Using Binary Decision Diagrams to Efficiently Represent Morphological Associative Memories

Itzamá López-Yáñez, Cornelio Yáñez-Márquez, Víctor M. Silva-García

Centro de Investigación en Computación, Instituto Politécnico Nacional
Av. Juan de Dios Bátiz s/n, México, D.F., 07738, México
ilopezyb05@ipn.mx
cyanez@cic.ipn.mx
vsilvag@ipn.mx

Abstract. Morphological Associative Memories, an associative memory model which is based on the basic operations of Mathematical Morphology, has been shown to surpass the classical models in practically every aspect. On the other hand, Binary Decision Diagrams have been successful forms to represent Boolean functions. In this paper the authors propose the use of Binary Decision Diagrams to represent morphological associative memories in order to merge two important and very active areas of contemporary scientific research, improving the applicability and benefits shown by both areas.

1. Introduction

Binary Decision Diagrams (BDDs) appeared two decades ago with Bryant's seminal article [1]. This work represented a breakthrough in describing digital circuits and processes of hardware verification. Since then, many authors have developed research works in this area, trying to improve BDDs ability to represent data or the efficiency of the algorithms that manipulate them [2-12]. BDDs have been so superior to prior models of Boolean functions representation that currently they are the state-of-the art data structure to represent Boolean functions and verify circuit correctness in digital circuit Computer Aided Design (CAD) applications [11]. On the other hand, associative memories have been an active area for research in computer sciences by roughly half a century. The ultimate goal of an associative memory is to correctly recall complete patterns from input patterns [13-15]. These patterns might be altered with additive, subtractive or mixed noise. The classical era of associative memories, which includes models such as the Lernmatrix [16], the Correlograph [17] and the Linear Associator [18-19], is represented by the model developed by Hopfield [20], which is simultaneously an associative memory and a neural network. In the late 1990s morphological associative memories were developed by Ritter *et. al.* [21], surpassing the learning and pattern recall capabilities offered by all previous models. Morphological associative memories are based in Mathematical Morphology [22-24].

In this paper we propose the use of Binary Decision Diagrams to represent Morphological associative memories. By doing so, we will be able to merge two important and very active areas of contemporary scientific research, with the intention of

© G. Sidorov, B. Cruz, M. Martínez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 34, 2008, pp. 47-58

Received 01/04/08
Accepted 26/04/08
Final version 28/04/08

improving and broadening the applicability of both models, Binary Decision Diagrams and Morphological Associative Memories. The remaining of the paper is organized as follows. In Sections 2 a survey of BDDs is provided. Section 3 is focused on explaining the Morphological associative memory model. Section 4 contains the core proposal. Section 5 is finally devoted to conclusions.

2. Binary Decision Diagrams

Boolean Algebra is, along with Set Theory, the foundation of all sciences. However, digital systems design and computer sciences are even more reliant on Boolean Algebra: the basic units for every piece of data, every circuit, and every operation are Boolean variables and Boolean operations. Many problems in digital logic design and testing, artificial intelligence, and combinatorics can be expressed as a sequence of operations on Boolean functions [1]. However, in order to properly employ Boolean Algebra, it is necessary to devise forms of representing and manipulating Boolean functions in a symbolic manner. Problem is, many of the tasks one would like to perform with Boolean functions, such as satisfiability or equivalence testing, require solutions to NP-Complete or coNP-Complete problems [25]. Therefore, many forms to represent and manipulate Boolean functions have arisen, being truth tables, Karnaugh maps and Disjunctive Normal Form some classical examples. A different approach, which has seen a lot of success, becoming a very important field of study, is that of Ordered Binary Decision Diagrams (OBDD), proposed initially by Bryant in his seminal article [1].

An OBDD is defined as a rooted, acyclic, and directed graph with a vertex set V containing two types of vertices: terminal and nonterminal. A nonterminal vertex v has as attributes an argument index $index(v) \in \{1, \ldots, n\}$, and two children $low(v)$, and $high(v) \in V$, while a terminal vertex v has as attribute a value $value(v) \in \{0,1\}$. Furthermore, for any nonterminal vertex v, if $low(v)$ is also nonterminal, then we must have $index(v) < index(low(v))$. Similarly, if $high(v)$ is nonterminal, then we must have $index(v) < index(high(v))$ [1, 25]. In order for an OBDD to be reduced, thus becoming a Reduced Ordered Binary Decision Diagram (ROBDD, usually referred to as BDD), it needs to comply with two restrictions: (i) it does not contain distinct vertices v and v' such that the subgraphs rooted by them are isomorphic, and (ii) it contains no vertex v with $low(v) = high(v)$ [1]. In this way, all redundancies of the graph are erased and the number of nodes is reduced to the minimum for the function the OBDD represents, becoming the best possible OBDD representation for that function.

