where M_i is the i-th row of M and $\langle \cdot, \cdot \rangle$ is the inner product. It is easy to see that this mapping from the rows of M to the function associated with the x_{n+1}-nodes of $\mathcal{B}$ is linear. Hence, $\mathrm{rank}\, M = \dim(\mathrm{span}\{f_{v_1}, \ldots, f_{v_n}\})$. Furthermore, the functions f_{v_i} are the subfunctions of f that can be obtained from f by replacing $x_1, \ldots, x_n$ by constants. Hence, B^f_{n+1} is the direct sum of $\mathrm{span}\{f_{v_1}, \ldots, f_{v_n}\}$ and B^f_{n+2}. Then $\mathrm{rank}\, M = \dim(\mathrm{span}\{f_{v_1}, \ldots, f_{v_n}\}) = \dim B^f_{n+1} - \dim B^f_{n+2}$, which by Theorem 1 is equal to the number of x_{n+1}-nodes of $\mathcal{B}'$.

Since it is not clear whether the constructed POBDD $\mathcal{B}$ may occur as the result of the synthesis of two reduced POBDDs, we provide an alternative proof. Again let a Boolean $n \times n$-matrix M be given. We compute in linear time a $2n \times 2n$-matrix M'. Let I_n be the $n \times n$-identity matrix and let 0_n be the $n \times n$-zero matrix. Then let

$$M' = \left(\begin{array}{c} M_1 \\ \hline M_2 \end{array}\right) = \left(\begin{array}{c|c} M & I_n \\ \hline 0_n & I_n \end{array}\right).$$

M_1 is an $n \times 2n$-matrix which consists of M and I_n. Hence, the rows of M_1 are linearly independent. The same holds for M_2, which is a $n \times 2n$-matrix consisting of the zero matrix and the identity matrix. Furthermore, observe that the left half of M', which consists of M and 0_n, has the rank of M and that the right half of M', which consists of two copies of I_n, has the rank n. Since the columns of both halves are linearly independent, it follows that $\mathrm{rank}\, M = \mathrm{rank}\, M' - n$.

Now we construct two POBDDs $\mathcal{B}_1$ and $\mathcal{B}_2$ from M_1 and M_2. Both POBDDs look like the POBDD used in the first proof with the exception that for both POBDDs an extra source is added. For an example see Fig. 2. The functions f_1 and f_2 represented by $\mathcal{B}_1$ and $\mathcal{B}_2$ are defined on the variables $x_1, \ldots, x_{4n+2}$. The POBDD $\mathcal{B}_1$ consists of the nodes $s, u_1, \ldots, u_n, v_1, \ldots, v_n, w_1, \ldots, w_{2n}$. The source s is labeled by x_1 and has as the only 0-successor the node u_1. The nodes $u_1, \ldots, u_n$ are labeled by $x_2, \ldots, x_{n+1}$, resp., and form a switch as in the first proof, i.e., the only 0-successor of u_i is w_i and the only 1-successor of u_i (where $i < n$) is u_{i+1}. The nodes $v_1, \ldots, v_n$ represent the rows of M_1, i.e., they are labeled by x_{2n+2}, and w_j is the 1-successor of v_i iff $M_1(i,j) = 1$. The nodes $w_1, \ldots, w_{2n}$ represent the columns of M_1. They are labeled by $x_{2n+3}, \ldots, x_{4n+2}$ and have as the only 1-successor the sink. If the j-th column of M_1 is a zero vector, we omit the node w_j, because it is not reachable from the source. Since the rows of M_1 are linearly independent, the POBDD $\mathcal{B}_1$ is reduced.

(Insert Figure 2 here.)

The POBDD $\mathcal{B}_2$ is defined in the same way for the matrix M_2. It consists of the nodes $s', u'_1, \ldots, u'_n, v'_1, \ldots, v'_n, w'_{n+1}, \ldots, w'_{2n}$. The source is the node s'. It is labeled by x_1 and has as the only 1-successor the node u'_1. The nodes $u'_1, \ldots, u'_n$ are labeled by $x_{n+2}, \ldots, x_{2n+1}$ and again they form a switch. The nodes $v'_1, \ldots, v'_n$ are labeled by x_{2n+2} and represent the rows of M_2. There are no nodes $w'_1, \ldots, w'_n$, since the corresponding columns of M_2 are zero vectors. The nodes $w'_{n+1}, \ldots, w'_{2n}$ represent the columns $n+1, \ldots, 2n$ of M_2. As in $\mathcal{B}_1$ these nodes are labeled by $x_{3n+3}, \ldots, x_{4n+2}$. Since the rows of M_2 are linearly independent, also B_2 is reduced.

