Decision Diagrams for Discrete Functions: Classification and Unified Interpretation

Radomir S. Stanković

Dept. of Computer Science
Faculty of Electronics
18000 Niš
Yugoslavia

Tsutomu Sasao

Dept. of Computer Science and Electronics
Kyushu Institute of Technology
Iizuka 820
Japan

Abstract— This paper classifies different decision diagrams (DDs) for discrete functions with respect to the domain and range of represented functions. Relationships among different DDs and their relations to spectral transforms are also shown. That provides a unified interpretation of DDs, and their further classification with respect to the spectral transforms.

I. Introduction

Decision diagrams (DDs) are data structures to represent discrete functions. DDs are derived from decision trees (DTs) by reduction of nodes and edges. This concept originates in the representations of discrete sets [17]. Applications of DDs to switching functions [1] and to multiple-valued (MV) functions [49] are well known. The present interest in DDs sprang up after the publication of [4]. A variety of DDs have been proposed to represent different classes of discrete functions by using compact DDs.

We encountered 43 different DDs in the last several years. Their mathematical classification is important and useful for both theoretical and practical applications. We attempt to show the first publication for each DD. When the same or similar DDs were presented by different authors at about the same time, we refer to an easily accessible publication. Suggestions and corrections for this welcomed. To classify different DDs, we focus on the algebraic structures where the represented functions are defined. We refer to the corresponding DTs, since they do not depend on peculiar properties of represented functions.

A. Domain

A discrete function is a mapping $f : D \rightarrow R$, where D is the domain and R is the range. If we have to consider the representation of the mapping only, we need not consider the algebraic structures of D and R. However, since we will consider spectral transformation of the functions, the algebraic structures of D and R are important. The group G has the weakest algebraic structure for D, but still provide a tractable model for f, and is sufficient for many applications. The number of variables, n, in f and the cardinality of the domain, $|D|$, determine a representation of G as the direct product of some subgroups G_i, $i = 1, \ldots, n$. This representation is mapped into the recursive structure of the DTs for f. Structure of a DT is defined by the number of levels and the number of outgoing edges of nodes in the DT. The number of outgoing edges of nodes at the i-th level determine the number of nodes at the $(i + 1)$-th level in the DT. Thus, representation of G determines the structure of DT. Therefore, as the first parameter, we use the domain of the functions to classify DDs. As for the domains, we consider the dyadic, p-adic, arbitrary finite Abelian, and non-Abelian groups. The set of tuples $(x_1, \ldots, x_n)$, $x_i \in P = \{0, \ldots, p - 1\}$ with the componentwise addition modulo p, expresses the structure of a group, which is called the p-adic group. When $p = 2$, i.e., when $P = B = \{0, 1\}$, it is called a dyadic group. If $p > 2$, and p is non-prime, the group operation in G can be defined in some other way that permits extension of the group into a field. Both approaches are used in MV functions.

B. Range

DTs for a given representation of G have the same structure. They differ in the values of constant nodes and decomposition rules by which f is assigned to the DT. Therefore, as the second parameter, we use the range of functions to classify DDs. As for the range, we consider a field F, that may be the field Q of rational numbers, the complex field C, or a Galois field $GF(p)$. The set of integers Z is considered as a subset of Q. Bit-level DDs have logic values, while word-level DDs have values in $GF(p)$, Z, or C. We also consider vector-valued and matrix-valued DDs over $GF(p)$, and over C.

C. Spectral Transforms

Spectral transform DDs (STDDs) [44] uniformly interpret different DDs. This interpretation establishes relationships between decomposition rules applied at the nodes of DTs and spectral transforms. It shows that for each DT, a given f is assigned to the DT through decomposition with respect to a spectral transform. Therefore, DDs classified with respect to the domain and range, are further classified with respect to the spectral transforms.

The transforms used in spectral interpretation of DDs are derived or closely related to the Fourier transforms on finite groups. Recursive structure of DTs corresponds to the recursive block structure of the related transform matrices. Thus, they are defined as the Kronecker product of some basic transform matrices. This Kronecker product corresponds to the direct product of subgroups G_i in representation for G, and in that way, to the structure of the DT. To each constituent subgroup G_i of G a basic transform matrix $\mathbf{Q}_i$ is assigned. Written in the symbolic notation, $\mathbf{Q}_i$ determines the decomposition rule applied at the nodes at the i-th level in the DT [44].

For Abelian groups, each transform matrix defined as the Kronecker product of n basic submatrices, can be factored into the ordinary matrix product of n sparse matrices $\mathbf{C}_i$, $i = 1, \ldots, n$. Each of them performs the spectral transform with respect to a variable in f. Thus, C_i defines the partial spectral transform for f with respect to x_i.

