Multi. Val. Logic, 1997, Vol. 2, pp. 183–216
Reprints available directly from the publisher
Photocopying permitted by license only

Set-Valued Logic Algebra: A Carrier Computing Foundation

ALIOUNE NGOM[a], CORINA REISCHER[b], DAN A. SIMOVICI[c] and IVAN STOJMENOVIC[d]

[a]*Computer Science Department, University of Ottawa, Ottawa, Ontario K1N 9B4, Canada;* [b]*Department of Math. and Comp. Sci., University of Quebec at Trois-Rivieres, Trois-Rivieres, Quebec G9A 5H7, Canada;* [c]*Department of Math. and Comp. Sci., University of Massachusetts, Boston, MA 02125;* [d]*Computer Science Department, University of Ottawa, Ottawa, Ontario K1N 9B4, Canada*

Recommended for publication by R. Melter

(Received 27 January 1996; In final form 11 May 1996)

An r-valued set logic is the study of functions mapping n-tuples of subsets into subsets over r values. Such functions are called set logic functions or set-valued functions. Completeness and approximation properties for set logic functions are of great interest to engineering research because of recent applications in the design of optical, biological and electronical circuitry to create simpler and more economical circuits specially suitable for parallel computing. Their research on biological molecular computing suggested the interest of studying set logic functions and switching devices. Starting from the needs of these designers the class of bio-algebras were introduced as a class of primal algebras, that is capable of model computations done by bio-circuits. Functions implemented by these circuits are defined on the sets of subsets of finite sets and range over similar sets of subsets and, in general, are non-Boolean. The subsets of the set of Boolean set logic functions are not complete, that is one cannot obtain all set logic functions from such sets. On the other hand, Boolean set logic functions are very important set logic functions, because they can be realized using binary standard electronic circuitry. It is shown that r-valued set logic is isomorphic to 2^r-valued logic.

Keywords: Set-valued functions; completeness; approximation; bio-computing; Boolean function; non-Boolean function; bio-algebras; biological computing

1. INTRODUCTION

The works on carrier computing suggested the study of set-valued logic functions and the complexity of their implementation using

Boolean and non Boolean switching devices. Set logic algebra (SLA) is a special class of multiple-valued logic algebra. It was proposed first as a new foundation of biological molecular computing in [1]. Introductory presentations of this new computing paradigm can be found in [2, 3]. More formal algebraic aspects of SLA have been studied in [4–13]. As an ultra higher-valued logic system, a set logic system offers a new solution to the interconnection problems that occur in highly parallel VLSI systems. The fundamental concept inherent to a set logic system is multiplex computing or logic values multiplexing: this means the simultaneous transmission of logic values. This basic concept enables the realization of superchips free from interconnection problems. Parallel processing with multiplexable information carriers makes it possible to construct large-scale highly parallel system with reduced interconnections. Since the multiplexing of logic values increases the information density, several binary functions can be executed in parallel in a single module. Therefore a great reduction of interconnections can be achieved using optimal multiplexing scheme (see [14, 15]). Possible approaches to the implementation of the set logic system are based on frequencies multiplexing, waves multiplexing and molecules multiplexing, and are called carrier computing systems. For a general perspective on the applications of multivalued logic see [16].

Let U denote the set of all n-place (set logic) functions ($n \geqslant 1$) ranging over a set V of k values and with all values in V. The set U is called the set of n-ary operations on v and the set of n-place functions of k-valued logic in the propositional calculus of k-valued logic. Consider a subset F of U. The functions from F can be combined by attaching values (outputs) of certain functions to variables (inputs) of certain functions in an arbitrary way so that we obtain a single value and no feedback is created. This single value defines a unique function f of external variables to which no value of F functions is attached. We say that f is a composition (in the ordinary sense) of functions from F. The closure of F is the set of all compositions of functions in F and is denoted by $[F]$. The set F is complete in U if $[F] = U$, or equivalently, if each function from U is a composition of functions in F.

