On Functions Defined on Free Boolean Algebras

Ivo Rosenberg
Université de Montréal, C.P. 6128,
succ. A, Montréal,
P.Q. H3C 3J7, Canada

Dan A. Simovici
University of Massachusetts at Boston,
Department of Computer Science,
Boston, Massachusetts 02125, USA

Szymon Jaroszewicz
University of Massachusetts at Boston,
Department of Computer Science,
Boston, Massachusetts 02125, USA

Abstract

We characterize measures on free Boolean algebras and we examine the relationships that exists between measures and binary tables in relational databases. It is shown that these measures are completely defined by their values on positive conjunctions and an algorithm that leads to the construction of measures starting from its values on positive conjunction is also given, including a formula that allows the evaluation of measures for arbitrary polynomials. Finally, we study pairs of measures generated by ternary tables, that is, by tables that contain missing or unknown values.

1. Introduction

The focus of this paper is a study of measures on free Boolean algebras with a finite number of generators (abbreviated as MFBAs) who take their values in the set $\mathbb{N}$ of natural numbers. As we shall see, these measures play an important role in query optimization in relational databases, and also, in the study of frequent sets in data mining. We show that these measures can help us to mine multi-valued data, in particular, tables that contain null or undefined values.

Let $\mathcal{B} = (B, \mathbf{0}, \mathbf{1}, \bar{}, \vee, \wedge)$ be a Boolean algebra, where $\mathbf{0}, \mathbf{1} \in B$ are two distinguished elements of $\mathcal{B}$, $\bar{}$ is a unary operation, and $\vee, \wedge$ are two binary associative, commutative, and idempotent operation that satisfy the usual axioms of Boolean algebras (see, for example [6, 2]). Here $\mathbf{0}$ and $\mathbf{1}$ are the least and the largest element of the algebra, respectively.

We define

$$x^b = \begin{cases} x & \text{if } b = \mathbf{1} \\ \bar{x} & \text{if } b = \mathbf{0}, \end{cases}$$

for $x \in B$ and $b \in \{\mathbf{0}, \mathbf{1}\}$.

It is a well-known fact (see, for instance [6]) that a Boolean algebra $\mathcal{B} = (B, \mathbf{0}, \mathbf{1}, \bar{}, \vee, \wedge)$ defines a Boolean ring structure, $\mathcal{B} = (B, \mathbf{0}, \mathbf{1}, \wedge, \oplus)$, where $\wedge$ plays the role of the multiplication, and $\oplus$ the role of addition, where

$$x \oplus y = (x \wedge \bar{y}) \vee (\bar{x} \wedge y)$$

for $x, y \in B$. This ring is unitary, commutative, and has characteristic 2 (since $x \oplus x = \mathbf{0}$ for every x). Also, $\mathbf{1} \oplus x = \bar{x}$.

Let $A = \{a_1, \ldots, a_n\}$ be a set of n variables. The set $\mathsf{pol}(A)$ of *Boolean polynomials of the n variables in A* is defined inductively by:

1. $\mathbf{0}$, $\mathbf{1}$, and each a_i belong to $\mathsf{pol}(A)$ for $1 \leq i \leq n$;
2. if p, q belong to $\mathsf{pol}(A)$, then $\bar{p}$, $(p \vee q)$, and $(p \wedge q)$ belong to $\mathsf{pol}(A)$.

If $p, q \in \mathsf{pol}(A)$, then we denote by $(p \oplus q)$ the polynomial $((p \wedge \bar{q}) \vee (\bar{p} \wedge q))$.

Boolean polynomials of the form $(\cdots((p_1 \omega p_2)\omega p_3)\omega \cdots \omega p_n)$ are denoted by $(p_1 \omega p_2 \omega \cdots \omega p_n)$, where $\omega \in \{\vee, \wedge, \oplus\}$. Also, we denote by $\mathsf{var}(p)$ the set of variables that occur in the polynomial p.

Let $\mathcal{B} = (B, \mathbf{0}, \mathbf{1}, \bar{}, \vee, \wedge)$ be a Boolean algebra and let $A = \{a_1, \ldots, a_n\}$ be a set of n attributes. The n-ary function $f_p : B^n \longrightarrow B$ generated by a polynomial $p \in \mathsf{pol}(A)$ is defined as follows:

1. $f_0(x_1, \ldots, x_n) = \mathbf{0}$, $f_1(x_1, \ldots, x_n) = \mathbf{1}$, and $f_{a_i}(x_1, \ldots, x_n) = x_i$ for $1 \leq i \leq n$;

2. $f_{\bar{p}}(x_1, \ldots, x_n) = \overline{f(x_1, \ldots, x_n)}$;

3. $f_{(p \vee q)}(x_1, \ldots, x_n) = f_p(x_1, \ldots, x_n) \vee f_q(x_1, \ldots, x_n)$, and $f_{(p \wedge q)}(x_1, \ldots, x_n) = f_p(x_1, \ldots, x_n) \wedge f_q(x_1, \ldots, x_n)$,

for $(x_1, \ldots, x_n) \in B^n$. We write $p = q$ for $p, q \in \mathsf{pol}(A)$ if $f_p = f_q$.

An A-minterm is a Boolean polynomial

$$p_{b_1, \ldots, b_n} = a_1^{b_1} \wedge \cdots \wedge a_n^{b_n},$$

where $b_i \in \{\mathbf{0}, \mathbf{1}\}$ for $1 \leq i \leq n$. The set of A-minterms is denoted by $\mathsf{mint}(A)$. Any Boolean polynomial in $\mathsf{pol}(A)$ can be uniquely written as a disjunction of some subset of A-minterms (up to the order of the disjuncts). This observation implies that the Boolean algebra $\mathsf{pol}(A)$ is isomorphic to the Boolean algebra of collections of subsets of the set A; thus, $\mathsf{pol}(A)$ has 2^{2^n} elements.

For $A = \{a_1, \ldots, a_n\}$ and $I = \{i_1, \ldots, i_m\} \subseteq \{1, \ldots, n\}$ we denote by a_I the conjunction $a_{i_1} \wedge \cdots \wedge a_{i_m}$. For the special case, when $I = \emptyset$ we write $a_I = \mathbf{1}$.

