Approximation of non-Boolean functions by Boolean functions and applications in non-standard computing

Dan A. Simovici
University of Massachusetts Boston
Computer Science Department
Boston, Massachusetts 02125, USA
dsim@cs.umb.edu

Abstract

We survey the research and we report new results related to the relationships that exist between Boolean and non-Boolean functions defined on Boolean algebras. The results included here are relevant for set-valued logic that is useful in several non-standard types of circuits: interconnection-free biomolecular devices, devices based on optical wavelength multiplexing, etc. We extend our previous results on approximation of single-argument non-Boolean functions to multi-argument functions.

1. Introduction

Let $\mathcal{B} = (B, \vee, \cdot, ', 0, 1)$ be a Boolean algebra, where B is a set, $\vee$ and $\cdot$ are binary operations on B called disjunction and conjunctions, respectively, $'$ is a unary operation, called the complementation operation, and $0, 1$ are two special elements of B, with $0 \neq 1$ such that the usual axioms of Boolean algebras are satisfied as given in [15] or [6].

Boolean functions are those functions $f : B^n \longrightarrow B$ for some Boolean algebra $\mathcal{B} = (B, \vee, \cdot, ', 0, 1)$ that are obtained from variables and constants by successive applications of the operations of Boolean algebras: $\vee, \cdot$, and $'$. A Boolean function is completely determined by its values on the vectors $A \in \{0, 1\}^n$. Therefore, there are 2^{2^n} Boolean functions of the form $f : B^n \longrightarrow B$. If $\mathcal{B}$ is a finite Boolean algebra that has m atoms, the number of functions $F : B^n \longrightarrow B$ is $2^{m2^{mn}}$ may be considerably larger even for small values of m. In case of a two-element Boolean algebras, these number coincide. In larger algebras, the fraction of Boolean functions becomes minuscule: the Boolean functions represent $\frac{1}{2^{m2^{mn}-2^n}}$ of all functions.

Research in new computing paradigms that originates mainly in Japan brought to the forefront the use of non-Boolean functions over Boolean algebras in implementing new types of circuitry. Bio-switching devices introduced and studied in [7] use the specificity of the reaction between enzimes and substrata in order to compute multi-valued switching functions. This kind of circuitry allows ultra-high-valued data processing and a high degree of computing parallelism. In [2, 7] the authors consider several fundamental bio-circuits which correspond to the *bio-pass*, *bio-output* and *bio- complement*. If $\mathbf{r} = \{0, 1, \ldots, r-1\}$ is the set of fundamental values of an r-valued logic then every j, $0 \leq j \leq r-1$ represents a pair substratum-enzyme (assuming that we deal with r distinct enzymes).

The bio-circuits mentioned above operate on the set of subsets of $\mathbf{r}$, denoted as usual by $\mathcal{P}(\mathbf{r})$ and, therefore, can be described as functions of the form $F : \mathcal{P}(\mathbf{r})^n \longrightarrow \mathcal{P}(\mathbf{r})$ over the Boolean algebra $(\mathcal{P}(\mathbf{r}), \cup, \cap, ', \emptyset, \mathbf{r})$. A point made in [2] is the fact that these circuits allow the synthesis of any set-valued function, $F : (\mathcal{P}(\mathbf{r}))^n \longrightarrow \mathcal{P}(\mathbf{r})$.

Non-Boolean functions play in important role in other novel idea in circuit design: the use of optical wavelength multiplexing for designing a set-logic network [8, 18], where the conversion gate operation is described a non-Boolean function.

For a Boolean algebra $\mathcal{B} = (B, \vee, \cdot, ', 0, 1)$ we shall denote the elements of B^n by capital letters $X, Y, \ldots$, while elements of the algebra B will be denoted with small letters. Boolean functions will be denoted with small letters $f, g, \ldots$. Arbitrary functions will be denoted by capital letters: $F, G, \ldots$.

If $x \in B$ and $a \in \{0, 1\}$ we use the notations $x^1 = x$ and $x^0 = x'$ for $x \in B$. Note that $a^a = 1$ for $a \in \{0, 1\}$. If $X = (x_1, \ldots, x_n) \in B^n$ and $A = (a_1, \ldots, a_n) \in \{0, 1\}^n$, define X^A as $x_1^{a_1} \cdots x_n^{a_n}$. We have $A^C = 1$ if $A = C$ and $A^C = 0$, otherwise for every $A, C \in \{0, 1\}^n$. Also, if $A \neq C$, we have $X^A X^C = 0$ for every $X \in B^n$.