The closure of the Boolean set $C = \{\cup, \cap, ^-, \text{constants}\}$, i.e. $[C]$, is referred to as the *set of Boolean set logic functions with constants.* Consider any subset S of C and a subset F of non-Boolean functions from U. It is well known that S is not complete, that is $[S] \neq U$. However, Boolean functions are convenient choices as building blocks in the design of carrier computing circuits; they are considered to be relatively cheap elements, that is, they are usually obtained with small-cost considering the complexity of their implementation versus that of non-Boolean functions. A variation of the definition of completeness is then the concept of S-completeness, which assumes that for composition besides functions in F one can freely utilize functions in S. The concept of S-completeness brings new interesting questions on the properties of S itself. For instance, given two distinct subsets of C, they may lead exactly or approximately to the same S-completeness criteria, and the subset with the smallest cardinality is probably the most interesting for engineers who develop efficient carrier computing chips or circuits.

1.1. The Concept of Set Logic

We consider the set $R = \{e_0, \ldots, e_{r-1}\}$ as a set of information carriers. The carrier computing circuits operate on the power set of R, that is, the set of all subsets of R, denoted by 2^R. An element $X \in 2^R$ is called a logic set. Thus, if R contains r carrier elements then 2^R contains 2^r logic sets. The basic operations over 2^R are the set-theoretic union $\cup$, intersection $\cap$, and complement $^-$ operations. It is well known that 2^R is a Boolean algebra $(2^R, \emptyset, R, \cup, \cap, ^-)$ when equipped with these three set-theoretic operations.

Mathematically, set logic algebra is based on the isomorphism between the Boolean algebras $(B^r, 0, 1, +, \cdot, ')$ and $(2^R, \emptyset, R, \cup, \cap, ^-)$. B^r is the Cartesian product of r 2-element Boolean algebras $B = \{0, 1\}$. A carrier computing circuit can be described as a set logic function of n variables, which is a mapping $f: (2^R)^n \rightarrow (2^R)$ that maps n-tuples of subsets of R into a subset of R. The set of all such functions is referred to as r-valued set logic. The number of n-place set logic functions is quite considerable: there are $(2^r)^{(2^r)^n}$ such functions [1]. As the radix r increases, this number becomes enormously large. For example, for $n = 1$ and $r = 2$ there are 256 functions, while for $n = 1$ and $r = 3$ we find 16,777,216 such one-place functions.

A small fraction of these functions are Boolean functions, that is functions that can be constructed from constants and variables, using union, intersection and complementation. The number of n-place r-valued Boolean functions of set logic is $(2^r)^{2^n}$. For instance, for $r = 2$ and $r = 3$ we find 16 and 64 (respectively) one-place Boolean set logic functions.

1.2. Set Logic Algebra as Multiple-Valued Logic Algebra

Another way of looking to set logic functions is considering them as functions over a logic with 2^r values which provides a rich collection of functions over logic with very high radix. We shall see that there are certain computational advantages in adopting this dual point of view.

Consider the following situation arising in the synthesis of switching circuits. We have certain basic elements called gates. Each gate has one or several inputs and a single output. The gate receives signals on the inputs and transforms them into the output signal. For simplicity we assume that all signals belong to the same finite set denoted by $L_k = \{0, \ldots, k-1\}$, $k \geqslant 1$. The functioning of the gate can be described by the assignment of the output value $f(x)$ to every ordered n-tuple $x = (x_1, \ldots, x_n)$ of input values. Thus the gate realizes a function f of n variables ranging in the finite set L_k with values in L_k. In other words, f maps the Cartesian power L_k^n (of all ordered n-tuples of elements of L_k) into L_k. Denote by $P_k(n)$ the set of all such functions. Thus $P_k(n)$ consists of k^{k^n} functions. The set $P_k(n)$ is called the set of n-ary operations on L_k in universal algebras and the set of n-ary functions of k-valued logic in multiple-valued logic algebras. The set P_k is defined by $P_k = \cup \{P_k(n) \mid n \geqslant 1\}$.