A *measure* on a Boolean algebra $\mathcal{B} = (B, \mathbf{0}, \mathbf{1}, \bar{}, \vee, \wedge)$ is a non-negative, real-valued function $\mu : B \longrightarrow \mathbb{R}$ such that $\mu(x \vee y) = \mu(x) + \mu(y)$ for every $x, y \in B$ such that $x \wedge y = \mathbf{0}$.

2 A Representation Result for MFBAs

Let $A = \{a_1, \ldots, a_n\}$ be a set of variables. In this context, we find convenient to use the relational database terminology and we refer to the the members of A as *attributes*. We attach a set $\mathsf{Dom}(a_i)$ to each attribute a_i such that $|\mathsf{Dom}(a_i)| \geq 2$. The set $\mathsf{Dom}(a_i)$ is the *domain of* a_i.

A table is a triple $\tau = (T, A, \rho)$, where T is the name of the table, $A = \{a_1, \ldots, a_n\}$ is the heading of the table and $\rho = \{t_1, \ldots, t_m\}$ is a finite set of functions of the form $t_i : A \longrightarrow \bigcup_{a \in A} \mathsf{Dom}(a)$ such that $t_i(a) \in \mathsf{Dom}(a)$ for every $a \in A$. Following the relational database terminology we shall refer to these functions as A-tuples, or simpler, as tuples. If $\mathsf{Dom}(a_i) = \{\mathbf{0}, \mathbf{1}\}$ for $1 \leq i \leq n$, then τ is a binary table.

Let $\tau = (T, A, \rho)$ be a binary table. A *query on the table* τ is a Boolean polynomial in $\mathsf{pol}(A)$. This definition of queries is a formalization of the usual notion of query in databases.

Example 2.1 To retrieve in SQL all tuples t of τ such that at least two of $t(a_1), t(a_2)$ and $t(a_3)$ equal $\mathbf{1}$ we write the query as

select * **from** T **where** ($a_1 = \mathbf{1}$ **and** $a_2 = \mathbf{1}$) **or**
($a_2 = \mathbf{1}$ **and** $a_3 = \mathbf{1}$) **or**
($a_1 = \mathbf{1}$ **and** $a_3 = \mathbf{1}$);

The condition specified in this select corresponds to the polynomial $(a_1 \wedge a_2) \vee (a_2 \wedge a_3) \vee (a_1 \wedge a_3)$. □

A query p defines a table $\mathcal{Q}(p, \tau) = (T_p, A, \rho_p)$, where ρ_p is defined inductively as follows:

1. $\rho_0 = \emptyset$ and $\rho_1 = \rho$;

2. if $p = a_i$, then $\rho_p = \{t \in \rho | t(a_i) = \mathbf{1}\}$;

3. if $p = \bar{q}$, then $\rho_p = \rho - \rho_q$;

4. if $p = (q_1 \vee q_2)$, then $\rho_p = \rho_{p_1} \cup \rho_{p_2}$ and,

5. if $p = (q_1 \wedge q_2)$, then $\rho_p = \rho_{p_1} \cap \rho_{p_2}$.

It is easy to see that for a conjunction

$$p = a_{i_1}^{b_1} \wedge \cdots \wedge a_{i_m}^{b_m},$$

where $b_i \in \{\mathbf{0}, \mathbf{1}\}$ for $1 \leq i \leq m$, the set ρ_p consists of those tuples t such that $t(a_{i_\ell}) = b_\ell$ for $1 \leq \ell \leq m$.

Theorem 2.2 *A function $\mu : \mathsf{pol}(A) \longrightarrow \mathbb{N}$ is a measure if and only if there exists a binary table $\tau = (T, A, \rho)$ such that $\mu(p) = |\rho_p|$ for all $p \in \mathsf{pol}(A)$.*

Proof. Suppose that $\tau = (T, A, \rho)$ is a table. Define the mapping $\mu_\tau : \mathsf{pol}(A) \longrightarrow \mathbb{R}$ by $\mu(p) = |\rho_p|$ for every $p \in \mathsf{pol}(A)$. Let p, q be two polynomials such that $(p \wedge q) = \mathbf{0}$. Then, $\mu_\tau(p \vee q) = |\rho_{p \vee q}| = |\rho_p \cup \rho_q|$. Since $p \wedge q = \mathbf{0}$ we have $\rho_p \cap \rho_q = \emptyset$, so $\mu_\tau(p \vee q) = \mu_\tau(p) + \mu_\tau(q)$. Thus, μ_τ is a measure on $\mathsf{pol}(A)$.

Conversely, let μ be a measure on $\mathsf{pol}(A)$, where $A = \{a_1, \ldots, a_n\}$. If $\vec{b} = (b_1, \ldots, b_n) \in \{\mathbf{0}, \mathbf{1}\}^n$, $p_{\vec{b}} = a_1^{b_1} \wedge \cdots \wedge a_n^{b_n}$ is a minterm and $\mu(p_{\vec{b}}) = k$ consider a set $\sigma_{p_{\vec{b}}}$ of k tuples $t_{\vec{b}}^1, \ldots, t_{\vec{b}}^k$, where $t_{\vec{b}}^j(a_i) = b_i$ for every i, j, $1 \leq j \leq k$, and $1 \leq i \leq n$. Define the table $\tau_\mu = (T, A, \rho_\mu)$, where $\rho = \bigcup \{\sigma_{p_{\vec{b}}} | p_{\vec{b}} \in \mathsf{mint}(A)\}$.

We claim that $\mu(p) = |\rho_p|$ for every polynomial $p \in \mathsf{pol}(A)$. Suppose that p can be expressed as a disjunction of minterms $p = p_{\vec{b}_1} \vee \cdots \vee p_{\vec{b}_k}$, where $\vec{b}_1, \ldots, \vec{b}_k \in \{\mathbf{0}, \mathbf{1}\}^n$. Then, $\mu(p) = \sum_{j=1}^k \mu(p_{\vec{b}_j})$, because $p_{\vec{b}_l} \wedge p_{\vec{b}_h} = \mathbf{0}$ when $l \neq h$. On the other hand, $|\rho_p| = |\bigcup_{j=1}^k \rho_{p_{\vec{b}_j}}| = \sum_{j=1}^k |\rho_{p_{\vec{b}_j}}|$, so $\mu(p) = |\rho_p|$. ■

We shall refer to μ_τ as the *measure induced by the table* τ on $\mathsf{pol}(A)$.