The binary operation "+" is defined on B by $x + y = xy' \vee x'y$ for $x, y \in B$. An easy argument by induction on n shows that if $z_1, \ldots, z_n \in B$ such that $z_i z_j = 0$ for

$1 \leq i, j \leq n$ and $i \neq j$, then $\bigvee_{1 \leq i \leq n} z_i = \sum_{1 \leq i \leq n} z_i$.

If $X = (x_1, \ldots, x_n) \in B^n$, then we denote $x_1 \vee x_2 \vee \cdots \vee x_n$ by $X^{\vee}$.

2 Characterizations of Boolean Functions

Characterizations of Boolean functions allow us to identify the few Boolean functions that exist among the vast number of Boolean functions. Research in this problem is quite old: indeed, the first such characterization was obtained by McColl [9, 10, 11], who proved that a function $F : B^n \longrightarrow B$ is Boolean iff it satisfies 2^n conditions of the form $X^A F(X) = X^A F(A)$, where $A \in \{0,1\}^n$. More recently, in [16], we obtained the following result:

Theorem 2.1 *Let $\mathcal{B} = (B, \vee, \cdot, ', 0, 1)$ be a Boolean algebra and let $F : B^n \longrightarrow B$ be a function. The following statements are equivalent:*

(i) *F is a Boolean function;*

(ii) $F(X) + F(Y) \leq 1 + \bigvee_{A \in \{0,1\}^n} X^A Y^A$ *for every $X, Y \in B^n$;*

(iii) $F(X) + F(Y) \leq (X + Y)^{\vee}$ *for every $X, Y \in B^n$.*

The necessity of the last condition was observed by McKinsey in [12]. Our result shows that this condition is also sufficient and this, it is actually a characterization of Boolean functions.

Example 2.2 The bio-output function $\mathsf{bo} : B^2 \longrightarrow B$ on a Boolean algebra $\mathcal{B} = (B, \vee, \cdot, ', 0, 1)$ was introduced in [7, 2] is given by

$$\mathsf{bo}(x, y) = \begin{cases} x & \text{if } y = 0 \\ 0 & \text{otherwise,} \end{cases}$$

is not Boolean. Indeed, let $x \neq 0$ be an element of B. Observe that $\mathsf{bo}(1,1) + \mathsf{bo}(x,0) = 0 + x = x$, while $((1,1) + (x,0))^{\vee} = (x', 1)^{\vee} = x'$, and $x \not\leq x'$, which contradicts the third statement of Theorem 2.1. ∎

Another important example is related to the set-theoretical literals that are used in [1, 5, 4, 3] designing interconnection-free biomolecular computing systems.

Example 2.3 Let $p, q \in B$ such that $p \leq q$. The set-theoretical literal ${}^p x^q$ is a function from B to B given by:

$${}^p x^q = \begin{cases} 1 & \text{if } p \leq x \leq q \\ 0 & \text{otherwise,} \end{cases}$$

for every $x \in B$. Suppose that $p \neq 0$ and $p < q$. Note that $r = q'$ does not belong to the interval $[p, q]$ because this would imply $p = 0$. Thus, we have ${}^p p^q + {}^p r^q = 1 + 0 = 1$; on another hand. $p + r = p + q' < 1$ because $p + q' = 1$ would imply $p = q$. This contradicts the third part of Theorem 2.1. Thus, ${}^p x^q$ is not a Boolean function, in general. ∎

Example 2.4 Let $\mathcal{B} = (B, \vee, \cdot, ', 0, 1)$ be a finite Boolean algebra, whose set of atoms is denoted by $\mathsf{AT}(B)$ and let $\sigma : \mathsf{AT}(B) \longrightarrow B$ be a function. The σ-conversion function is a unary operation $C_\sigma : B \longrightarrow B$ where $C_\sigma(x) = \bigvee \{\sigma(a) | a \in \mathsf{AT}(B), a \leq x\}$. If σ is chosen, for example, such that $a \not\leq \sigma(a)$ for $a \in \mathsf{AT}(B)$, it is easy to see that x^σ violates the conditions of Theorem 2.1. ∎

The relaxation of the second condition of Theorem 2.1 suggests the introduction of another class of non-Boolean functions that extend to the case of functions of n arguments the classes of upper and lower semi-Boolean functions of one-variable introduced in [13].

Definition 2.5 Let $\mathcal{B} = (B, \vee, \cdot, ', 0, 1)$ be a Boolean algebra and let $F : B^n \longrightarrow B$ be a function. For $A \in \{0,1\}^n$, F is an A-Boolean function if $F(X) + F(Y) \leq 1 + X^A Y^A$ for every $X, Y \in B^n$. ∎

Note that if F is A-Boolean for every $A \in \{0,1\}^n$, then F is a Boolean function. For $n = 1$, every 1-Boolean function is an upper semi-Boolean function and every 0-Boolean functions is a lower semi-Boolean function. Further, it is possible to show that a function $F : B^n \longrightarrow B$ is A-Boolean if and only if it satisfies the McColl's condition for A, namely $X^A F(X) = X^A F(A)$ for every $X \in B^n$.