BDDs have some interesting properties. They provide compact, canonical representations of Boolean expressions, and there are efficient algorithms for performing all kinds of logical operations on BDDs [25]. These operations include equivalence checking, satisfiability test, satisfiability count, synthesis (computation of f $\otimes$ g from f and g, where $\otimes$ is a Boolean operation), replacement of variables by functions (substitution), and redundancy test ("does $f(x_1, \ldots, x_n)$ depend on x_i?") [26]. These operations are based on one characteristic BDDs have: for any Boolean function there is exactly one BDD that represents it. Therefore, it is possible to test in constant time whether a BDD is constantly true or false [25]. Note that for this claim to remain true, it is necessary to take into account the order of the variables in the BDD.

The main drawback presented by BDDs is that their efficiency, in terms of the time required by their operations, depends on the size of the graph. The size, meaning the number of nodes, is in turn very strongly dependant on the ordering in which the variables that make up the Boolean function represented by the BDD are processed (i.e. the order in which these variables are passed to the algorithms that build the BDD). A good ordering will give a relatively small graph, while a different order may give a very large graph, whose size grows almost exponentially on the number of variables. There is yet no formal method for finding optimal orderings. However, there are some heuristic methods for finding orderings which do not yield BDDs with exponential size, depending on the family of the Boolean function to be represented [27]. One example of the BDDs dependency on ordering for efficiency is that of the function $(p_1 \wedge q_1) \vee (p_2 \wedge q_2)$, shown in figure 1. When using the ordering $p_1 < q_1 < p_2 < q_2$, the resulting BDD has 4 nodes, as in (a). On the other hand, when using the ordering $p_1 < p_2 < q_1 < q_2$, the resulting BDD has 6 nodes, as in (b).

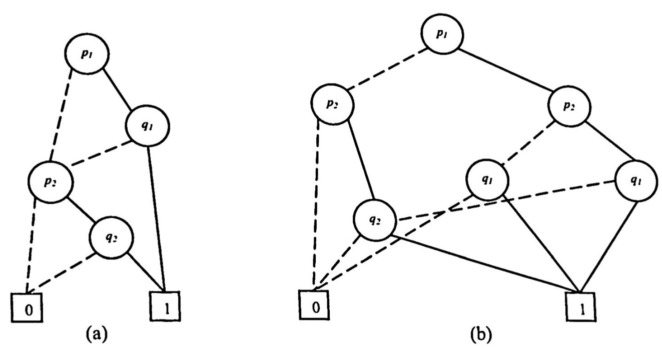

Fig. 1. OBDD's for Boolean function $(p_1 \wedge q_1) \vee (p_2 \wedge q_2)$: (a) with ordering $p_1 < q_1 < p_2 < q_2$, and (b) with ordering $p_1 < p_2 < q_1 < q_2$. *low*(v) children are represented with dotted lines, while *high*(v) are represented with solid lines

BDD's have been applied in a wide number of fields. First of all, the obvious: to represent and solve Boolean functions. However, some of their most important uses have been formal verification of models, hardware verification (for instance, to ascertain that an arithmetic circuit is correct), hardware design (as in CAD applications, where BDDs are the state-of-the-art data structure for representing Boolean functions), complexity theory analysis, planning in non-deterministic domains, image compression, program verification, and cryptanalysis [2-8, 12, 25-28].

Some models used to represent different types of information have been developed, based on BDDs. For instance, Binary Moment Diagrams (and their improvement Multiplicative Binary Moment Diagrams, proposed by Bryant and Chen; BMDs and *BMDs respectively) are used to overcome the weakness of BDDs before integer multiplication, generalizing them in order to work within other ambits, such as integer or real numbers [3-4]. Figure 2 shows the *BMD for the addition and multiplication of two words of three bits each. The size of the graph grows linearly on the size of the words, as reported by Bryant and Chen in [4].