For example, several DDs are related to the arithmetic transform, which can be defined as the Reed-Muller transform with calculations in $GF(2)$ replaced by that in Z. In the symbolic notation, the basic arithmetic transform matrix is defined by the A-expansion $f = 1 \cdot f_0 + x_i(f_1 - f_0)$, where x_i is a switching variable, $f_0 = f(x_i = 0)$, and $f_1 = f(x_i = 1)$. This expansion defines the partial arithmetic transform, which is used in analysis of edge-valued versions of some arithmetic transform related DTs.

The same consideration or relationships between DTs and spectral transforms extends to spectral transforms on non-Abelian groups in terms of the generalized matrix multiplications [48].

II. Classification of DDs by Domains

A. DDs on Dyadic Groups

A.1 Bit-level DDs

Binary DDs (BDDs)[4] represent switching functions $f : B^n \rightarrow B$, where n is the number of input variables. A BDD represent f graphically by recursive application of the Shannon decompositions as a *disjoint sum-of-products expression* [32], [33].

Functional DDs (FDDs) [15] represent f as the *positive polarity Reed-Muller expression* (PPRM). Thus, FDDs are also called *positive polarity Reed-Muller DDs* (PPrmDDs) [33]. FDDs efficiently represent functions for which BDD have exponential size, and vice versa.

Several other DDs are defined by combining the Shannon, the positive Davio and the negative Davio decompositions. They include *fixed polarity Reed-Muller DDs* (FPrmDDs) [33], *Kronecker DDs* (KDDs) [10], and *pseudo-Kronecker DDs* (Pseudo-KDDs) [33]. These DDs represent various AND-EXOR expressions [33], [28], so they are denoted as AND-EXOR related DDs.

Unified DDs (UDDs) [16] represent AND-OR/EXOR expressions, where each node has an operation. Thus, UDDs are DDs with operational nodes.

In *Ternary DDs* (TDDs), each node has three outgoing edges. In TDD-U [14], outgoing edges correspond to logical 0,1, and unspecified value u (don't care). TDD-Us represent incompletely specified switching functions, which formally, can be considered as functions in $GF(3)$.

In other TDDs [31], the first two edges point to the cofactors $f_0 = f(x_i = 0)$ and $f_1 = f(x_i = 1)$, and the third edge points to $f_0 \circledast f_1$. Different choices for the operation $\circledast$ produce different TDDs. EXOR-TDDs [31], *AND-TDDs*, and *Prime-TDDs* (PTDDs) are useful for logic minimization [34], while *Kleene-TDDs* [13] are useful to evaluate switching functions in the presence of unknown inputs. For a given function f, a TDD represents the extended-truth vector $\mathbf{F}_e$ [47]. For a given f, $\mathbf{F}_e$ is determined by the extended Reed-Muller (ERM) transform in terms of $\circledast$ as the addition and the logical AND as the multiplication. The constant nodes in AND-TDDs, EXOR-TDDs, and Prime-TDDs represent logical 0 and 1, while in Kleene-TDDs, they represent 0,1, and u. Therefore, $GF(2)$ is assumed for the range of AND-TDDs, EXOR-TDDs, and Prime-TDDs, while the Kleene algebra is assumed for the range of Kleene-TDDs. Note that the Kleene algebra is not a field.

Shared BDDs (SBDDs) [22] represent multi-output switching functions, while *Zero-suppressed BDDs* (ZBDDs) [23] represent sets of cubes.

The operational edge often reduces the number of nodes in DDs. It assigns an operation to an edge in the DD. The simplest example is the negative edge, that complements the function value. This concept originates in an inverter [1] and typed node [19].

A.2 Word-Level DDs

In many cases, representation of Boolean functions at the integer level denoted in the computer representations as the *word* level is useful. For example, n-variable multiple-output switching functions $(f_0, f_1, f_2 \ldots, f_{m-1})$ are represented by the integer-valued function f whose values at $x = (x_1, \ldots, x_n)$ are

$$f(x) = \sum_{i=0}^{m-1} f_i(x_1, \ldots, x_n)2^i.$$

Multi-terminal binary DDs (MTBDDs) [6] were introduced first to represent Walsh transform coefficients. The constants 0 and 1 at the edges of MTBDDs represent integers 0 and 1, respectively, and the constant nodes represent integers. Therefore, MTBDDs are integer counterparts of BDDs. MTBDDs are also called *arithmetic DDs* [2], and can also represent functions $B^n \to C$.

Shared MTBDDs (SMTBDDs) [36] represent multi-output switching functions. They combine merits of SBDDs and MTBDDs. In SMTBDDs, the constant nodes are binary vectors [12].

In the spectral interpretation, BDDs and MTBDDs are DDs derived from DTs defined with respect to the same basis. This basis corresponds to the block pulse functions, or δ-functions in the classical mathematical analysis, defined by $\delta_y(x) = 1$ for $x = y$, and $\delta_y(x) = 0$, otherwise. Since we are working on finite discrete structures, we denote it as the *trivial basis*. In the matrix notation, the columns of the identity matrices over $GF(2)$ represent the trivial basis for BDDs, and that over Z represent the trivial basis for MTBDDs [44].