As discussed in [11], every r-valued set logic function can be regarded as a k-valued logic function for $k = 2^r$, as follows. Without loss of generality we may use characteristic binary vectors to represent the elements of 2^R as binary numbers. A subset $X \in 2^R$ is represented as binary number $x_0 x_1 \ldots x_{r-2} x_{r-1}$ determined by $x_i = 1$ if and only if $e_i \in X$ for $0 \leqslant i \leqslant r-1$. Next, X is mapped into the natural number x which has binary representation $x_{r-1} x_{r-2} \ldots x_1 x_0$, i.e., $x = 2^{r-1} x_{r-1} + 2^{r-2} x_{r-2} + \ldots + 2x_1 + x_0$. We also refer to x_i, $0 \leqslant i \leqslant r-1$ as the i-th coordinate of x.

For instance, for $r = 2$ we have $R = \{e_0, e_1\}$ and the elements of $2^{\{e_0, e_1\}}$ are represented in the following way:

Set X	Binary Repr. x_1x_0	Decimal Repr. x
$\emptyset$	00	0
$\{e_0\}$	01	1
$\{e_1\}$	10	2
$\{e_0, e_1\}$	11	3

The operations $\cap, \cup, ^-$ are represented by the following tables:

$\cup$	0	1	2	3
0	0	1	2	3
1	1	1	3	3
2	2	3	2	3
3	3	3	3	3

$\cap$	0	1	2	3
0	0	0	0	0
1	0	1	0	1
2	0	0	2	2
3	0	1	2	3

$-$	
0	3
1	2
2	1
3	0

In general, $x \cup y = u$ and $x \cap y = v$ are determined by $u_i = \max\{x_i, y_i\}$ and $v_i = \min\{x_i, y_i\}$ for $0 \leqslant i \leqslant r-1$, while $\bar{x} = k-1-x$. We refer to these functions as union, intersection and complement functions in P_k, $k = 2^r$. For example, for $r = 3$ (i.e. $k = 8$) we have $3 \cup 5 = 011 \cup 101 = 111 = 7$, $3 \cap 5 = 011 \cap 101 = 001 = 1$, and $\bar{3} = \overline{011} = 100 = 4$.

The symmetric difference over 2^R will be denoted by $\oplus$ and is defined as $X \oplus Y = (X - Y) \cup (Y - X)$. If the elements of 2^R are represented as binary numbers, then $x \oplus y = w$ is determined by $w_i = 0$ if $x_i = y_i$ and $w_i = 1$ if $x_i \neq y_i$, for $0 \leqslant i \leqslant r-1$. This is simply the exclusive or operation in P_k, $k = 2^r$. We have the following table for $r = 2$:

$\oplus$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

1.3. Boolean Set Logic Functions

The set of all Boolean set logic functions is denoted by $[C]$, that is, the closure of the set $C = \{\cup, \cap, ^-, c_0, \ldots, c_{k-1}\}$, where c_i is the constant function i. $[B]$ is the set of Boolean functions obtained from the set $B = \{\cup, \cap, ^-\}$ (constants are not involved in their composition). Also, let $[C]_k(n)$ be the set of all n-place Boolean functions in P_k. The number of such functions is $|[C]_k(n)| = k^{2^n}$, where $k = 2^r$. For example,

$|[C]_4(1)| = 16$, $|[C]_8(2)| = 4096$. $[C]([B])$ is also referred to as the set of Boolean functions with constants (without constants, respectively).

If $X, Y \in 2^R$, then $X^0 = R, X^1 = X$ and $XY = X \cap Y$. As it was observed in [17], p. 37, a (set logic) function b is Boolean if and only if it can be written as

$$b(X_1, \ldots, X_n) = \sum_{(i_1 \ldots i_n)} A^n_{i_1 \ldots i_n} X_1^{i_1} \ldots X_n^{i_n},$$