3 An Exclusion-Inclusion Property for MF-BAs

Let p be a polynomial in $\mathsf{pol}(A)$. It is known that p can be uniquely written as

$$p = \bigoplus_{(i_1,\ldots,i_m)} c_{(i_1,\ldots,i_m)} \wedge a_{i_1} \wedge \cdots a_{i_m},$$

where the summation $\bigoplus$ involves the "exclusive or" operation $\oplus$ and is extended to all subsets $\{i_1,\ldots,i_m\}$ of $\{1,\ldots,n\}$. The coefficients $c_{(i_1,\ldots,i_m)}$ belong to the set $\{\mathbf{0},\mathbf{1}\}$. Thus, for a measure μ on $\mathsf{pol}(A)$ it is interesting to evaluate $\mu(p_1 \oplus p_2 \oplus \cdots \oplus p_m)$, where $p_1,\ldots,p_m$ are polynomials in $\mathsf{pol}(A)$.

Theorem 3.1 *Let $\mu : \mathsf{pol}(A) \longrightarrow \mathbb{N}$ be a measure on the free Boolean algebra $\mathsf{pol}(A)$, where $A = \{a_1,\ldots,a_n\}$. If $p_1,\ldots,p_m$ belong to $\mathsf{pol}(A)$, then*

$$\begin{aligned}
&\mu(p_1 \oplus \cdots \oplus p_m)\\
&= \sum_{i=1}^{m} \mu(p_i) - 2 \cdot \sum_{i_1<i_2} \mu(p_{i_1} \wedge p_{i_2}) + \cdots\\
&\quad +(-1)^{\ell-1} \cdot 2^{\ell-1} \cdot \sum_{i_1<\cdots<i_\ell} \mu(p_{i_1} \wedge \cdots \wedge p_{i_\ell}) + \cdots\\
&\quad +(-1)^{m-1} \cdot 2^{m-1} \cdot \mu(p_1 \wedge \cdots \wedge p_m).
\end{aligned}$$

Proof. The statement is clarly true for $m = 1$. We give an argument by induction on m for $m \geq 2$. For $m = 2$ we can write:

$$\begin{aligned}
\mu(p_1 \oplus p_2) &= \mu((\bar{p_1} \wedge p_2) \vee (p_1 \wedge \bar{p_2}))\\
&= \mu(\bar{p_1} \wedge p_2) + \mu(p_1 \wedge \bar{p_2}),
\end{aligned}$$

since $(\bar{p_1} \wedge p_2) \wedge (p_1 \wedge \bar{p_2}) = \mathbf{0}$. Note that $(\bar{p_1} \wedge p_2) \vee (p_1 \wedge p_2) = p_2$, so $\mu(\bar{p_1} \wedge p_2) + \mu(p_1 \wedge p_2) = \mu(p_2)$, which implies $\mu(\bar{p_1} \wedge p_2) = \mu(p_2) - \mu(p_1 \wedge p_2)$. Similarly, $\mu(p_1 \wedge \bar{p_2}) = \mu(p_1) - \mu(p_1 \wedge p_2)$, which yields

$$\mu(p_1 \oplus p_2) = \mu(p_1) + \mu(p_2) - 2 \cdot \mu(p_1 \wedge p_2),$$

which is the desired equality for $m = 2$.

Suppose that the equality holds for m polynomials and let $p_1,\ldots p_m, p_{m+1}$ be $m+1$ polynomials. By the inductive hypothesis we can write

$$\begin{aligned}
&\mu(p_1 \oplus \cdots \oplus p_m \oplus p_{m+1}) =\\
&\mu(p_1 \oplus \cdots \oplus (p_m \oplus p_{m+1})) =\\
&\mu(p_1) + \cdots + \mu(p_{m-1}) + \mu(p_m \oplus p_{m+1})\\
&-2 \cdot \textstyle\sum_{i_1<i_2<m} \mu(p_{i_1} \wedge p_{i_2})\\
&-2 \cdot \textstyle\sum_{i_1<m} \mu(p_{i_1} \wedge (p_m \oplus p_{m+1})) + \cdots\\
&+(-1)^{\ell-1} \cdot 2^{\ell-1} \cdot \textstyle\sum_{i_1<\cdots<i_\ell<m} \mu(p_{i_1} \wedge \cdots \wedge p_{i_\ell})\\
&+(-1)^{\ell-1} \cdot 2^{\ell-1} \cdot \textstyle\sum_{i_1<\cdots<i_{\ell-1}<m} \mu(p_{i_1} \wedge \cdots \wedge p_{i_{\ell-1}}\\
&\qquad \wedge(p_m \oplus p_{m+1})) + \cdots\\
&+(-1)^{m-1} \cdot 2^{m-1} \cdot \mu(p_1 \wedge \cdots \wedge (p_m \oplus p_{m+1})).
\end{aligned}$$

Observe now that

$$\begin{aligned}
&\mu(p_{i_1} \wedge \cdots \wedge p_{i_{\ell-1}} \wedge (p_m \oplus p_{m+1}))\\
&\quad = \mu((p_{i_1} \wedge \cdots \wedge p_{i_{\ell-1}} \wedge p_m)\\
&\qquad \oplus(p_{i_1} \wedge \cdots \wedge p_{i_{\ell-1}} \wedge p_{m+1}))\\
&\quad = \mu(p_{i_1} \wedge \cdots \wedge p_{i_{\ell-1}} \wedge p_m)\\
&\qquad +\mu(p_{i_1} \wedge \cdots \wedge p_{i_{\ell-1}} \wedge p_{m+1})\\
&\qquad -2 \cdot \mu(p_{i_1} \wedge \cdots \wedge p_{i_{\ell-1}} \wedge p_m \wedge p_{m+1}).
\end{aligned}$$