In the same interpretation, FDDs are defined through decomposition of f with respect to the Reed-Muller basis over $GF(2)$, represented by columns of the Reed-Muller matrix. *Arithmetic Transform DDs* (ACDDs) are defined with respect to the same basis over the set of integers Z [44]. We denote it as the integer Reed-Muller basis. DTs for *Binary moment diagrams* (BMDs) [5] are defined with respect to the same basis, but are derived by using different reduction rules. The BDD and zero-suppressed BDD reduction rules [24] are applied to ACDDs and BMDs, respectively. These rules are defined for the bit-level DDs. However, their use in ACDDs and BMDs is possible, since the labels at the edges in these DDs are 1 and x_i. A distinction of both constant 1 and the variable $x_i \in \{0,1\}$ over $GF(2)$ and over Z is just formal in this case. ACDDs and BMDs are integer counterparts of FDDs. ACDDs and BMDs represent f in the form of the arithmetic expression, which is the integer counterpart of the PPRM. In this setting, *the Walsh transform DDs* (WDDs) [44] are DDs derived with respect to the integer Reed-Muller basis, but in $(0,1) \to (1,-1)$ coding of switching variables. WDDs represent f in the form of the Walsh expression for f, which is a particular example of the Fourier series expansions on finite groups. The integer counterparts of KDDs are *Kronecker BMDs* (KBMDs) [8], [9]. They represent f in the form of integer counterparts of *Kronecker* (KRO) expressions [30], [33].

Hybrid DDs (HDDs) [7] use the complex-valued matrices of order 2 and 4 in the decompositions to derive the DTs. Elements of these matrices are taken from the set $\{0, 1, -1, i, -i, 1+i, 1-i, i-1, -i-1\}$. *Complex Hadamard DDs* (CHDDs) and their edge-valued versions (EVCHDDs) are defined with respect to the complex Hadamard transform (CHT) [11]. HDDs, CHDDs, and EVCHTDDs represent f in the form of Fourier series-like expressions in terms of the corresponding transforms over C.

Spectral Transform DDs (STDDs) [44] are a generalized concept of these DDs. STDDs permit spectral interpretation of DDs and, thus, involve various DDs as examples defined with respect to various bases. From STDDs, different DDs are defined by specifying particular bases.

A.3 Edge-Valued DDs

Edge-Valued Binary DDs (EVBDDs) [18] represent integer-valued functions $B^n \to Z$. Similar to MTBDDs, EVBDDs are derived from DTs defined in terms of the trivial basis. However, in EVBDDs the constant nodes are set to zero, and instead of them additive weights at the edges are introduced. Subtraction of the values of constant nodes, and the recursive structure of DTs, determine that weights at the edges must be equal to the coefficients of the corresponding partial arithmetic transforms [39]. Therefore, EVBDDs relate to ACDDs, since similar to ACDDs, EVBDDs represent f in the form of the arithmetic expression. However, in ACDDs, decomposition of f is performed by using partial arithmetic transforms recursively, level by level over the DT. In EVBDDs, to determine the weights at the edges for each level the partial arithmetic transforms are applied each time to f. EVBDDs reduce to particular pseudo-Kronecker DDs over Z, and can be further generalized into a family of *DDs with attributed edges*.

The demerit of EVBDDs is that they require integer weights at the edges to represent single-output switching functions. Spectral interpretation of EVBDDs introduced *Edge-Valued Functional DDs* (EFDDs) defined in terms of the partial Reed-Muller transforms [40]. EVFDDs for switching functions have weights of logical values at the edges.

Edge-valued version of BMDs were provided as $*$BMDs [5]. The arithmetic transform coefficients in constant nodes of BMDs are factored, and common factors are used as multiplicative weights at the edges in $*$BMDs.

Factored Edge-Valued Binary DDs (FEVBDDs) are a generalization of EVBDDs by permitting additive and multiplicative weights at the edges [50]. Both additive and multiplicative weights at the edges connecting non-terminal nodes are determined as combinations of partial arithmetic transform coefficients. At the edges connecting non-terminal nodes with constant nodes, the additive weights are taken as 1 and the multiplicative weights as 0. Thus, FEVBDDs are EVBDDs with normalized weights at the edges towards the constant nodes. Similar to ACDDs and BMDs, EVBDDs, FEVBDDs and $*$BMDs represent f in the form of the arithmetic expression.

Similar to FEVBDDs, *K$*$BMDs* have both additive and multiplicative factors derived from $*$BMDs [8]. However, they combine properties of KBMDs and $*$BMDs, since the basic DT of K$*$BMDs is the Kronecker tree.

B. DDs on p-adic Groups

Multiple-place DDs (MDDs) [49] represent multiple-valued (MV) functions $f : P^n \to P$. An MDD for a function f has nodes with p outgoing edges, and MDD is defined on the p-adic groups. Similar to BDDs, MDDs are defined in terms of the trivial basis on p-adic groups. Thus, MDDs represent f in the form of sum-of-product expressions for MV functions [29].