for every $X_1, \ldots, X_n \in 2^R$, where $A^n_{i_1 \ldots i_n}$ are constants in 2^R while the sum is extended over all binary numbers $i_1 \ldots i_n$ (i.e. $i_j \in \{0, 1\}$ for $1 \leqslant j \leqslant n$) between 0 and $2^n - 1$. In the above formula, the sum represents the extension of $\oplus$. For the case $n = 2$, for example, the indices range between 0 and 3 and we have

$$b(X_1, X_2) = A^2_{00} X_1^0 X_2^0 \oplus A^2_{01} X_1^0 X_2^1 \oplus A^2_{10} X_1^1 X_2^0 \oplus A^2_{11} X_1^1 X_2^1$$

$$= A^2_0 \oplus A^2_1 X_1 \oplus A^2_2 X_2 \oplus A^2_3 X_1 X_2$$

Almost all set logic functions are non-Boolean. That is, they are not polynomials of the Boolean algebra of 2^R, since they cannot be written in the canonical polynomial form. Therefore, we cannot construct all the set logic functions from any subset of C by composition, that is, subsets of Boolean set logic functions are not complete. On the other hand, Boolean set logic functions are very cheap functions. In switching circuits usually the Boolean functions are at our disposal at small cost, that is, they can be realized using binary standard electronic circuitry.

If $x, y \in L_k$, then we write $xy = x \cap y$. Therefore, a (set logic) function b is Boolean if and only if it can be written as

$$b(X_1, \ldots, X_k) = a_0 \oplus \sum_{i_1 \ldots i_m} a_{i_1 \ldots i_m} x_{i_1} \ldots x_{i_m},$$

where a_0 and $a_{i_1 \ldots i_m}$ are constants in L_k and the sum is extended over all subsets $\{i_1, \ldots, i_m\}$ of m distinct indices from the set $\{1, \ldots, n\}$. The sum represents the extension of the operation $\oplus$ over L_k. The coefficients a_0 and $a_{i_1 \ldots i_m}$ are uniquely determined by the function b. This canonical polynomial form is sometime called the Rudeanu formula or the Zhegalkin polynomial.

2. FUNCTIONAL COMPLETENESS

Completeness properties are of great importance to designers of combinatorial logic circuits (which include carrier computing circuits) because they allow the reduction of the implementation complexity of the carrier computing circuits.

Investigations of completeness and related topics, which are usually called functional completeness problems, are directly related to logic circuit design, and they have a wide area of applications in addition to their mathematical importance.

The compositions of functions from a set F of (set logic) functions are precisely the functions that can be realized by combinatorial switching or logic circuits constructed from gates with transformation functions from F. A gate is of type f (or realizes f) if it transforms any input $(x_1, x_2, \ldots, x_n) \in L_k^n$, into a single output $f(x_1, x_2, \ldots, x_n)$ depending on $x_1, x_2, \ldots, x_n$ only. Thus the type of a gate is completely described by a single function of n variables ranging over L_k and with all values in L_k. Consider a collection of gates of type $f_i \in P_k$. These gates can be combined into logic circuits by attaching outputs of certain gates to inputs of certain gates in an arbitrary way so that the resulting circuit has a single output and no feedback is created. This means that the single output of such a circuit defines a function $f \in P_k$ of external inputs to which no gate is connected. This function can be described as the composition (or superposition) of the f_i's.

The compositions includes permuting variables in a function, identifying two variables, and replacing variables by functions from F. If the functions from F are treated as circuits then the composition is the creation of new circuits by using the output of some circuits as input to other ones, where it is allowed to use multiple copies of same output or to permute input wires. The closure of F, that is, $[F]$, contains all compositions of functions from F.

A subset F of P_k is said to be closed if it contains all compositions (or superpositions) of its members [18, 19], that is, if the formation of composition does not lead outside F, or equivalently, if $[F] = F$. It is well known that the set of all closed sets ordered by set inclusion is an algebraic lattice.

For closed sets F such that $F \subset P_k$(proper inclusion), F is a P_k-maximal set if there is no closed set G such that $F \subset G \subset P_k$ (i.e., P_k covers F in the set of closed sets ordered by $\subseteq$). That is F cannot be properly extended to a closed subset G of P_k.