Consequently, we can write:

$$\begin{aligned}
&(-1)^{\ell-1}2^{\ell-1} \textstyle\sum_{i_1<\cdots<i_\ell<m} \mu(p_{i_1} \wedge \cdots \wedge p_{i_\ell})\\
&+(-1)^{\ell-1}2^{\ell-1} \textstyle\sum_{i_1<\cdots<i_{\ell-1}<m} \mu(p_{i_1} \wedge \cdots \wedge p_{i_{\ell-1}}\\
&\qquad \wedge(p_m \oplus p_{m+1}))\\
&= (-1)^{\ell-1}2^{\ell-1} \textstyle\sum_{i_1<\cdots<i_\ell} \mu(p_{i_1} \wedge \cdots \wedge p_{i_\ell})\\
&-(-1)^{\ell}2^{\ell} \mu(p_{i_1} \wedge \cdots \wedge p_{i_{\ell-1}} \wedge p_m \wedge p_{m+1}).
\end{aligned}$$

The last term of the last sum will be attributed to the next term of the necessary sum for $\mu(p_1 \oplus \cdots \oplus p_m \oplus p_{m+1})$. This completes the argument for the above equality. ∎

Corollary 3.2 *Let $\mu, \mu' : \mathsf{pol}(A) \longrightarrow \mathbb{N}$ be two measures on the free Boolean algebra $\mathsf{pol}(A)$, where $A = \{a_1,\ldots,a_n\}$. If $\mu(p) = \mu'(p)$ for every conjunction p of the form $p = a_{i_1} \wedge \cdots \wedge a_{i_m}$, then $\mu = \mu'$.*

Proof. The result follows immediately from Theorem 3.1. ∎

Example 3.3 Consider the "majority polynomial" $p = (a_1 \wedge a_2) \vee (a_2 \wedge a_3) \vee (a_1 \wedge a_3)$. For f_p we have $f_p(x_1,x_2,x_3) = 1$ if and only if at least two of its arguments are equal to 1. Note that

$$p = (a_1 \wedge a_2) \oplus (a_2 \wedge a_3) \oplus (a_1 \wedge a_3).$$

Theorem 3.1 allows us to write

$$\begin{aligned}
\mu(p) &= \mu(a_1 \wedge a_2) + \mu(a_2 \wedge a_3) + \mu(a_1 \wedge a_3)\\
&\quad -2\mu((a_1 \wedge a_2) \wedge (a_2 \wedge a_3))\\
&\quad -2\mu((a_1 \wedge a_2) \wedge (a_1 \wedge a_3))\\
&\quad -2\mu((a_2 \wedge a_2) \wedge (a_1 \wedge a_3))\\
&\quad +4\mu((a_1 \wedge a_2) \wedge (a_2 \wedge a_3) \wedge (a_1 \wedge a_2))\\
&= \mu(a_1 \wedge a_2) + \mu(a_2 \wedge a_3)\\
&\quad +\mu(a_1 \wedge a_3) - 2\mu(a_1 \wedge a_2 \wedge a_3).
\end{aligned}$$

□

Corollary 3.2 shows that the values of a measure on $\mathsf{pol}(A)$ is completely determined by its values on positive conjunctions of the form a_I for $I \subseteq \{1,\ldots,n\}$.

Let $\tau = (T, H, \rho)$ be a table. Note that the contribution of a tuple $(b_1,\ldots,b_n) \in \rho$ to the value of $\mu_\tau(I)$ equals 1

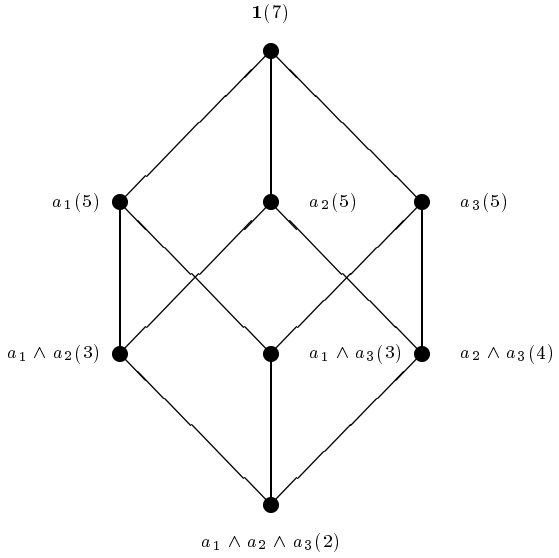

Figure 1. Lattice of Positive Conjunctions

for every set I such that $I \subseteq \{i \in \{1, \ldots n\} \mid b_i = \mathbf{1}\}$. Thus, it is possible to formulate an algorithm that constructs a table τ starting from the values of $\mu(a_I)$ such that $\mu = \mu_\tau$.

Let $\mathsf{PosConj}(A)$ be the set of all positive conjunctions of attributes of A, that is, the set $\{a_I \mid I \subseteq \{1, \ldots, n\}\}$. Consider the Hasse diagram of the poset $(\mathsf{PosConj}(A), \leq)$, where $a_I <= a_J$ if $J \subseteq I$ and label every vertex a_I by $\mu(a_I)$. The algorithm includes the following steps:

1. Initialize the variable ℓ to 0 and ρ to an empty table having the heading A.
2. If there exists a conjunction a_K containing $n - \ell$ conjuncts such that $\mu(a_K) > 0$, then select one such conjunction such that $\mu(a_K)$ has the largest value and go to step 3; if no such conjunction exists, then increment ℓ by 1; if $\ell > n$, exit.
3. Add to ρ a number of $\mu(a_K)$ tuples whose components equal $\mathbf{1}$ for all attributes of K and $\mathbf{0}$, otherwise.
4. Subtract $\mu(a_K)$ from all labels of conjunctions of the form a_J such that $J \subseteq K$. Go to step 2.

Example 3.4 Let $A = \{a_1, a_2, a_3\}$, and let $\mu : \mathsf{pol}(A) \to \mathbb{N}$ be a measure on $\mathsf{pol}(A)$. Fig. 1 shows the lattice of positive conjunctions of attributes of A labeled with values of μ for each of them (in brackets). Let us now use the algorithm given above to construct a database table $\tau = (T, A, \rho)$ such that $\mu_\tau = \mu$. To begin, set the content of the table $\rho = \emptyset$ and consider the conjunction $a_1 \wedge a_2 \wedge a_3$. Since $\mu(a_1 \wedge a_2 \wedge a_3) = 2$, we add two $(1, 1, 1)$ rows to ρ and modify the labels of all conjunctions. The result is shown in Figure 2.