Multi-Terminal Multiple-Valued DDs (MTMDDs) and *Shared Multi-Terminal Multiple-Valued DDs* (SMTMDDs) are extensions of MTBDDs and SMTBDDs permitting binary vectors as constant nodes. These vectors are binary representations of integers in the constant nodes in MTBDDs and SMTBDDs [36].

If the elements of $GF(p)$ are interpreted as the first p non-negative integers of C, then MDDs can represent functions $P^n \rightarrow C$ [20], [43]. In this case, MDDs are generalized by allowing complex-numbers for the constant nodes. They are denoted as *Multi-terminal DDs* (MTDDs). To point out the resemblance with FFT, the MDDs with operational edges are suggested in calculation of Vilenkin-Chrestenson transforms [43]. To represent spectral coefficients of MV functions, the complex Moraga's coding [25] for MV functions is suggested [20]. Thanks to this coding, constant nodes in MDDs for Vilenkin-Chrestenson spectrum of MV functions represent first p non-negative integers [20].

BDDs are often optimized when the symmetric variables are kept adjacent [26]. Multiple-Valued DDs with Symmetric Variables (MDD-SV) [21] are a generalization of MDDs that exploit this property for MV functions. Unlike other DDs on Abelian groups, MDD-SVs reduce the number of levels in MDDs, in addition to the number of nodes.

The depth in DDs for switching functions is also reduced by the *Quaternary DDs* (QDDs) [35] that are introduced by pairing binary-valued variables to make four-valued variables. In the word of group theory, dyadic groups C_2^n consisting of products of n cyclic groups of order 2, are replaced by the groups C_4^n consisting of $n/2$ cyclic groups of order 4. In the spectral interpretation, QDDs are extensions of AND-EXOR related DDs to groups of order 4. Design of QDDs and Pseudo-QDDs [35] uses the linearly independent bases for switching functions over $GF(2)$ [27]. Similar as MDDs, under an appropriate interpretation of labels at the edges, these DDs can represent functions $f : C_4^n \rightarrow C$.

Extensions of FDDs to MV functions are *the Galois field DDs* (GFDDs) and *Reed-Muller-Fourier DDs* (rmFDDs) [38]. MDDs, QDDs, and Pseudo-QDDs are DDs on p-adic groups, where the decomposition rules are described by matrices with binary elements. On the other hand, in GFDDs and rmFDDs, the decomposition rules are described by matrices with p-valued elements. Edge valued versions of these DDs, *Edge-Valued Galois field DDs* (EVGFDDs) [41] and *Edge-Valued Reed-Muller-Fourier DDs* (EVrmFDDs) [45] are DDs derived from the corresponding partial transforms for MV functions. *Kronecker Galois field DDs* (KGFDDs) [42] are defined through a generalization of KDDs to MV functions in Galois fields. These DDs represent f in the form of corresponding generalizations of AND-EXOR expressions of MV functions [29].

C. DDs on Arbitrary Abelian Groups

Generalization of DDs for functions on arbitrary Abelian groups into an arbitrary field assumes that the domain group G of order $|G| = g$ is represented as

$$G = \times_{i=1}^{n} G_i, \quad g = g_1 \cdots g_i \cdots g_n, \tag{1}$$

where $|G_i| = g_i$, $i = 1, \ldots, n$, is the order of G_i. In such DDs, decomposition rules in nodes at the i-th level can be chosen among all possible non-singular matrices of orders g_i. For example, if we use the identity matrices, the MTDDs on arbitrary Abelian groups are defined.

Function Graphs (FGs) [3] represent multiple-output MV functions with variables taking values in finite sets of arbitrary cardinalities. The nodes are particular generalizations of Shannon and Davio nodes over $GF(p)$. The columns of the basic transform matrices used in FGs relate to the characteristic functions in $GF(p)$.

For some functions, MTDDs for the Fourier spectrum are smaller than the MTDDs for original functions [20]. Therefore, *Fourier DDs on Abelian groups* (FADDs) defined in terms of group characters [46] efficiently represent such functions. Note that a function $\chi_w : G \rightarrow C$ is the *group character* for G iff $\|\chi_w(x)\| = 1$, $\forall x \in G$, and $\chi_w(x \circ y) = \chi_w(x)\chi_w(y)$, $\forall x, y \in G$, where $\circ$ denotes the group operation in G, and $\|x + iy\| = \sqrt{x^2 + y^2}$. WDDs are the example of FADDs on the dyadic groups, since the discrete Walsh functions are group characters of these groups. FADDs on p-adic groups are defined in terms of the Vilenkin-Chrestenson functions, which are the group characters of p-adic groups. FNADDs represent f in the form of the Fourier series expansions for f.

D. DDs on non-Abelian Groups

If the domain group G is an Abelian group represented as in (1), then the depth of the DT is equal n. The nodes at the i-th level have g_i outgoing edges, and g_i determines the number of non-terminal nodes at the $(i+1)$-th level in the DT. Therefore, the reduction of depth by using Abelian subgroups of large orders g_i tends to increases the width of the DD.