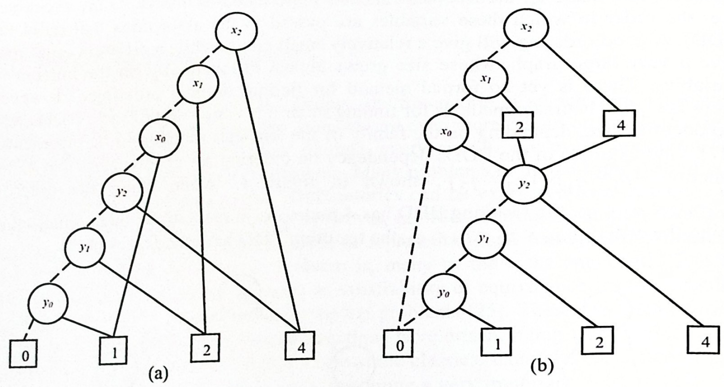

Fig. 2. *BMDs for (a) X + Y, and (b) X * Y, respectively. Both X and Y are words of size 3

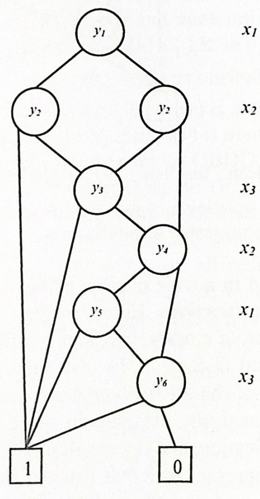

Fig. 3. Example of an implementation of *k*BDDs

Another example is Boolean Functional Vectors (and Partially Ordered Boolean Functional Vectors, proposed by Goel and Bryant in [10]), which represent a bit-level decomposition of the state-set suitable for symbolic simulation, offering also an often more compact representation than BDDs [9-10]. Read-*k*-times BDDs are another variant, which relaxes some of the restrictions on order imposed by BDDs, allowing a variable to be present more than once, which in turns yields better results in some families of functions that had bad behaviour on BDDs. While there had been some theoretical work in Read-k-times BDDs, even defining some different classes, Gunther and Dreschler offer an implementation in [11]. Figure 3 shows an example of a *k*BDD, the general class of Read-k-times BDDs.

3. Morphological Associative Memories

Basic concepts about associative memories were established three decades ago in [14-15, 29], nonetheless here we use the concepts, results and notation introduced in the Yáñez-Márquez's PhD Thesis [13]. An associative memory **M** is a system that relates input patterns, and output patterns, as follows: **x→M→y**, with **x** and **y** being the input and output pattern vectors, respectively. Each input vector forms an association with a corresponding output vector. For *k* integer and positive, the corresponding association will be denoted as $\left(x^k, y^k\right)$. Associative memory **M** is represented by a matrix whose *ij*-th component is m_{ij}. Memory **M** is generated from an *a priori* finite set of known associations, known as the fundamental set of associations. If μ is an index, the fundamental set is represented as: $\left\{\left(x^\mu, y^\mu\right) \mid \mu = 1,2,\mathrm{K}, p\right\}$ with *p* being the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that $x^\mu = y^\mu, \forall \mu \in \left\{1,2,\mathrm{K}, p\right\}$, then **M** is autoassociative, otherwise it is heteroassociative. In this latter case it is possible to establish that $\exists \mu \in \left\{1,2,\mathrm{K}, p\right\}$ for which $x^\mu \neq y^\mu$. A distorted version of a pattern x^k to be recalled will be denoted as $\tilde{x}^k$. If when feeding a distorted version of x^ϖ with $\varpi = \left\{1,2,\mathrm{K}, p\right\}$ to an associative memory **M**, it happens that the output corresponds exactly to the associated pattern y^ϖ, we say that recall is perfect.

Among the variety of associative memory models described in the scientific literature, there are two models that, because of their relevance, it is important to emphasize: morphological associative memories, which were introduced by Ritter *et al.* [21], and Alpha-Beta associative memories, which were introduced by Yáñez-Márquez [13, 22-23]. Because of their excellent characteristics, which allow them to be superior in many aspects to other models for associative memories [21], morphological associative memories served as starter point for the creation and development of the Alpha-Beta associative memories. Thus, both of these models greatly surpass the limits shown by the classical models of associative memories, whether it is learning capacity, recalling accuracy and robustness before noise.