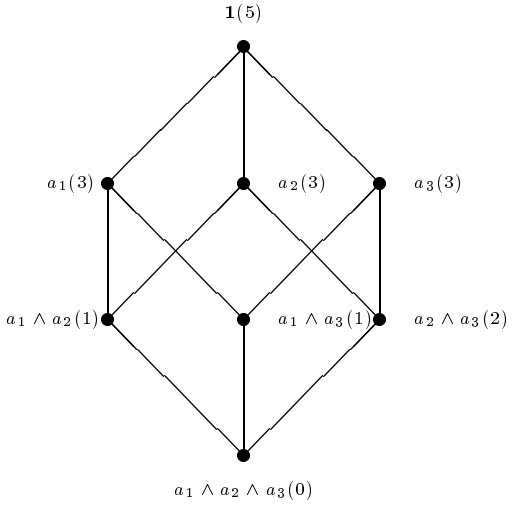

Figure 2. Intermediate Step of the Algorithm

Then, we consider all two-attribute conjunctions, adding rows $(1, 1, 0), (1, 0, 1), (0, 1, 1), (0, 1, 1)$ and modify labels of conjunctions a_1, a_2, a_3 and $\mathbf{1}$ accordingly (cf. Fig. 3).

The last step (not shown) is adding the row $(1, 0, 0)$ to ρ, all the labels become 0, and the algorithm is complete. The table τ is given below

a_1	a_2	a_3
1	1	1
1	1	1
1	1	0
1	0	1
0	1	1
0	1	1
1	0	0

□

Let $\mathcal{V} : \{\mathbf{0}, \mathbf{1}\} \longrightarrow \{0, 1\}$ be the bijection defined by $\mathcal{V}(\mathbf{0}) = 0$ and $\mathcal{V}(\mathbf{1}) = 1$, where $0, 1 \in \mathbb{R}$. Note that

$$\mathcal{V}(a \vee b) = \mathcal{V}(a) + \mathcal{V}(b) - \mathcal{V}(a)\mathcal{V}(b), \quad (1)$$

$$\mathcal{V}(a \wedge b) = \mathcal{V}(a)\mathcal{V}(b), \quad (2)$$

$$\mathcal{V}(\bar{a}) = 1 - \mathcal{V}(a), \quad (3)$$

for every $a, b \in \{\mathbf{0}, \mathbf{1}\}$.

For a Boolean function $f : \{\mathbf{0}, \mathbf{1}\}^n \longrightarrow \{\mathbf{0}, \mathbf{1}\}$ define the real-valued function $\phi_f : \{0, 1\}^n \longrightarrow \{0, 1\}$ by

$$\phi_f(\xi_1, \ldots, \xi_n) = \mathcal{V}(f(\mathcal{V}^{-1}(\xi_1), \ldots, \mathcal{V}^{-1}(\xi_n)))$$

for every $\xi_1, \ldots, \xi_n \in \{0, 1\}$.

Example 3.5 It is easy to verify that if p is the majority polynomial considered in Example 3.3, then for the numerical function ϕ_{f_p} we can write:

$$\phi_{f_p}(\xi_1, \xi_2, \xi_3) = \xi_1\xi_2 + \xi_2\xi_3 + \xi_1\xi_3 - 2\xi_1\xi_2\xi_3$$

for every $\xi_1, \xi_2, \xi_3 \in \{0, 1\}$. Note that the coefficients are the same as the ones in Example 3.3. □

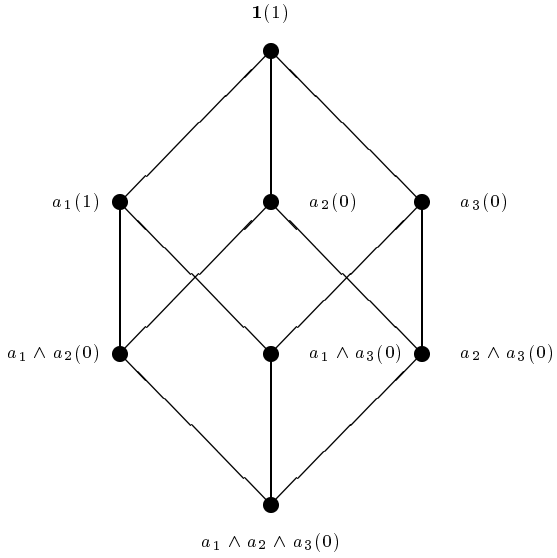

Figure 3. One Further Intermediate Step of the Algorithm

The remark contained in the above Example is not a coincidence. Next, we prove that for every polynomial p, the numerical function ϕ_{f_p} can be expressed as a sum of monomials multiplied by the coefficients that occur in the expansion of $\mu(p)$ given in Theorem 3.1.

Theorem 3.6 *Let $A = \{a_1, \ldots, a_n\}$ and $p \in \mathsf{pol}(A)$. Suppose that*

$$\mu(p) = \sum_{I \in \mathcal{I}} c_I \mu(a_I),$$

where $\mathcal{I}$ is a family of subsets of $\{1, \ldots, n\}$. Then, we have:

$$\phi_{f_p}(\xi_1, \ldots, \xi_n) = \sum_{I \in \mathcal{I}} c_I \xi_I,$$

where ξ_I is the monomial $\xi_I = \xi_{i_1} \cdots \xi_{i_m}$.

Proof. Let $p, q, r \in \mathsf{pol}(A)$ such that $r = (p \vee q)$. We have:

$$\begin{aligned}
&\phi_{f_r}(\xi_1, \ldots, \xi_n) \\
&\quad = \mathcal{V}(f_r(\mathcal{V}^{-1}(\xi_1, \ldots, \xi_n))) \\
&\quad = \mathcal{V}(f_p(\mathcal{V}^{-1}(\xi_1, \ldots, \xi_n)) \vee f_q(\mathcal{V}^{-1}(\xi_1, \ldots, \xi_n)) \\
&\quad = \mathcal{V}(f_p(\mathcal{V}^{-1}(\xi_1, \ldots, \xi_n)) + f_q(\mathcal{V}^{-1}(\xi_1, \ldots, \xi_n)) - \\
&\qquad \mathcal{V}(f_p(\mathcal{V}^{-1}(\xi_1, \ldots, \xi_n))) \cdot \mathcal{V}(f_q(\mathcal{V}^{-1}(\xi_1, \ldots, \xi_n))),
\end{aligned}$$