Reduction of depth and width of DDs at the same time is achieved in *Fourier DDs with non-Abelian groups* (FNADDs) as the domain for f [46]. Decomposition rules for these DDs are derived from the Fourier transform on groups, and are defined in terms of the group characters for Abelian and unitary irreducible representations for non-Abelian groups [46]. Therefore, Fourier spectrum on non-Abelian groups is a matrix-valued function defined on a set whose cardinality is always smaller than $|G|$. This permits reduction of the width of the DT comparing to the DTs on Abelian groups. Constant nodes in FNADDs are Fourier coefficients, and some of them are matrices. Therefore, FNADDs are matrix-valued DDs (mvFNADDs). If in FNADDs, the matrix-valued nodes are represented by DDs, that may be again Fourier DDs, or MTDDs, or any other DD, then the number-valued FNADDs (nvFNADDs) are derived. As FADDs, FNADDs represent f in the form of the Fourier series expansion for f on the non-Abelian groups.

FNADDs reduce the number of non-terminal nodes at the price of increased number of constant nodes and take advantages in matrix representations in spectral domain. Note that group representations can be defined over finite fields $GF(p)$ providing that some relationships between G and $GF(p)$ are satisfied. In this case, the number of different constant nodes in FNADDs is at the most p. *Fourier DDs on finite non-Abelian groups with Preprocessing* (FNAPDDs) [46] are an extension of FNADDs to matrix-valued functions. Thus, FNADDs are the matrix-valued DDs (mvFNAPDDs). Unlike mvFNADDs, in mvFNAPDDs, all the constant nodes are matrices. In application to number-valued functions, they take advantages in the matrix notation in both original and spectral domain. The number-valued versions of FNAPDDs (nvF-NAPDDs) are defined by representation of matrix-valued constant nodes by DDs of small sizes.

III. Classification of DDs by Ranges

Table 1 classifies DDs with respect to the range of the represented functions. These DDs have numbers, binary vectors, and matrices in constant nodes. In TDD-Us, a constant node may have value u, which represents *don't care* in incompletely specified switching functions. Fourier DDs on Abelian or non-Abelian groups may be defined over $GF(p)$ or over C.

Fig. 1 is the DD that classifies various DDs with respect to both domain G and the range F of the represented functions.

IV. Unified Interpretation of DDs

Previous discussion and analysis of DTs briefly summarizes spectral interpretation of DDs. It shows that each DT is based on a spectral transform. That means, a given f is assigned to a DT by the decomposition with respect to a spectral transform. In a DT, the values of constant nodes are the spectral transform coefficients. Therefore, DDs represent f in the form of spectral transform expressions. In the case of WDDs, and Fourier DDs, these are Fourier series expansions. In other cases, they are Fourier series-like expansions for f. The identical mapping is involved as a trivial example of spectral transforms.

The weights in edge-valued DDs are the partial spectral transform coefficients, or are determined by the factorization of spectral transform coefficients. However, in both cases the edge-valued DDs represents f in the form of spectral transform expressions for f.

Thanks to this unified interpretation, DDs classified with respect to the domain and range are further classified with respect to the spectral transforms.

V. Classification of DDs by Spectral Transforms

Table 2 shows spectral transforms used to define various DDs. The basic functions for DDs given in the parenthesis are not directly equal to a particular transform, but are derived from it or are closely related to it. The basic functions for pseudo-QDDs are defined through (4×4) non-singular binary-valued matrices. The basic functions for FGs are derived through particular generalizations of the Shannon and Davio nodes to MV functions. These functions are expressed through the characteristic functions in the set theory.

VI. Closing Remarks

In this paper, we classified various DDs for discrete functions, first with respect to the domain of the represented functions, and second with respect to the range of the functions. DDs on the same domain are derived from the DTs of the equal structures, but can be different with respect to the decomposition rules as well as the range for the represented functions. We considered integer and complex-valued counterparts of DDs for switching and MV functions, respectively. Furthermore, we classified DDs with respect to spectral transforms used in definition of the corresponding DTs.

Acknowledgments

Part of this work was done during R. S. Stanković's stay in Kyushu Institute of Technology with the support of KIT fellowship. T. Sasao was supported by a Grant in Aid for Scientific Research of the Ministry of Education, Science, Culture, and Sports of Japan.