The morphological associative memories are of two kinds, *min* and *max*, and are able to operate in two different modes, heteroassociative and autoassociative. In both cases the learning and recalling phases are based on the use of maximums and minimums of additions, as opposed to the classical models, which use conventional matrix operations for the learning phase and the addition of products for the recalling phase. For the *max* morphological memories, the learning phase uses the maximum of additions, which is equivalent to the definition of the basic morphological operation of *dilation*. On the other hand, for the learning phase of the *min* kind of morphological associative memories, the minimum of additions is used, being this operation equivalent to the basic morphological operation of *erosion*. It is due to this equivalence that morphological associative memories have been given this name [24].

In general terms, morphological associative memories can handle real-valued patterns. However, for the sake of simplicity and expediency, we will only work with integer patterns in this paper.

3.1. The Minimum and Maximum Products

Before stating how morphological associative memories operate, it is necessary to define two basic operations, employed during both the learning and recalling phase. These two matrix operations are the minimum product and the maximum product, which are explained below, as seen on [24].

Let $\mathbf{D}$ be a matrix $\left[d_{ij}\right]_{m\times p}$ and $\mathbf{H}$ be a matrix $\left[h_{ij}\right]_{p\times n}$ whose components are integer numbers. The *maximum product* of $\mathbf{D}$ and $\mathbf{H}$, denoted by $\mathbf{C} = \mathbf{D}\nabla\mathbf{H}$, gives as result a matrix $\mathbf{C} = \left[c_{ij}\right]_{m\times n}$ and is defined as:

$$c_{ij} = \bigvee_{k=1}^{p} \left(d_{ik} + h_{kj}\right) \tag{1}$$

Two particular cases of great importance during the application of the maximum product arise. The first is when $\mathbf{D}$ is a matrix with dimensions mxn and $\mathbf{H}$ is a column vector of dimension n. When applying equation 1 to compute the maximum product $\mathbf{C} = \mathbf{D}\nabla\mathbf{H}$ of $\mathbf{D} = \left[d_{ij}\right]_{m\times n}$ and $\mathbf{H} = \left[h_i\right]_n$ a column vector $\mathbf{C}$ of dimension m is obtained, whose i-th component is:

$$c_i = \bigvee_{j=1}^{n} \left(d_{ij} + h_j\right) \tag{2}$$

The second case is when the maximum product between a column vector of dimension m and a row vector of dimension n is calculated. When applying equation 1 to compute the maximum product $\mathbf{C} = \mathbf{D}\nabla\mathbf{H}$ of $\mathbf{D} = \left[d_i\right]_m$ and $\mathbf{H} = \left[h_j\right]_n$ a matrix $\mathbf{C} = \left[c_{ij}\right]_{m\times n}$ is obtained:

$$c_{ij} = \bigvee_{k=1}^{1} \left(d_{ik} + h_{kj}\right) = \left(d_{i1} + h_{1j}\right) \tag{3}$$

which can be simplified to $c_{ij} = (d_{i1} + h_{1j})$.

Similarly, the *minimum product* $\mathbf{C} = \mathbf{D}\Delta\mathbf{H}$ is defined as:

$$c_{ij} = \bigwedge_{k=1}^{p} (d_{ik} + h_{kj}) \tag{4}$$

Both noteworthy cases arise with respect to the minimum product, just as happened with the maximum product. Their results can be obtained from equations 2 and 3, by substituting the maximum operator $\vee$ by the minimum operator $\wedge$.

3.2. The Learning and Recalling Phases

Because there are two kinds of morphological associative memories, $\bigvee$ and $\bigwedge$, and if we consider that each one of these kinds is able to operate in two different modes, heteroassociative and autoassociative, we have four different available choices.

In this issue, we only talk about the morphological autoassociative memories of kind $\bigvee$. Therefore, the fundamental set takes the form: $\{(x^\mu, x^\mu) \mid \mu = 1,2,\text{K},p\}$.

Besides, the input and output patterns have the same dimension n, and the memory is a square matrix: $\mathrm{M} = [m_{ij}]_{n\times n}$

LEARNING PHASE

STEP 1:

For each $\mu = 1,2,\text{K},p$, and from each couple (x^μ, x^μ) build the matrix: $\left[x^\mu \Delta (-x^\mu)^t\right]_{n\times n}$.