For any function $F : B^n \longrightarrow B$ the function $G : B^n \longrightarrow B$ defined by $G(X) = X^A F(X)$ for $X \in B^n$ is an C-Boolean function for every $C \neq A$ because $X^C G(X) = X^C X^A F(X) = 0$ and $X^C G(C) = C^A F(C) = 0$.

It is easy to see that if $F : B^n \longrightarrow B$ is an A-Boolean function, then we have:

$$F(X) = X^A F(A) + F(X) \sum_{D \in \{0,1\}^n - \{A\}} X^D \quad (1)$$

for every $X \in B^n$. Therefore, if F is A-Boolean, then $G(X) = X^A F(X)$ is a Boolean function. Moreover, a function $F : B^n \longrightarrow B$ is A-Boolean if and only if there is an element $k \in B$ and a function $K : B^n \longrightarrow B$ such that

$$F(X) = X^A k + K(X) \sum_{D \in \{0,1\}^n - \{A\}} X^D$$

for $X \in B^n$. F is a Boolean function if and only if K is a D-Boolean function for every $D \neq A$.

Example 2.6 We have shown in [17] that the Boolean operations together with the function $\beta : B \longrightarrow B$ defined by

$$\beta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

for $x \in B$, form a complete set of functions for the set of functions defined over B. Let β_n be the generalization of β defined by

$$\beta_n(X) = \begin{cases} 1 & \text{if } X = (0, \ldots, 0) \\ 0 & \text{otherwise} \end{cases}$$

for $X \in B^n$. The function β_n is not Boolean because it is not $(0, \ldots, 0)$-Boolean. Indeed, by taking $A = (0, \ldots, 0)$ and $X \notin \{(0, \ldots, 0), (1, \ldots, 1)\}$ we have $F(X) + F(A) = 0 + 1 = 1$ and $(X + A)^{\vee} = X^{\vee} < 1$. ∎

Theorem 2.7 *Let $h : B^2 \longrightarrow B$ be a Boolean function. For two arbitrary functions $F : B^n \longrightarrow B$ and $G : B^n \longrightarrow B$ define the function $F \diamond_h G : B^b \longrightarrow B$ by*

$$F \diamond_h G(X) = h(F(X), G(X))$$

for $X \in B^n$.

If both F and G are A-Boolean functions, then $H = F \diamond_h G$ is also A-Boolean.

Theorem 2.7 implies immediately that the set of A-Boolean functions is a subalgebra of the Boolean algebra of all functions from B^n to B.

3 Approximations of Non-Boolean Functions

Let $F : B^n \longrightarrow B$ be a function. Define the set $B_X^F \subset B^n$ by:

$$B_X^F = \{U \in B^n \mid F(X) + F(U) \leq 1 + \bigvee_{A \in \{0,1\}^n} X^A U^A\}.$$

The binary relation $\sim_F$ is given by $X \sim_F Y$ if $B_X^F = B_Y^F$. is an equivalence on B^n. The equivalence class of an element X will denoted by $\langle X \rangle_F$.

For $n = 1$, $X = (x), W = (w)$, and we have $1 + \bigvee_{A \in \{0,1\}} X^A W^A = 1 + x^0 w^0 + x^1 w^1 = 1 + (1 + x)(1 + w) + xw = x + w$. Thus, the current definition reduces to the definition of the equivalence $\sim_F$ that we introduced in [14] for the case $n = 1$. In the same reference we obtained a complete classification of one-argument functions over Boolean algebras with four elements based on the number of Boolean functions needed for approximation.

Let $\langle X \rangle$ be an equivalence class of the relation $\sim_F$. Define the Boolean function $f_{\langle X \rangle} : \mathcal{B}^n \longrightarrow \mathcal{B}$ by

$$f_{\langle X \rangle}(U) = \bigvee_{Y \in \langle X \rangle} F(Y) \cdot \bigvee_{A \in \{0,1\}^n} U^A Y^A$$

for $U \in B^n$. Equivalently, the function $f_{\langle X \rangle}$ can be written as

$$f_{\langle X \rangle}(U) = \bigvee_{A \in \{0,1\}^n} \left(\bigvee_{Y \in \langle X \rangle} F(Y) \cdot Y^A \right) U^A$$

for $U \in B^n$. Then, $f_{\langle X \rangle}$ is also given by:

$$f_{\langle X \rangle}(A) = \bigvee_{Y \in \langle X \rangle} F(Y) \cdot Y^A \tag{2}$$

for $A \in \{0,1\}^n$.