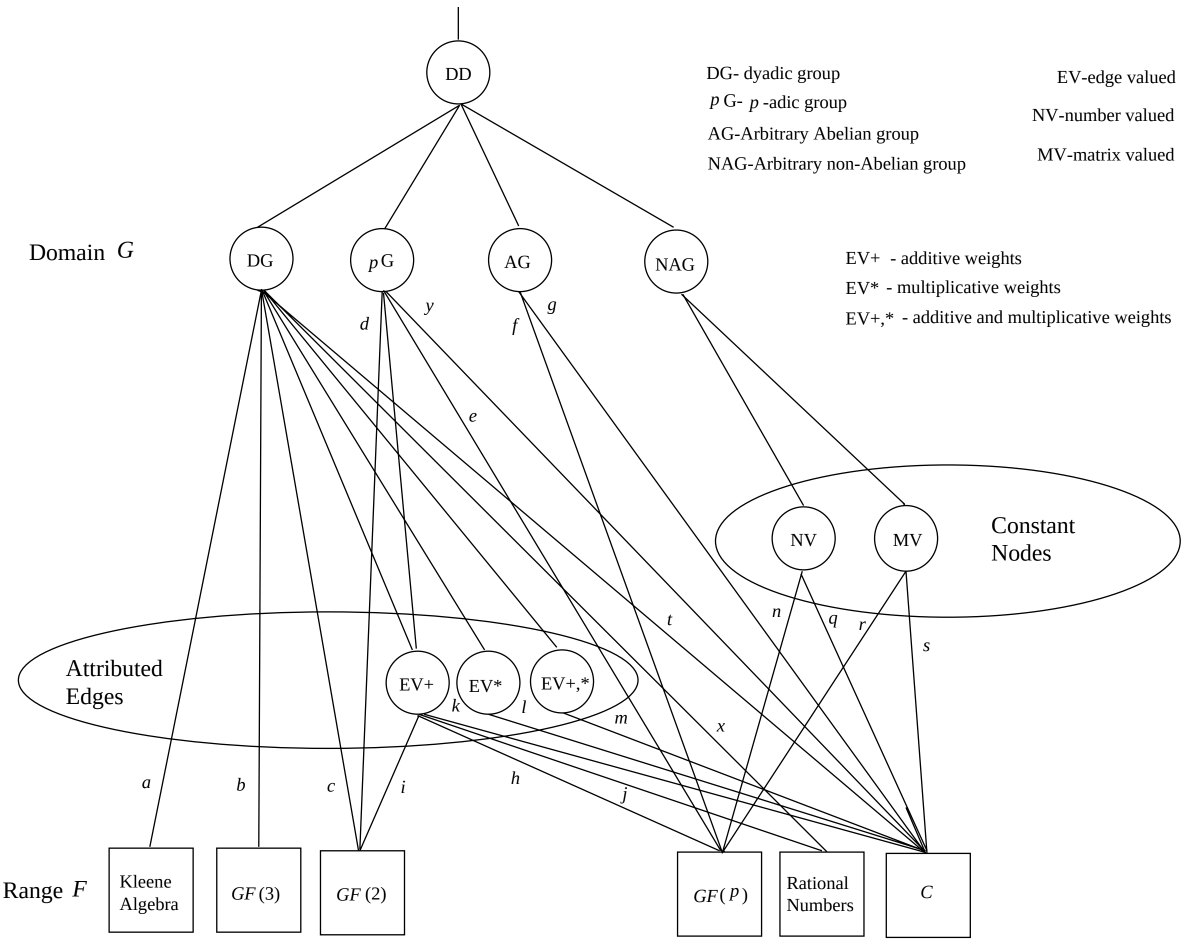
References

[1] S.B. Akers, "Binary decision diagrams," *IEEE Trans. on Computers*, Vol. C-27, No. 6, 1978, 509-516.

[2] R.I. Bahar, E.A. Frohm, C.M. Gaona, G.D. Hachtel, E. Macii, A. Pardo, and F. Somenzi, "Algebraic decision diagrams and their applications," *ICCAD*, November 1993, 188-191.

[3] B. Becker and R. Drechsler, "Efficient graph based representations of multi-valued functions with an application to genetic algorithm," *ISMVL-24*, 1994, 65-72.

[4] R.E. Bryant, "Graph-based algorithms for Boolean functions manipulation," *IEEE Trans. Comput.*, Vol.C-35, No.8, 1986, 667-691.



a = Kleene TDD

b =TDD-U

c =BDD, FDD=PPrmDD, FPrmDD, Pseudo-rmDD, KDD, Pseudo-KDD, UDD, AND-TDD, EXOR-TDD, Prime-TDD

d =QDD, Pseudo-QDD

e =MDD, MDD-SV, GFDD, rmFDD

f =MTDD, FG

g =MTDD

h =EVFDD

i =EVGFDD, EVrmFDD

j =EVBDD

k =EVCHDD

l =*BMD

m =FEVBDD, K*BMD

n =nvFNADD, nvFNAPDD

q =nvFNADD, nvFNAPDD

r =mvFNADD, mvFNAPDD

s =mvFNADD, mvFNAPDD

t =MTBDD, CHDD, HDD

x =MTBDD, ACDD, BMD, KBMD, HDD

y =MDD

Fig. 1. Classification of DDs for discrete functions

[5] R.E. Bryant and Y-A. Chen, "Verification of arithmetic functions with binary moment diagrams," May 31, 1994, CMU-CS-94-160.