STEP 2:

Apply the binary $\bigvee$ operator to the matrices obtained in step 1 to get $\mathbf{M}$ as: $\mathbf{M} = \bigvee_{\mu=1}^{p} \left[x^\mu \Delta (-x^\mu)^t\right] = [m_{ij}]_{n\times n}$. The ij-th entry is given as $m_{ij} = \bigvee_{\mu=1}^{p} (x_i^\mu - x_j^\mu)$. It is obvious that, $v_{ij} \in B, \forall i \in \{1,2,\text{K},n\}, \forall j \in \{1,2,\text{K},n\}$, where B = {0, 1} since we are dealing with the binary version of morphological autoassociative memories.

RECALLING PHASE

A pattern x^ϖ, with $\omega \in \{1,2,\ldots,p\}$ is presented to the morphological autoassociative memory of kind *max* and the following operation is done: $\mathbf{y} = \mathbf{M}\Delta\mathbf{x}^\varpi$. The result is a column vector y of dimension n, with i-th component given as:

$$y_i = \bigwedge_{j=1}^{n} (m_{ij} + x_j^\omega)$$

4. Merging BDDs and Morphological Associative Memories

Our core proposal consists of representing the operations $r = x_i^\mu - x_j^\mu$, $s = m_{ij} + x_j^\omega$, $\bigvee_{\mu=1}^{p}\left(r^\mu\right)$, and $\bigwedge_{j=1}^{n}\left(s^j\right)$, by employing Binary Decision Diagrams, thus characterizing the learning and recalling phases of morphological associative memories on BDDs.

Given that the output of the operations to be characterized is not bound to the binary set but includes all integer numbers, the original model of BDDs is not suitable to this task. Rather, an extension to this model should be used for this purpose. In this work, Binary Moment Diagrams [3-4] (BMD) are used.

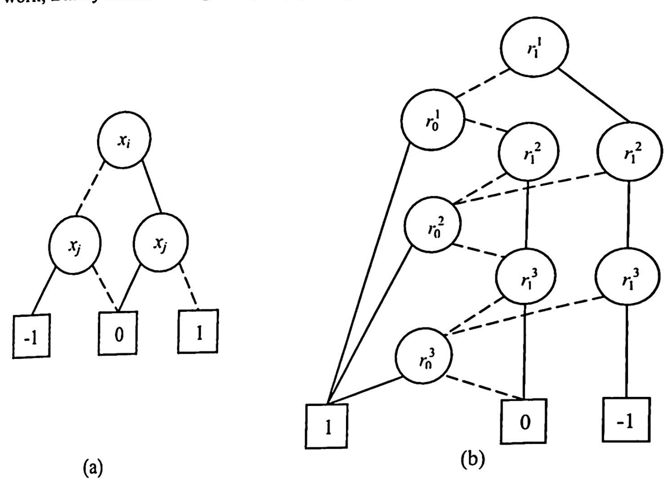

Fig. 4. Characterization on BMD's of the operations used during the learning phase: (a) $r = x_i^\mu - x_j^\mu$, and (b) $\bigvee_{\mu=1}^{p}\left(r^\mu\right)$

One possible characterization on BDDs of the operations $r = x_i^\mu - x_j^\mu$ and $\bigvee_{\mu=1}^{p}\left(r^\mu\right)$, which make up the learning phase of a morphological autoassociative memory of kind *max*, can be seen in figure 4, while figures 5 and 6 depict the operations used during the recalling phase: operation $s = m_{ij} + x_j^\omega$ and operation $\bigwedge_{j=1}^{n}\left(s^j\right)$, respectively.

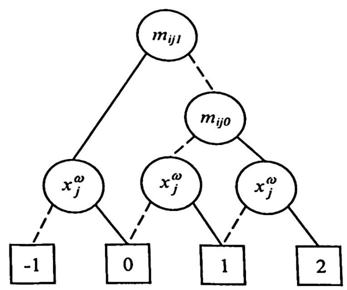

Fig. 5. Characterization on BMD's of the operation $s = m_{ij} + x_j^\omega$, used during the recalling phase

In the case of the $\vee$ and $\wedge$ operations, both are done for three different variables. Notice that, for more than three variables, there are two possible choices: either to build the corresponding BMD for each particular value of p and/or n, or to compare using only two (or three) variables, for as many times as necessary.

The main premise behind our proposal is that through the use of BDDs and their variants to characterize the basic operations of morphological associative memories, we can improve their implementation by lowering their arithmetic density. Also, once the morphological memory model of a specific application has been characterized on BDDs, all the previous results can be applied to the morphological memory too. These results include the development of efficient data structures and associated algorithms to manipulate Boolean and semi-Boolean functions [1-4, 9-12, 27-28].