for $(x_1, \ldots, x_n) \in \{\mathbf{0}, \mathbf{1}\}^n$, by equation 1. Thus,

$$\begin{aligned}
\phi_{f_r}(\xi_1, \ldots, \xi_n) \quad = \quad & \phi_{f_p}(\xi_1, \ldots, \xi_n) + \phi_{f_q}(\xi_1, \ldots, \xi_n) \\
& - \phi_{f_p}(\xi_1, \ldots, \xi_n)\phi_{f_q}(\xi_1, \ldots, \xi_n)
\end{aligned}$$

for $(\xi_1, \ldots, \xi_n) \in \{0, 1\}^n$. Since $\phi_{f_{p \wedge q}}(\xi_1, \ldots, \xi_n) = \phi_{f_p}(\xi_1, \ldots, \xi_n)\phi_{f_q}(\xi_1, \ldots, \xi_n)$, it follows that if $p \wedge q = \mathbf{0}$, then

$$\phi_{f_r}(\xi_1, \ldots, \xi_n) = \phi_{f_p}(\xi_1, \ldots, \xi_n) + \phi_{f_q}(\xi_1, \ldots, \xi_n)$$

for every $(\xi_1, \ldots, \xi_n) \in \{0, 1\}^n$. This show that for a every $\xi = (\xi_1, \ldots, \xi_n)$, the mapping $\mu : \mathsf{pol}(A) \longrightarrow \mathbb{R}$ defined by $\mu_\xi(p) = \phi_{f_p}(\xi)$ is a measure on $\mathsf{pol}(A)$, so Theorem 3.1 is applicable and we can write:

$$\phi_{f_p}(\xi_1, \ldots, \xi_n) = \sum_{I \in \mathcal{I}} c_I \xi_I,$$

for every $(\xi_1, \ldots, \xi_n) \in \{0, 1\}^n$. ∎

4 Applications in Data Mining and Database Query Optimization

In database query optimization and in data mining, it is often necessary to estimate the number of rows in a database table satisfying a given query. Unfortunately in most cases the exact number of rows satisfying a query cannot be computed exactly and has to be estimated (usually using the assumption of statistical independence between attributes).

Let $\tau = (T, A, \rho)$ be a binary table and let K be a set of attributes, $K \subseteq A$. The support of the set K relative to the table τ is defined as the number:

$$\mathtt{supp}_\tau(K) = \{t \in \rho \mid t(a) = \mathbf{1} \text{ for all } a \in K\}.$$

Thus, $\mathtt{supp}_\tau(K) = \mu_\tau(a_{k_1} \wedge \ldots \wedge a_{k_m})$, where $K = \{a_{k_1}, \ldots, a_{k_m}\}$. In other words, the support of an attribute set K in the table τ can we viewed as the value of the measure induced by the table on the Boolean polynomial that describes the attribute set. By extension, we can regard the number $\mu_\tau(q)$ as the support of the query q and we denote this number by $\mathtt{supp}(q)$. There is a considerable research effort in data mining for designing algorithms for discovering all sets of attributes with high support. An idea has recently been raised by H. Manilla in a seminal paper ([4, 5]) is to use supports of attribute sets discovered with a data mining algorithm to obtain the size of a database query. If $q \in \mathsf{pol}(A)$ is a query involving a table $\tau = (T, A, \rho)$ such that q can be written as

$$q = c \oplus \overset{\oplus}{\sum_{I \in \mathcal{I}}} a_I,$$

where $c \in \{\mathbf{0}, \mathbf{1}\}$ and $\mathcal{I}$ is a collection of subsets of $\{1, \ldots, n\}$, then $\mathtt{supp}(q)$ can be obtained from Theorem 3.1 using the numbers $\mathtt{supp}_\tau(a_I)$. Methods that obtain approximative estimations of query sizes been proposed [4], including the use of Maximum Entropy Principle. An open

problem is whether we can give any guarantee on the quality of such an approximation.

The computation of the size of the query using Theorem 3.1 can be often simplified if there is a known maximal number of **1** components in the tuples of the table. For example, in a store that sells 1000 items (corresponding to 1000 attributes in a table that contains the records of purchases) it is often the case that we can use an empirical limit of, say, 8 items per tuple. In this case, conjunctions that contain more than 8 conjuncts can be discarded and the estimation is considerably simplified. Even, if such an upper bound cannot be imposed apriori, it is often the case that we can discard large conjunctions (which have low support). However, there are some risks when approximations of this nature are performed due to the the large values of coefficients that multiply the supports for large conjunctions.

Indeed, consider the tables $\tau^n_{odd} = (T_o, A, \rho_{odd})$, $\tau^n_{even} = (T_e, A, \rho_{even})$, where

$$\begin{aligned} \rho_{odd} &= \{t \in \mathsf{Dom}(A) : n_1(t) \text{ is odd}\}, \\ \rho_{even} &= \{t \in \mathsf{Dom}(A) : n_1(t) \text{ is even}\}, \end{aligned}$$

where $n_1(t)$ denotes the number of attributes equal to **1** in tuple t and $|A| = n$.

Note that for proper subset K of A, we have $\mathtt{supp}_{\tau_{odd}}(K) = \mathtt{supp}_{\tau_{even}}(K)$, while

$$\mathtt{supp}_{\tau^n_{odd}}(A) = \begin{cases} 1 & \text{if } n \text{ is odd} \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\mathtt{supp}_{\tau^n_{even}}(A) = \begin{cases} 1 & \text{if } n \text{ is even} \\ 0 & \text{otherwise.} \end{cases}$$

Thus, from the point of view of the supports of any proper subset of the attribute set the tables τ^n_{odd} and τ^n_{even} are indiscernible. However, the support of certain queries can be vastly different on these tables. For example, consider the polynomial $p = a_1 \oplus a_2 \oplus \ldots \oplus a_n$. We have $\mu_{\tau^n_{odd}}(p) = |\rho_{odd}| = 2^{n-1}$ and $\mu_{\tau^n_{even}}(p) = |\rho_{even}| = 0$. So, ignoring the term that corresponds to the support for a single attribute set (note that this is also the attribute set with the smallest possible support) has a huge impact on $\mu(p)$. Note that the result is consistent with Theorem (3.1) which gives the set of attributes A a coefficient 2^{n-1}. We stress however that the negative result above does not rule out practical applicability of approximating the values of μ_τ since the parity function query used above is by no means a typical database query.