The first and most natural problem is the characterization of subsets F of P_k such that $[F] = P_k$. Such sets are called (functionally) complete in P_k. The algebras (L_k, F) with F complete in P_k are called *primal algebras*. Thus F is complete if and only if any function of P_k can be realized by a circuit constructed exclusively from gates realizing functions from F. That is, a subset F of P_k is complete in P_k if P_k is the least closed set containing F (in other words, if the functions in F can produce by composition any function in P_k).

A complete set F in P_k is called a *base* of P_k if no proper subset of F is complete in P_k. The *rank of a base* is the number of its elements. The unique function of a complete singleton set $F = \{f\}$ is called a *Sheffer function* for P_k; in other words, f is a Sheffer function if $\{f\}$ is a base (of rank 1).

A first general completeness criterion was given in [20] in 1921. This criterion, which has been rediscovered many times since, is most naturally expressed in terms of P_k-maximal sets.

THEOREM 2.1 (Completeness criterion). *A subset $F \subseteq P_k$ is complete in P_k, if and only if F is a subset of no P_k-maximal set.*

This is, in some sense, the most general completeness criterion because it is defined in terms of F only and covers all cases. Thus the completeness problem is solved by the determination of all P_k-maximal sets.

To describe the P_k-maximal sets, we need the following essential concept of *function preserving a relation* (see [19]).

Let $h \geqslant 1$. An h-ary relation ρ on L_k is a subset of L_k^h (i.e., a set of h-tuples over L_k) whose elements are written as columns. Given h rows n-vectors $a_i = (a_{i1}, a_{i2}, \ldots, a_{in})$ $(1 \leqslant i \leqslant h)$ we write $(a_1, a_2, \ldots, a_h)^T \in \rho$ to indicate that $(a_{1j}, \ldots, a_{hj})^T \in \rho$ for all $1 \leqslant j \leqslant n$, where T denotes the transpose (this means that the $h \times n$ matrix with rows $a_1, a_2, \ldots, a_h$ has all columns in ρ). We say that an n-place $f \in P_k$ preserves ρ if $(f(a_1), f(a_2), \ldots, f(a_h))^T \in \rho$ whenever $(a_1, a_2, \cdots, a_h)^T \in \rho$. Then, the set of functions preserving ρ is denoted by Pol ρ.

It was observed in [21] that the P_k-maximal sets can be best described as set of functions preserving a relation see also [22, 23]. All P_k-maximal sets are of the form Pol ρ for some relation ρ. Their full description is given in [19, 24]. They are grouped into six classes and [19] formulated the following general completeness criterion.

THEOREM 2.2 (Rosenberg's General Completeness Criterion). *Each P_k-maximal set is of the form* Pol ρ, *where ρ is one of the six following relations on L_k:*

R_1 *Every partial order on L_k with a least and a greatest element.*
R_2 *Every relation $(x, s(x))| x \in L_k$ where s is a permutation of L_k with k/p cycles of the same length $p(k = pr$ and p prime).*
R_3 *Every quaternary relation*

$$\{(a_1, a_2, a_3, a_4) \in L_k^4 | a_1 + a_2 = a_3 + a_4\},$$

where $(L_k, +)$ is a p-elementary abelian group ($k = p^r$ and p prime).
R_4 *Every non-trivial equivalence relation on L_k.*
R_5 *Every central relation on L_k.*
R_6 *Every relation λ_T determined by an h-regular family T of equivalence relations on L_k (also referred to as a predicate relation).*

A set $F \subseteq P_k$ is complete if and only if for every relation ρ described under R_1–R_6 above, there exists an $f \in F$ not preserving ρ.