Our vision is that by enhancing the low arithmetic density of morphological associative memories when compared to other models (in particular to alpha-beta associative memories, its strongest competitor) we will further improve the model, allowing it to be applied to fields in which it has found little interest by the scientific and industry community. Given that lower arithmetic density implies fewer basic operations, this means more efficient results, since larger patterns can be processed in less time. This could lead to an extensive application of morphological associative memories to information retrieval, filtering, and extraction operations on extremely large data bases, such as Bioinformatics (also referred to as Computational Biology). Another implication of this reduced density is the reduced quantity of circuits needed to implement morphological memories on hardware, which would mean better response times and lower power requirements. We strongly believe that this would lead to the application of morphological associative memories on the solution of real time problems, such as control and automation.

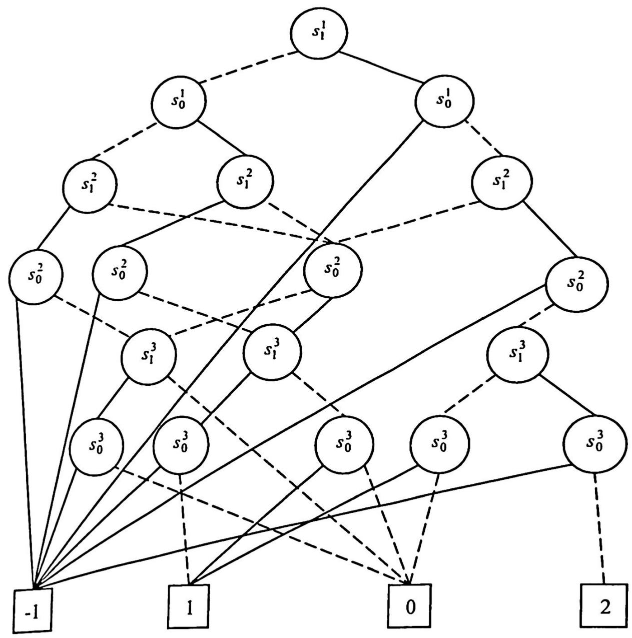

Fig. 6. Characterization on BMD's of the operation $\bigwedge_{j=1}^{n}\left(s^{j}\right)$, used during the recalling phase

Both the application of morphological associative memories to Bioinformatics and their implementation on hardware are areas of great interest to the Alpha-Beta Group. In fact, these two particular aspects are currently being researched and worked upon by members of this group, although their final consequences have not yet been completely explored nor reached. Also, it remains to be seen just how much can the morphological associative memories model be improved by employing BDDs, or if other models can yield better results. The application of morphological associative memories to Data Mining remains practically unexplored, too, making it a new venue for research. This particular area of research is somewhat related to their application to Bioinformatics, since the basis is practically the same: extracting relevant information from a huge repository of data. However, the specific characteristics of each area could mean better results on one than those obtained on the other. In the end, further work should be done here too.

5. Conclusions

In this paper we propose the merging of two areas of computer science research that have been quite successfully used to solve specific problems in a wide range of different fields, both in research and applications. These two areas, Binary Decision Diagrams on one hand, and Morphological Associative Memories on the other hand, and their derived variants, represent competitive models in their respective fields, being also two areas of current and active research. We have presented some challenges that arise with the merging of these two areas, showing the feasibility of applying this merge to solve problems in several research areas.

Acknowledgements
The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, and CIC), the CONACyT, and SNI for their economical support to develop this work.