5 Measures Generated by Multi-Valued Tables

Tables that contain incomplete information have been intensively studied in databases. In this section we investigate measures that can be attached to such tables.

Suppose now that $\tau = (T, A, \rho)$, where $A = \{a_1, \ldots, a_n\}$ and all attributes have the same domain $\mathsf{Dom}(a_i) = P$ for $1 \leq i \leq n$, where $P = \{\mathbf{0}, \mathbf{u}, \mathbf{1}\}$. The symbol $\mathbf{u}$ represents *null values*, that is, values that are missing or undefined. We refer to such tables as *ternary tables*. A total order is defined on P by $\mathbf{0} < \mathbf{u} < \mathbf{1}$. The set of functions $P^A = \{f | f : A \longrightarrow P\}$ is ordered by the order defined on P and it can be organized as a Post algebra:

$$\mathcal{P}_A = (P^A, t_{\mathbf{0}}, t_{\mathbf{u}}, t_{\mathbf{1}}, C_{\mathbf{0}}, C_{\mathbf{u}}, C_{\mathbf{1}}, \vee, \wedge),$$

where t_k is the constant function in P^A which has value k for $k \in P$, $C_{\mathbf{0}}, C_{\mathbf{u}}, C_{\mathbf{1}}$ are three unary operations such that if $t' = C_k(t)$, then

$$t'(a) = \begin{cases} \mathbf{1} & \text{if } t(a) = k \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

If D_1 is the unary operation given by $D_1(t) = C_0(C_0(t))$ and $s = D_1(t)$, then

$$s(a) = \begin{cases} \mathbf{1} & \text{if } t(a) \in \{\mathbf{u}, \mathbf{1}\} \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

If $\tau = (T, A, \rho)$ is a ternary table, then $\tau^\uparrow = (T^\uparrow, A, \rho^\uparrow)$ is the binary table defined by $\rho^\uparrow = \{t^\uparrow \mid t \in \rho\}$, where $t^\uparrow = D_1(t)$. Similarly, $\tau^\downarrow = (T^\downarrow, A, \rho^\downarrow)$ is the binary table given by $\rho^\downarrow = \{t^\downarrow \mid t \in \rho\}$, where $t^\downarrow = C_1(t)$. Thus, for a polynomial $p \in \mathsf{pol}(A)$ we have an optimistic evaluation, $\mu_{\tau^\uparrow}(p) = |\rho^\uparrow_p|$, when all missing components $\mathbf{u}$ are interpreted as **1**s, and a pessimistic evaluation, $\mu_{\tau^\downarrow}(p) = |\rho^\downarrow_p|$, when all $\mathbf{u}$s are interpreted as $\mathbf{0}$. In other words, a ternary table $\tau = (T, A, \rho)$ generates two measures $\mu_{\tau^\uparrow}$ and $\mu_{\tau^\downarrow}$ on the set $\mathsf{pol}(A)$.

Note that $|\rho| = |\rho^\uparrow| = |\rho^\downarrow|$ since the tables that concern us here admit multiple tuples with the same content.

Lemma 5.1 *Let $\tau_1 = (T_1, A, \rho_1)$ and $\tau_2 = (T_2, A, \rho_2)$ be two binary tables. There is a ternary table $\tau = (T, A, \rho)$ such that $\rho_1 = \rho^\downarrow$ and $\rho_2 = \rho^\uparrow$ if and only if there exists a bijection $\beta : \rho_1 \longrightarrow \rho_2$ such that $t' = \beta(t)$ implies $t(a) \leq t'(a)$ for every $a \in A$.*

Proof. Suppose that $\rho_1 = \rho^\downarrow = \{t_1, \ldots, t_n\}$ and $\rho_2 = \rho^\uparrow = \{t'_1, \ldots, t'_n\}$, where $\rho = \{s_1, \ldots, s_n\}$ and $t_j(a) \leq s_j(a) \leq t'_j(a)$ for $a \in A$ and $1 \leq j \leq n$. In this case, we can define the function $\beta : \rho_1 \longrightarrow \rho_2$ by $\beta(t_j) = t'_j$ for $1 \leq j \leq n$. It is clear that β has the desired properties.

Conversely, suppose that such a bijection exists. For every $t \in \rho_1$ define the tuple s given by:

$$s_t(a) = \begin{cases} u & \text{if } t(a) < \beta(t)(a) \\ a & \text{if } t(a) = \beta(t)(a) \end{cases}$$

for $a \in A$. The relation ρ is given by

$$\rho = \{s_t \mid t \in \rho_1\}.$$

It is immediate that $\rho_1 = \rho^{\downarrow}$ and $\rho_2 = \rho^{\uparrow}$. ∎

Lemma 5.2 *Let $\tau = (T, A, \rho)$ and $\tau' = (T', A, \rho')$ be two tables such that there is a bijection $\beta : \rho \longrightarrow \rho'$ such that $t(a) \leq \beta(t)(a)$ for every $a \in A$. Then, for each conjunction of the form $p = a_{i_1} \wedge \cdots \wedge a_{i_\ell}$ we have $\mu_\tau(p) \leq \mu_{\tau'}(p)$.*

Proof. Without loss of generality we can assume that $\rho = \{t_1, \ldots, t_n\}$ and $\rho' = \{t'_1, \ldots, t'_n\}$, where $\beta(t_i) = t'_i$ for $1 \leq i \leq n$. Thus, if $\rho_p = \{t_{j_1}, \ldots, t_{j_k}\}$, where $k = \mu_\tau(p)$, we have $t_{i_p}(a) = \mathbf{1}$ for every $a \in \{a_{i_1}, \ldots, a_{i_\ell}\}$. This, in turn, implies $t'_{i_p}(a) = \mathbf{1}$ for every $a \in \{a_{i_1}, \ldots, a_{i_\ell}\}$, which yields $k \leq \mu_{\tau'}(p)$. ∎

The inverse of Lemma 5.2 is given next.