Let $R_i(k)$ denote the number of relations in the class R_i for $1 \leqslant i \leqslant 6$, and let $\pi(k) = \Sigma_{1 \leqslant i \leqslant 6} R_i(k)$ denote the total number of P_k-maximal sets (i.e., number of relations in P_k). In [24] the reader can find a rather complicated formula for $\pi(k)$ obtained through a purely combinatorial argument. The following table contains the number of various types of maximal sets:

k	R_1	R_2	R_3	R_4	R_5	R_6	$\pi(k)$
2	1	1	1	0	2	0	5
3	3	1	1	3	9	1	18
4	18	3	1	13	40	7	82
5	190	6	6	50	335	36	643
6	3285	35	0	201	11490	171	15182
7	88851	120	120	875	7758205	813	7848984
8	3640440		30	4138	549758283980		

As k grows, most of the relations belongs to the class R_5, and $\pi(k)$ increases very rapidly.

3. BOOLEAN COMPLETENESS IN SET LOGIC

Using only Boolean functions, one cannot achieve functional completeness in many-valued set logic (i.e., the case $r \geqslant 2$). The Boolean algebra $(2^R, \emptyset, R, \cup, \cap, ^-)$ is not functionally complete except for the case $r = 1$. A variation on the definition of functional completeness is then the concept of Boolean completeness in which we allow the use of Boolean functions in the compositions of non Boolean functions from a given subset F of $P_k - [C]$. In the sequel, we let $S(n)$ denote the set of n-place functions of a given set S and assume $k = 2^r$.

In [1, 25–29] the question of constructing all set logic functions using Boolean functions is studied. Boolean functions are those composed from the set $\{\cup, \cap, ^-, c_0, c_1, \ldots, c_{k-1}\}$, where c_i is the constant function i. Since the set is not complete (except for $k \leqslant 2$), some functions are added to the set of Boolean functions to form a complete set. Boolean set logic functions (which are a small fraction of the collection of set logic functions) are very important set logic functions, because they can be realized using binary standard electronic circuitry. In these considerations Boolean functions are considered cheap elements in the design of set logic functions. Following these investigations, it is interesting to characterize all sets of functions which become complete when Boolean functions are added to them.

Let S be a set of cheap functions. In set logic, since we are more interested to determine the completeness criteria under some set of Boolean functions, we assume S to be a non empty subset of C. A maximal set F in P_k is said to be S-maximal in P_k if $S \subseteq F$. A subset F of P_k–$[S]$ is said to be S-complete in P_k if the set $S \cup F$ is complete in P_k. An S-complete set F in P_k is called a S-base of P_k if no proper subset of F is S-complete in P_k. The rank of a S-base is the number of its elements. A function $f \in P_k$ is S-Sheffer (or Sheffer with S functions) for P_k if $\{f\} \cup S$ is complete in P_k, or equivalently, if $\{f\}$ is an S-base (of rank 1) of P_k. For example, bio-output [1], conversion [30] and literal [28] functions are respectively $\{\cup, ^-, -$, constants$\}$-Sheffer, $\{\cup, ^-$, constants$\}$-Sheffer, and $\{\cup, \cap$, constants$\}$-Sheffer, respectively.

In [6] the following general result is obtained.

THEOREM 3.1 (General S-Completeness Theorem). *A subset F of functions in P_k is S-complete in P_k if and only if it is contained in no S-maximal set in P_k.*

We saw in the previous section that there is an exponential (in k) number of P_k-maximal sets divided into six classes R_1–R_6. We refer to these classes also as partial order, self-dual, quasilinear, equivalence, central and predicate classes of maximal sets, respectively. To determine which of the $\pi(k)$ maximal sets are S-maximal sets, it is sufficient to check for each of them whether it contains all S-functions. The general solution to the S-completeness problem is then to find the S-maximal sets.

In the search of S-maximal sets, the following theorem is very useful. Let a three-place majority function be any function f that satisfies the following property: $f(x,x,y)=f(x,y,x)=f(y,x,x)$ for any x and y. One such function is

$$f(x,y,z)=(x\cap y)\cup(x\cap z)\cup(y\cap z).$$

The following result is obtained in [31].

THEOREM 3.2 *If a closed set can be described as set of functions preserving an h-ary relation ρ and it contains a majority function, then $h\leqslant 2$.*

The next result (see [7]) follows from Theorem 3.2.