References

1. Bryant, R. E.: Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers, C-35(8) (1986) 677–691
2. Bryant, R. E., Kukula, J. H.: Formal Methods for Functional Verification. In Kuehlmann, A. (ed): The Best of ICCAD: 20 Years of Excellence in Computer-Aided Design. Kluwer Academic Publishers (2003) 3-16
3. Chen, Y.-A., Bryant, R. E.: An Efficient Graph Representation for Arithmetic Circuit Verification. Computer-Aided Design of Integrated Circuits and Systems. IEEE Transactions, 20(12) (2001) 1443-1454
4. Bryant, R. E., Chen, Y.-A. Verification of Arithmetic Circuits Using Binary Moment Diagrams. Internacional Journal on Software Tools for Technology Transfer (STTT), 3(2) (2001) 137–155
5. Jensen, R. M., Veloso, M. M., Bryant, R. E. Guided Symbolic Universal Planning. Proceedings of the 13th International Conference on Automated Planning and Scheduling ICAPS-03 (2003) 123-132
6. Esparza, J., Schwoon, S.: A BDD-Based Model Checker for Recursive Programs. Lecture Notes in Computer Science, Vol. 2102. Springer-Verlag, Berlin Heidelberg New York (2001) 324.
7. Krause, M.: BDD-Based Cryptanalysis of Keystream Generators, Lecture Notes in Computer Science, Vol. 2332. Springer-Verlag, Berlin Heidelberg New York (2002) 222
8. Starkey, M., Bryant, R. E.: Using Ordered Binary-Decision Diagrams for Compressing Images and Image Sequences. Technical Report CMU-CS-95-105. Carnegie Mellon University, School of Computer Science (1995)
9. Goel, A., Bryant, R. E.: Set Manipulation with Boolean Functional Vectors for Symbolic Reachability Analysis. In 2003 Design Automation and Test in Europe DATE'03 (2003) 816-821
10. Goel, A., Bryant, R. E.: Symbolic Simulation, Model Checking and Abstraction with Partially Ordered Boolean Functional Vectors. In: Alur, R., Peled, D. A. (eds.) Computer-Aided Verification CAV 2004. Lecture Notes in Computer Science, Vol. 3114. Springer-Verlag, Berlin Heidelberg New York (2004) 255-267

11. Gunther, W., Drechsler, R.: Implementation of Read-k-times BDDs on Top of Standard BDD Packages. The 14th International Conference on VLSI Design VLSID '01 (2001) 173
12. López Yáñez, I., Yáñez Márquez, C., Camacho Nieto, O. (2007). "Binary Decision Diagrams", Colección CIDETEC, Instituto Politécnico Nacional, México. ISBN: 978-970-94696-4-6
13. Yáñez-Márquez, C.: Associative Memories Based on Order Relations and Binary Operators (In Spanish). PhD Thesis. Center for Computing Research, México (2002)
14. Kohonen, T.: Self-Organization and Associative Memory. Springer-Verlag, Berlin Heidelberg New York (1989)
15. Hassoun, M. H.: Associative Neural Memories. Oxford University Press, New York (1993)
16. Steinbuch, K.: Die Lermatrix, *Kybernetik*, 1(1) 1961, 26–45.
17. Wilshaw, D., Buneman, O. and Longuet-Higgings, H.: Non-holographic associative memory, Nature 222 (1969), 960–962.
18. Anderson, J. A.: A simple neural network generating an interactive memory, Mathematical Biosciences, 14 (1972), 197–220.
19. Kohonen, T.: Correlation matrix memories, IEEE Transactions on Computers, 21(4) (1972), 353–359.
20. Hopfield, J. J.: Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79 (1982), 2554–2558.
21. Ritter, G.X., Sussner, P., Diaz-de-Leon, J.L.: Morphological Associative Memories. IEEE Transactions on Neural Networks. 9 (1998) 281-293
22. Yáñez-Márquez, C., Díaz de León-Santiago, J.L.: Memorias Asociativas Basadas en Relaciones de Orden y Operaciones Binarias. Computación y Sistemas. México, 6(4). (2003) 300-311
23. Yáñez-Márquez, C., Díaz de León-Santiago, J.L.: Memorias Asociativas con Respuesta Perfecta y Capacidad Infinita. In: Albornoz, A., Alvarado, M. (eds.): Taller de Inteligencia Artificial TAINA'99. México (1999) 245-257
24. Yáñez-Márquez, C., Díaz de León-Santiago, J.L.: Memorias Morfológicas Autoasociativas (In Spanish). Technical Report 58, Green Series. National Polytechnic Institute. Center for Computing Research, México (2001)
25. Andersen, H. R.: An Introduction to Binary Decision Diagrams. Technical University of Denmark (1997)
26. Gröpl, C., Prömel, H. J., Srivastav, A.: Ordered Binary Decision Diagrams and the Shannon Effect. Discrete Applied Mathematics, 142(1-3) (2004) 67-85
27. Bryant, R. E.: Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams. ACM Computing Survey, 24(3) (1992) 293-318
28. Wegener, I.: BDDs - Design, Analysis, Complexity, and Applications. Discrete Applied Mathematics, 138(1-2) (2004) 229-251
29. Kohonen, T.: Correlation Matrix Memories. IEEE Transactions on Computers. 21(4). (1972) 353-359