Now we apply the $\oplus$-synthesis on $\mathcal{B}_1$ and $\mathcal{B}_2$ and obtain a POBDD $\mathcal{B}$ for $f = f_1 \oplus f_2$. The source of $\mathcal{B}$ is obtained by merging s and s'. Also the sinks are merged. It is easy to see that also the nodes

w_i and w'_i, where $i \in \{n+1, \ldots, 2n\}$ can be merged. Since there is only one computation path from the source to each node v_i and each node v'_i, the functions associated with the nodes v_i and v'_i are the subfunctions of f that can be obtained from f by replacing $x_1, \ldots, x_{2n+1}$ by constants. Now we may apply exactly the same arguments as in the first proof and obtain that the number of x_{2n+2}-nodes after the reduction of $\mathcal{B}$ is equal to the rank of M'. Because of $\operatorname{rank} M = \operatorname{rank} M' - n$ the rank of M can be computed in linear time from the result of the reduction.

Conclusion

By providing linear time transformations from matrix rank computation we showed that it is quite unlikely that there are linear time algorithms for the reduction of POBDDs. Since the reduction has to be performed quite frequently, OBDDs cannot be replaced by POBDDs in general. Hence, POBDDs should be used only for the representation and manipulation of functions for which OBDDs are much larger than POBDDs.

Acknowledgment

We would like to thank Martin Dietzfelbinger for helpful discussions.

References

Brace, K.S., Rudell, R.L. and Bryant, R.E. (1990). Efficient implementation of a BDD package. In *Proc. of Design Automation Conference DAC'90*, 40–45.

Bryant, R.E. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers* 35, 677–691.

Coppersmith, D. and Winograd, S. (1987). On the asymptotic complexity of matrix multiplication. *SIAM Journal on Computing* 11, 472–492.

Gergov, J. and Meinel, C. (1994). Efficient Boolean manipulation with OBDD's can be extended to FBDD's. *IEEE Transactions on Computers* 43, 1197–1209.

Gergov, J. and Meinel, C. (1996). Mod-2-OBDDs—A data structure that generalizes EXOR-sum-of-products and ordered binary decision diagrams. *Formal Methods in System Design* 8, 273–282.

Jain, J., Bitner, J.B., Abadir, M.S., Abraham, J.A. and Fussell, D.S. (1997). Indexed BDDs: algorithmic advances in techniques to represent and verify Boolean functions. *IEEE Transactions on Computers* 46, 1230–1245.

Jain, J., Bitner, J., Fussell, D.S. and Abraham, J.A. (1992). Functional partitioning for verification and related problems. In *Proc. of Brown/MIT VLSI Conference*, 210-226.

Jeong, S.-W., Plessier, B., Hachtel, G. and Somenzi, F. (1991). Extended BDD's: Trading off canonicity for structure in verification algorithms. In *Proc. of International Conference on Computer-Aided Design ICCAD'91*, 464–467.

Preparata, F.P. and Shamos, M.I. (1985). *Computational Geometry*, Springer.

Sieling, D. and Wegener, I. (1993). Reduction of OBDDs in linear time. *Information Processing Letters* 48, 139–144.

Sieling, D. and Wegener, I. (1995). Graph driven BDDs — a new data structure for Boolean functions. *Theoretical Computer Science* 141, 283–310.

Strassen, V. (1969). Gaussian elimination is not optimal. *Numerische Mathematik* 13, 354–356.

Strassen, V. (1990). Algebraic complexity theory. In *Handbook of Theoretical Computer Science, Vol. A, Algorithms and Complexity*, Van Leeuwen, J. (ed.), Elsevier, 633–672.

Waack, S. (1997). On the descriptive and algorithmic power of parity ordered binary decision diagrams. In *Proc. of 14th Symposium on Theoretical Aspects of Computer Science STACS'97*, Lecture Notes in Computer Science 1200, Springer, 201–212.

Werchner, R., Harich, T., Drechsler, R. and Becker, B. (1996). Satisfiability problems for OFDDs. In *Representations of Discrete Functions*, Sasao, T., Fujita, M. (eds.), Kluwer Academic Publishers, 233–248.