[6] E.M. Clarke, K.L. McMillan, X. Zhao, M. Fujita, and J. Yang "Spectral transforms for large Boolean functions with application to technology mapping," *DAC-1993*, June 1993.

[7] E.M. Clarke, M. Fujita, and X. Zhao, "Multi-terminal decision diagrams and hybrid decision diagrams", in: [37], 93-108.

[8] R. Drechsler and B. Becker, "OKFDDs-algorithms, applications and extensions," in: [37], 163-190.

[9] R. Drechsler and B. Becker, "Overview of decision diagrams", *IEE Proc.-Comput. Digit. Tech.*, Vol. 144, No. 3, May 1997, 187-193.

[10] R. Drechsler, A. Sarabi, M. Theobald, B. Becker, and M.A. Perkowski "Efficient representations and manipulation of switching functions based on Ordered Kronecker functional decision diagrams," *DAC-1994*, June 1994, 415-419.

[11] B.J. Falkowski and S. Rahardja, "Complex spectral decision diagrams," *ISMVL-26*, 1996, 255-260.

[12] H.Md. Hasan Babu and T. Sasao, "A method to represent multiple-output switching functions by using binary

decision diagrams," SASIMI'96, Nov. 25, 1996, 212-217.

[13] Y. Iguchi, T. Sasao, and M. Matsuura, "On properties of Kleene TDDs," *ASP-DAC'97*, 473-476.

[14] G. Jennings, "Symbolic incompletely specified functions for correct evaluation in the presence of indeterminate input variables", *Twenty-Eight Annual Hawaii Int. Conf. on System Sciences, HICSS-28*, Vol. I, *Architecture*, January 1995, 23-31.

[15] U. Kebschull, E. Schubert, and W. Rosenstiel, "Multilevel logic synthesis based on functional decision diagrams," *EDAC 92*, 1992, 43-47.

[16] T. Kozlowski, E.L. Dagless, and J.M. Saul, "Unified decision diagrams, a representation for mixed AND-OR/EXOR combinational networks," *Reed-Muller'95*, 177-184.

[17] Dj.R. Kurepa, *Ensembles ordonnés et ramifiès*, (Thèse), Paris, 1935, 6+138+2p.

[18] Y-T. Lai and S. Sastry, "Edge-valued binary decision diagrams for multi-level hierarchical verification," *DAC-92*, June 1992, 668-613.

[19] J.C. Madre and J.P. Billon, "Proving circuit characteristics using formal comparison between expected and extracted behaviour," *DAC-1988*, June 1988, 205-210.

[20] D.M. Miller, "Spectral transformation of multiple-valued decision diagrams," *ISMVL-24*, May 1994, 89-96.

[21] D.M. Miller and N. Muranaka, "Multiple-valued decision diagrams with symmetric variable nodes," *ISMVL-26*, May 1996, 242-247.

[22] S. Minato, N. Ishiura, and S. Yajima, "Shared binary decision diagrams with attributed edges for efficient Boolean functions manipulation," *Proc. DAC-1990*, June 1990, 52-57.

[23] S. Minato, "Zero-suppressed BDDs for set manipulation in combinatorial problems," *DAC-1993*, June 1993, 272-277.

[24] S. Minato, "Graph-based representations of discrete functions", in: [37], 1-28.

[25] C. Moraga, "Complex spectral logic," *ISMVL-8*, May 1978, 149-157.

[26] S. Panda and F. Somenzi, "Who are the variables in your neighborhood," *ICCAD-95*, 1995, 74-77.

[27] M.A. Perkowski, "The generalized orthonormal expansions of functions with multiple-valued inputs and some of its applications," *ISMVL-22*, May 1992, 442-450.

[28] T. Sasao, "AND-EXOR expressions and their optimizations," in: [37], 287-312.

[29] T. Sasao, "Logic synthesis with EXOR gates," in: T. Sasao, (ed.), *Logic Synthesis and Optimization*, Kluwer Academic Publishers, 1993, 259-285.

[30] T. Sasao, "Optimization of pseudo-Kronecker expressions using multiple-place decision diagrams," *IEICE Trans. Inf. and Syst.*, Vol. E-76-D, No. 5, 1993, 562-570.

[31] T. Sasao, "Ternary decision diagrams and their applications," *ISMVL-97*, May 1997, 241-250.

[32] T. Sasao and D. Debnath, "An exact minimization algorithm for generalized Reed-Muller expressions," *Proc. IEEE Asia-Pacific Conference on Circuits and Systems*, 5.-8.12.1994, Taipei, Taiwan.

[33] T. Sasao, "Representations of logic functions by using EXOR operators," in: [37], 29-54.

[34] T. Sasao, "Ternary decision diagrams and their applications," in: [37], 269-292.

[35] T. Sasao and J.T. Butler, "A design method for look-up table type FPGA by pseudo-Kronecker expansions," *ISMVL-24*, May 1994, 97-104.

[36] T. Sasao and J.T. Butler, "A method to represent multiple-output functions by using multi-valued decision diagrams," *ISMVL-26*, May 1996, 248-254.