Lemma 5.3 *Let μ, μ' be two measures on $\mathsf{pol}(A)$, where $A = \{a_1, \ldots, a_n\}$ such that $\mu(a_I) \leq \mu'(a_I)$ for every $a_I \in \mathsf{PosConj}(A)$ such that $I \neq \emptyset$ and $\mu(a_\emptyset) = \mu'(a_\emptyset)$. Then, there exist two tables $\tau = (T, A, \rho)$ and $\tau' = (T', A, \rho')$ such that:*

1. *$\mu = \mu_\tau$ and $\mu' = \mu_{\tau'}$;*
2. *there exists a bijection $\beta : \rho \longrightarrow \rho'$ such that $t(a) \leq \beta(t)(a)$ for every $a \in A$.*

Proof. Note that it suffices to prove that there exists an injection β with the property mentioned above, for if, such an injection exists and $|\rho| = \mu(a_\emptyset) = \mu'(a_\emptyset) = |\rho'|$, then β is a bijection.

We shall prove that, under the conditions of the lemma, for each $p \in \mathsf{PosConj}(A)$ there is an injection $\beta_p : \rho_p \longrightarrow \rho'_p$ such that:

1. $t(a) \leq \beta_p(t)(a)$ for every $t \in \rho_p$ and $a \in A$;
2. If $p_1, p_2 \in \mathsf{PosConj}(A)$ have the same number of conjuncts and $p_1, p_2 \leq p$, then $\beta_{p_1}(t) = \beta_{p_2}(t)$ for every $t \in \rho_{p_1} \cap \rho_{p_2}$.

The argument is by induction on the number m of attributes missing as conjuncts in p.

For the basis step, $m = 0$, we have $p = a_1 \wedge \cdots \wedge a_n$, β_p is defined by $\beta_p(t) = t$ for every $t \in \rho_p$, and the second part of the statement is vacuously true.

Suppose that the statement holds for conjuncts that miss m attributes and let $p = a_{i_1} \wedge \cdots \wedge a_{i_k}$ be a conjunct that misses $m+1$ attributes, $\{a_{h_1}, \ldots, a_{h_{m+1}}\}$. Define the conjunctions

$$q_1 = p \wedge a_{h_1}, \ldots, q_{m+1} = p \wedge a_{h_{m+1}}$$

Each of these conjunctions misses m attributes and we have the injections $\beta_1, \ldots, \beta_{m+1}$ such that $\beta_i(t)$ coincides with $\beta_j(t)$ on all tuples t in $\rho_{q_i} \cap \rho_{q_j}$, and $t(a) \leq \beta_i(t)(a)$ for every $t \in \rho_{q_i}$ for $1 \leq i \leq m+1$. Observe that

$$\rho_p = \rho_{p \wedge \bar{a}_{h_1} \wedge \cdots \wedge \bar{a}_{h_{m+1}}} \cap \rho_{q_1} \cap \cdots \cap \rho_{q_{m+1}}.$$

Define the injection β_p as follows. For $t \in \rho_{q_i}$ let $\beta_p(t) = \beta_{q_i}(t)$. Note that β_p is well-defined because of the conditions imposed on the injections β_{q_i}. The number of tuples in ρ'_p that are not in the images of the mappings β_{q_i} is :

$$\begin{aligned} |\rho'_p| - \sum_{i=1}^{m+1} \beta_i(\rho_{q_i}) & \\ = \quad \mu'(p) - \sum_{i=1}^{m+1} \mu_i(q_i) & \\ \geq \quad \mu(p) - \sum_{i=1}^{m+1} \mu_i(q_i) & \\ = \quad |\rho_{p \wedge \bar{a}_{h_1} \wedge \cdots \wedge \bar{a}_{h_{m+1}}}|. & \end{aligned}$$

Thus, the values of the injection β_p on tuples in $\rho_{p \wedge \bar{a}_{h_1} \wedge \cdots \wedge \bar{a}_{h_{m+1}}}$ can be assigned arbitrarily in the set $\rho'_p - \bigcup_{i=1}^{m+1} \beta_i(\rho_{q_i})$. Since the tuples in $\rho_{p \wedge \bar{a}_{h_1} \wedge \cdots \wedge \bar{a}_{h_{m+1}}}$ are the least among the tuples in ρ_p, we also have $t(a) \leq \beta_p(t)(a)$, which means that the injection β_p is definable. ∎

Theorem 5.4 *Let μ, μ' be two measures on $\mathsf{pol}(A)$. There exists a ternary table $\tau = (T, A, \rho)$ such that $\mu = \mu_{\tau^{\downarrow}}$ and $\mu' = \mu_{\tau^{\uparrow}}$ if and only if $\mu(a_I) \leq \mu'(a_I)$ for every $a_I \in \mathsf{PosConj}(A)$ such that $I \neq \emptyset$ and $\mu(a_\emptyset) = \mu'(a_\emptyset)$.*

Proof. This statement follows immediately from Lemmas 5.1, 5.2 and 5.3. ∎

6 Conclusions and Open Problems

We studied properties of measures defined on free Boolean algebras with finite sets of generators. Such measures arise naturally in the evaluation of sizes of queries applied to binary tables in relational databases, a type of tables that has received a large amount of attention in a data mining, particularly related to the identification of frequent set of items and to association rules (see [1, 3]). The measures associated with tables seem to be particularly useful for ternary tables that incorporate the idea of missing information and open a direction of investigation on mining information from incompletely specified databases.

References

[1] J.-M. Adamo. *Data Mining for Association Rules and Sequential Patterns*. Springer-Verlag, New York, 2001.

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

[3] J. Han and M. Kamber. *Data Mining – Concepts and Techniques*. Morgan Kaufmann, San Francisco, 2001.

[4] H. Manilla. Combining discrete algorithms and probabilistic approaches in data mining. In L. DeRaedt and A. Siebes, editors, *Principles of Data Mining and Knowledge Discovery*, volume 2168 of *Lecture Notes in Artificial Intelligence*, page 493. Springer-Verlag, Berlin, 2001.

[5] D. Pavlov, H. Manilla, and P. Smyth. Beyond independence: Probabilistic models for query approximation on binary transaction data. ICS TR-01-09, University of California, Irvine, 2001.

[6] S. Rudeanu. *Boolean Functions and Equation*. North-Holland, Amsterdam, 1974.