$$M = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

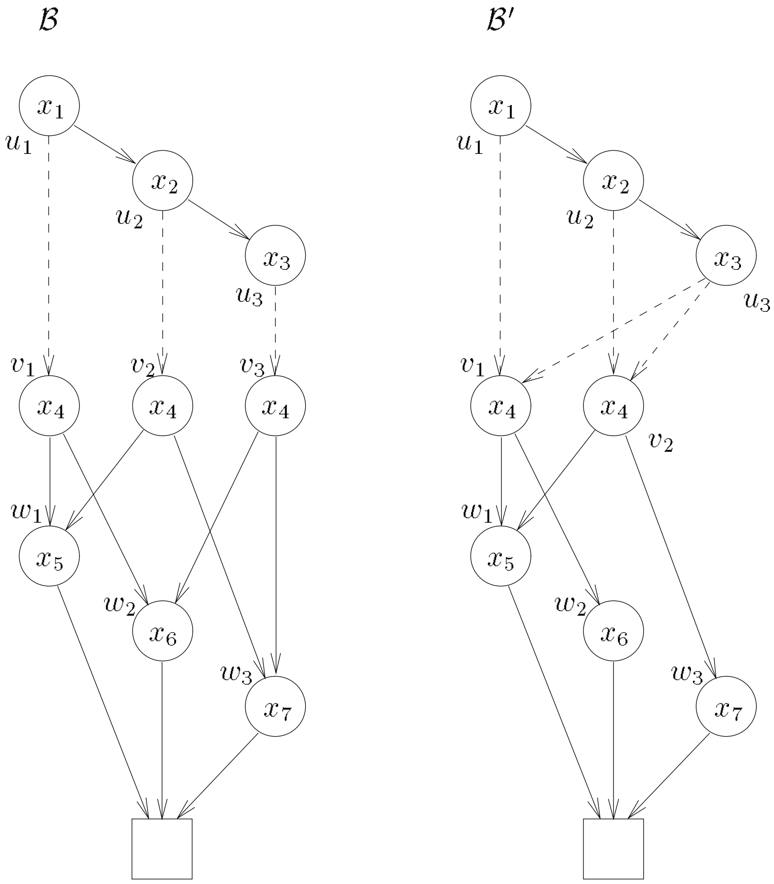

Figure 1: An example of a matrix M, the constructed POBDD $\mathcal{B}$ and the reduced POBDD $\mathcal{B}'$. The rank of M is equal to the number of x_4-nodes of $\mathcal{B}'$.

$$M = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$M' = \left(\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

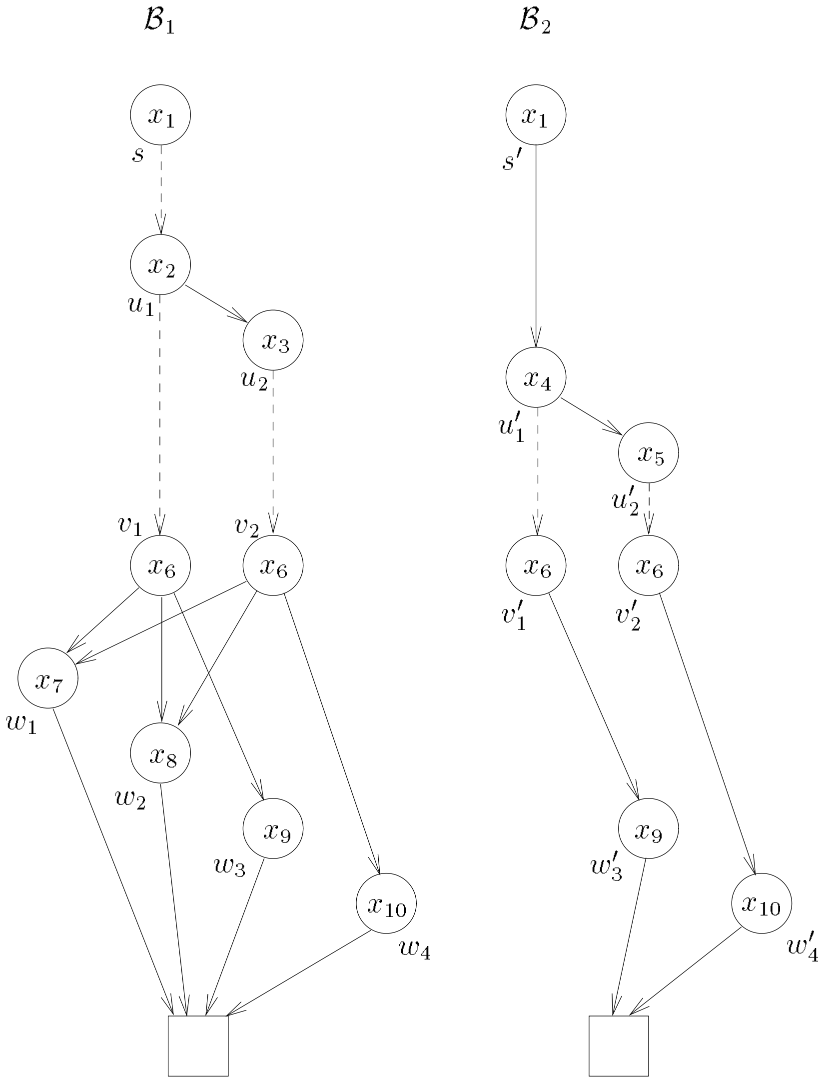

Figure 2: An example of a matrix M and the computed POBDDs $\mathcal{B}_1$ and $\mathcal{B}_2$.