The significance of the equivalence $\sim_F$ is highlighted by the next theorem that shows that for each such class $\langle X \rangle$ there is a Boolean function $f_{\langle X \rangle}$ that coincides with F on the members of the class.

Theorem 3.1 *Let $F : B^n \longrightarrow B$ be a function on the Boolean algebra $\mathcal{B} = (B, \vee, \cdot, ', 0, 1)$. We have $F(U) = f_{\langle X \rangle}(U)$ for every $U \in \langle X \rangle_F$ for every $X \in B^n$.*

It is interesting to observe that a function F is A-Boolean if and only if $B_A^F = B^n$.

For the special case of binary functions, that is, of function $F : B^n \longrightarrow B$ such that $F(B^n) \subseteq \{0, 1\}$ the sets B_X needed for the computations of the relation $\sim_F$ can be easily computed as shown by the next result.

Theorem 3.2 *Let $F : B^n \longrightarrow B$ be a binary function on the Boolean algebra $\mathcal{B} = (B, \vee, \cdot, ', 0, 1)$. We have: $B_X^F = F^{-1}(F(X)) \cup \{X'\}$ for every $X \in B^n$.*

Example 3.3 Let $P, Q \in B^n$ be two n-tuples such that $P \leq Q$. Denote the set $\{X \in B^n | P \leq X \leq Q\}$ by $[P, Q]$.

The literal $F(X) = \overset{P\ \ Q}{X}$ is a binary function, where $P \leq Q$. Suppose that $P \neq 0^n = (0, \ldots, 0)$ and that $Q \neq 1^n = (1, \ldots, 1)$. If $X \in [P, Q]$, then $B_X^F = [P, Q] \cup \{X'\}$ for every $X \in B^n$. Otherwise, $B_X^F = (B^n - [P, Q]) \cup \{X'\}$. Note that the conditions imposed on P and Q mean that at least one of X, X' does not belong to $[P, Q]$. Thus, each X in P, Q has a distinct set B_X^F and therefore, no two such n-tuples are $\sim_F$-equivalent. This makes the Boolean approximation of such literal particularly inefficient, when the interval P, Q is large (that is, P is closed to 0^n and Q is closed to 1^n). ∎

Example 3.4 Let $\mathcal{B} = (B, \vee, \cdot, ', 0, 1)$, where $B = \{0, a, a', 1\}$, and let $F : B^2 \longrightarrow B$, be the function given by

	$F(x_1, x_2)$	x_2: 0	a	a'	1
x_1	0	0	1	0	a
	a	1	0	a	0
	a'	0	a	0	1
	1	a	0	1	0

Since $F(0,a)+F(a,a)=1+0=1$, $((0,a)+(a,a))^{\vee}=(a,0)^{\vee}=a$, the function is clearly non-Boolean. The sets B_X are given by:

$$\begin{array}{l}
B_{(0,a')}=B_{(0,1)}=B_{(a,a')}=B_{(a,1)}=\\
B_{(a',0)}=B_{(a',a)}=B_{(1,0)}=B_{(1,a)}=B^2\\
B_{(0,0)}=B_{(a,a)}=B^2-\{(0,a),(a,0)\}\\
B_{(a',a')}=B_{(1,1)}=B^2-\{(1,a'),(a',1)\}\\
B_{(a,0)}=B_{(0,a)}=B^2-\{(0,0),(a,a)\}\\
B_{(1,a')}=B_{(a',1)}=B^2-\{(1,1),(a',a')\}.
\end{array}$$

which shows that F is A-Boolean on 8 out of the 16 possible pairs in B^2. Thus, there are five equivalence classes of $\sim_F$:

$$\begin{array}{rcl}
\mathcal{C}_1 & = & \{(0,a'),(0,1),(a,a'),(a,1),\\
 & & (a',0),(a',a),(1,0),(1,a)\},\\
\mathcal{C}_2 & = & \{(0,0),(a,a)\},\\
\mathcal{C}_3 & = & \{(a',a'),(1,1)\},\\
\mathcal{C}_4 & = & \{(a,0),(0,a)\},\\
\mathcal{C}_5 & = & \{(1,a'),(a',1)\}.
\end{array}$$