[37] T. Sasao and M. Fujita, (ed.), *Representations of Discrete Functions*, Kluwer Academic Publishers, 1996.

[38] R.S. Stanković, "Functional decision diagrams for multiple-valued functions," *ISMVL-25*, May 1995, 284-289.

[39] R.S. Stanković, "Some remarks about spectral transform interpretation of MTBDDs and EVBDDs," *ASP-DAC'95*, 1995, 385-390.

[40] R.S. Stanković and C. Moraga, "Edge-valued decision diagrams for switching functions based upon the partial Reed-Muller transforms," *Proc. 2nd Int. Conf. Application of Computer Systems*, Szczecin, Poland, 8-9.1995, 7-16.

[41] R.S. Stankovć and C. Moraga, "Edge-valued decision diagrams for multiple-valued functions," *Proc. 3rd Int. Conf. Application of Computer Systems*, Szczecin, Poland, November, 21-22, 1996.

[42] R.S. Stanković and R. Dreschler, "Circuit design form Kronecker Galois field decision diagrams for multiple-valued functions," *ISMVL-27*, May 1997, 275-280.

[43] R.S. Stanković, M. Stanković, C. Moraga, and T. Sasao, "Calculation of Vilenkin-Chrestenson transform coefficients of multiple-valued functions through multiple-place decision diagrams," *Proc. 5th Int. Workshop on Spectral Techniques*, March 15-17, 1994, Beijing, China, 107-116.

[44] R.S. Stanković, T. Sasao, and C. Moraga, "Spectral transform decision diagrams," in: [37], 55-92.

[45] R.S. Stanković, "Edge-valued decision diagrams based on partial Reed-Muller transforms," *Proc. Reed-Muller Colloquium UK'95*, Bristol, England, UK, December 19, 1995, 9/1-9/13.

[46] R.S. Stanković, "Fourier decision diagrams on finite non-Abelian groups with preprocessing," *ISMVL-27*, May 1997, 281-286.

[47] R.S. Stanković and T. Sasao, "Spectral Interpretation of TDDs," *SASIMI'97*, 45-50.

[48] R.S. Stanković, "Fast algorithms for calculation of Gibbs derivatives on finite groups," *Approx. Theory and Its Applications,* 7, 2, June 1991, 1-19.

[49] A. Thayse, M. Davio, and J.-P. Deschamps, "Optimization of multiple-valued decision diagrams," *ISMVL-8*, 1978, 171-177.

[50] S.B.K. Vrudhula, M. Pedram, and Y.-T. Lai, "Edge valued binary decision diagrams," in: [37], 109-132.

In the bibliography we use *ISMVL-XX* = *Proc. the XX-th Int. Symp. on Multiple-Valued Logic*

TABLE I

Classification of DDs by range

Range	Decision Diagram
$\{0,1\}$	BDD, FDD=PPrmDD, FPrmDD, Pseudo-rmDD, KDD, Pseudo-KDD, UDD, EVFDD, EXOR-TDD, AND-TDD, Prime-TDD
$\{0,1,u\}$	TDD-U, Kleene-TDD
Binary vectors	MTMDD, SMTMDD
$\{0,\ldots,p-1\}$	MDD, FG, QDD, GFDD, RMFDD, KGFDD
Rational numbers	MTBDD, BMD, KBMDD, ACDD, WDD, EVBDD, FEVBDD, $*$BMD, K$*$BMD,
Complex numbers	HDD, CHDD, FADD, nvFNADD, nvFNAPDD
Matrix over C or $GF(p)$	FNADD, FNAPDD

TABLE II

Discrete transforms used in definition of DDs

Domain	Range	Transform	Decision Diagram
Dyadic group	$GF(2)$	Identical	BDD,
		Reed-Muller	FDD=PPrmDD, FPrmDD, Pseudo-rmDD
		Partial Reed-Muller	EVFDD
		Identical and Reed-Muller	KDD, PKDD
		ERM	AND-TDD, EXOR-TDD, Prime-TDD
	$GF(3)$	Identical	TDD-U
	Kleene algebra	ERM	Kleene-TDD
	Rational numbers	Identical	MTBDD
		Arithmetic	BMD, ACDD, $*$BMD,
		Partial Arithmetic	EVBDD, FEVBDD
		Walsh	WDD
		Kronecker	KBMDD K$*$BMD
	C	CHT	CHDD, (HDD), EVCHDD
p-adic	$GF(p)$	Identical	MDD, MDD-SV, MTMDD, QDD
		GF-transform	GFDD, (Pseudo-QDD)
		RMF-transform	rmFDD
		Parial GF	EVGFDD
		Partial RMF	EVRMFDD
Arbitrary Abelian group	C or $GF(p)$	Identical	MTDD, (FG)
		Fourier	FADD
Arbitrary Non-Abelian group	Matrices over C or $GF(p)$	Fourier for number-valued functions	FNADD
		Fourier for matrix-valued functions	FNAPDD