COROLLARY 3.3 *If $\{\cup,\cap\}\subseteq S$, then there is no S-maximal set in the quasilinear and predicate classes and no S-maximal set corresponding to h-ary central relations for $h\geqslant 3$.*

For S Boolean, two cases of S-completeness have been studied in literature and known as the problem of C-completeness ($S=C=\{\cup,\cap,{}^-,c_0,c_1,\ldots,c_{k-1}\}$) and B-completeness ($S=B=\{\cup,\cap,{}^-\}$). The C-completeness, called weak completeness in [11] and Boolean completeness in [12] (with the same meaning), determines the completeness criteria in set logic under compositions with C functions while the B-completeness problem studied in [6] determines the completeness criteria in set logic under compositions with B functions.

Note that, in general, S is not necessarily a Boolean set and also k is not necessarily a power of 2. Earlier known S-completeness theorems cited in the survey paper of [19] are given in [32, 39] for S containing only constant k-valued logic functions (for any k). There are few results known in literature other than completeness with Boolean functions. Nevertheless, these results are from a more general concept of S-completeness called *relative completeness*.

3.1. Relative Completeness

Relative completeness (or R-completeness) is first mentioned and studied in [40]. To describe R-completeness, we need the following definitions. Define the i-th projection of arity n by $\pi_i^n(x_1,\ldots,x_n)=x_i$ for $1 \leqslant i \leqslant n$. Let Π_k denote the set of all the projections over L_k. $F \subseteq P_k$ is a clone of operations on L_k (or clone for short) iff $\Pi_k \subseteq F$ and F is closed. All the clones on L_k form a lattice. The clone generated by $F \subseteq P_k$ is the closure $[F \cup Pi_k]$. It is well known that $F \subseteq P_k$ is complete if $[F \cup \Pi_k] = P_k$. Let R be a clone and $F \subseteq P_k - R$. F is complete relative to R (or R-complete) iff the clone $[F \cup R]$ equals P_k.

The definition of R-completeness is almost the same as that of S-completeness; the only difference is that R is a clone. However, R-completeness is a generalization of functional completeness, because every functionally complete set F is Π_k-complete. On the other hand, an S-complete set F is a functionally complete set $G = F \cup S$ which in turn is Π_k-complete. The concept of R-completeness is then much more general than that of S-completeness.

Let R be a clone. A maximal set F in P_k is R-maximal in P_k if $R \subseteq F$. R-maximal sets are also called maximal clones. A R-complete set F in P_k is an R-base of P_k if it does not contain any complete proper subset. The rank of an R-base is the number of its elements. A function $f \in P_k$ is R-Sheffer for P_k if $\{f\}$ is an R-base of P_k.

The following theorem [40], analogous to the Post's completeness criterion, gives a necessary and sufficient condition for F to be R-complete.

THEOREM 3.4 (R-completeness criteria). *The set $F \subseteq P_k$ is R-complete if and only if it is contained in no maximal clone in P_k.*

Therefore, the technique used to determine R-completeness criteria is exactly the same as that of S-completeness: search for R-maximal sets in the six Rosenberg classes of maximal sets. In [40] R-completeness criteria are obtained for $R = \{\min(x, y\}, \bar{x}\}$ and $R = \{\tilde{x}, x^{(k+1)}\}$, where $\tilde{x} = x + 1(\text{mod } k)$ and, $x^{(k+1)} = k-1$ if $x = k-1$ and 0, otherwise. We won't cover all the known results in R-completeness as this will be out of the scope of set logic algebra.

4. *C*-COMPLETENESS AND *B*-COMPLETENESS IN SET LOGIC

In [11] a full description of C-complete sets, C-bases and C-Sheffer functions in P_4 is given. In [12] the general case of arbitrary r is studied and it is proved that there are $2^r - 2$ C-maximal sets in r-valued set logic, all defined by some equivalence relations. In [5] the classes of functions under compositions with C-functions are determined, and the C-bases and C-Sheffer functions for P_8 are enumerated. The reference [6] presents the B-completeness criteria and proves that there are $\text{Bl}(r) + 2^r - 3$ B-maximal sets in r-valued set logic (where $\text{Bl}(r)$ is a Bell number, that is, the number of set partitions of the set R), defined by $\text{Bl}(r) - 1$ central relations and $2^r - 2$ equivalence relations. These are all known results on C-completeness and B-completeness. The full description of B-complete sets, B-bases and B-Sheffer functions in P_4 and P_8 is given in [7,41].