The corresponding Boolean functions that approximate F can be obtained from formula (2):

$$\begin{array}{rcl}
f_{\mathcal{C}_1}(A) & = & a[(0,1)^A \vee (a,a')^A \vee (a',a)^A \vee (1,0)^A],\\
f_{\mathcal{C}_2}(A) & = & f_{\mathcal{C}_3}(A)=0,\\
f_{\mathcal{C}_4}(A) & = & (a,0)^A \vee (0,a)^A,\\
f_{\mathcal{C}_5}(A) & = & (1,a')^A \vee (a',1)^A,
\end{array}$$

for every $A \in \{0,1\}^2$. □

4 Open Problems

Example 3.4 suggests that one could measure the "Booleanicity" of an arbitrary function F by the number

$$\mathsf{bool}(F)=\frac{|\{A\in\{0,1\}^n | F \text{ is } A\text{-Boolean}\}|}{2^n}$$

The closer this number is to one, the fewer are the Boolean functions that are needed for approximating F. In the case of the n-dimensional literals ${}^{P}X^{Q}$ with $P \leq Q$, $P \neq 0^n$, and $Q \neq 1^n$, we have $\mathsf{bool}({}^{P}X^{Q})=\frac{1}{2^n}$, while in the case of the function considered in Example 3.4 we have $\mathsf{bool}(F)=0.5$.

It would be interesting to investigate classes of non-Boolean function whose measure of Booleanicity would be high, and identify classes where this measure is especially low and examine in greater depth the relationship between $\mathsf{bool}(F)$ and the complexity of the Boolean circuits that approximate F.

References

[1] T. Aoki and T. Higuchi. Impact of interconnection-free biomolecular computing. In *Proc.23rd IEEE Int. Symp. Multiple-Valued Logic*, pages 271–276, 1993.

[2] T. Aoki, M. Kameyama, and T. Higuchi. Design of a highly parallel set logic network based on a bio-device model. In *Proc. of the 19th Symposium for Multiple-Valued Logic*, pages 360–367, 1989.

[3] T. Aoki, M. Kameyama, and T. Higuchi. Interconnection-free set logic network based on a bio-device model. *IEEE Letters*, 26:1015–1016, 1990.

[4] T. Aoki, M. Kameyama, and T. Higuchi. Design of interconnection-free biomolecular computing system. In *Proc.21st IEEE Int. Symp. Multiple-Valued Logic*, pages 173–180, 1991.

[5] T. Aoki, M. Kameyama, and T. Higuchi. Interconnection-free biomolecular computing. *IEEE Computer*, 25:41–50, 1992.

[6] P. R. Halmos. *Lectures on Boolean Algebras*. Springer-Verlag, New York, 1974.

[7] M. Kameyama and T. Higuchi. Prospects of multiple-valued bio-information processing systems. In *Proc. of the 18th Symposium for Multiple-Valued Logic*, pages 237–242, 1988.

[8] S. Maeda, T. Aoki, and T. Higuchi. Set logic network based on optical wavelength multiplexing. In *Proc. 22nd IEEE Int. Symp. on Multiple-Valued Logic*, pages 282–290, 1992.

[9] H. McColl. The calculus of equivalent statements. *Proc. London Mathematical Society*, 9:9–20, 1877.

[10] H. McColl. The calculus of equivalent statements. *Proc. London Mathematical Society*, 10:16–28, 1878.

[11] H. McColl. The calculus of equivalent statements. *Proc. London Mathematical Society*, 11:113–121, 1879.

[12] J. C. C. McKinsey. On Boolean functions of many variables. *Transactions of American Mathematical Society*, 40:343–362, 1936.

[13] R. Melter and S. Rudeanu. Functions characterized by functional equations. *Colloquia Mathematica Societatis Janos Bolyai*, 33:637–650, 1980.

[14] C. Reischer and D. A. Simovici. On the implementation of set-valued non-Boolean switching functions. In *Proceedings of the 21st International Symposium for Multiple-Valued Logic*, pages 166–172, 1991.

[15] S. Rudeanu. *Boolean Functions and Equations*. North-Holland/American Elsevier, Amsterdam, 1974.

[16] D. A. Simovici. Several remarks on non-boolean functions over boolean algebras. Technical report, University of Massachusetts Boston, October 2002. Submitted to ISMVL 2003, Tokyo, Japan.

[17] D. A. Simovici and C. Reischer. Several remarks on the complexity of set-valued switching functions. In *Proc. of the 26th Int. Symp. on Multiple-Valued Logic*, pages 166–170, 1996.

[18] Y. Yuminaka, T. Aoki, and T. Higuchi. Design of wave-parallel computing circuits for densely connected architecture. In *Proc. 24th IEEE Int. Symp. on Multiple-Valued Logic*, pages 207–214, 1994.