The following table constructed from results obtained in [6, 12] shows the Rosenberg classes containing B-maximal or C-maximal sets (by entry 0) or not (by entry 1).

	R_1	R_2	R_3	R_4	R_5	R_6
B	1	1	1	0	0	1
C	1	1	1	0	1	1
Reference	[12]	[6, 12]	[12]	[6, 12]	[12]	[6, 12]

In contrast to other classes of Rosenberg relations, some of the equivalence relations are preserved by operation from [C] or [B]. Constant functions trivially satisfy any equivalence relation and thus the analysis of C-maximal sets from [12] can be applied in the B-completeness criteria, meaning that a maximal set from equivalence relation is B-maximal if and only if it is C-maximal. So, B-maximal sets from equivalence relations are equivalent to C-maximal sets.

There are $k-2$ B-maximal sets and $k-2$ C-maximal sets in the equivalence class R_4.

For convenience, we denote the equivalence relations by ε_i, $1 \leqslant i \leqslant k-2$, and the corresponding C-maximal (B-maximal) sets by $E_i = \mathrm{Pol}\, \varepsilon_i$. The relation ε_i is defined as follows. Each equivalence class of ε_i is a set interval $[x, x \cup i]$, where $x \cap i = 0$ and $\subseteq$ is the set inclusion ($a \subseteq b$ iff $a_m \leqslant b_m$ for $0 \leqslant m \leqslant r-1$). There are 2^t equivalence classes of ε_i, each containing 2^{r-t} elements (t is the number of 0's in the binary representation of i). For example, for $r=3$ the six equivalence relations which are preserved by all $[C]$ and $[B]$ functions are the following:

$$
\begin{aligned}
\varepsilon_1 &= \{(0,1), (2,3), (4,5), (6,7)\} \\
\varepsilon_2 &= \{(0,2), (1,3), (4,6), (5,7)\} \\
\varepsilon_3 &= \{(0,1,2,3), (4,5,6,7)\} \\
\varepsilon_4 &= \{(0,4), (1,5), (2,6), (3,7)\} \\
\varepsilon_5 &= \{(0,1,4,5), (2,3,6,7)\} \\
\varepsilon_6 &= \{(0,2,4,6), (1,3,5,7)\}
\end{aligned}
$$

For $r=2$ we have 2 equivalence relations

$$
\begin{aligned}
\varepsilon_1 &= \{(0,1), (2,3)\} \\
\varepsilon_2 &= \{(0,2), (1,3)\}
\end{aligned}
$$

The c-th coordinate i_c of i is called fixed (free) if $i_c = 0 (i_c = 1)$. All elements of an equivalence class have the same c-th coordinate whenever c is a fixed coordinate (i.e., if x and y are in the same equivalence class of ε_i and $i_c = 0$ then $x_c = y_c$).

If the elements of L_k are drawn as edges of an r-dimensional hypercube then the equivalence relations ε_i are all partitions of the hypercube into equal size subcubes. The subdivision is uniquely determined by the subcube containing 0, and the subcube containing 0 is uniquely determined by its largest element (between 1 and $k-2$).

Only some unary central relations from R_5 are preserved by $[B]$ functions. The unary central relations are all non empty proper subsets of L_k. Let $\mathrm{Bl}(r)$ be a Bell number, that is, the number of set partitions of a set R with r elements. Let γ be a unary central relation. The relation γ is any subset of L_k having the following properties: if

$a \in \gamma$ then $\bar{a} \in \gamma$; if $a, b \in \gamma$, then $a \cap b \in \gamma$. Each element of γ is a subset of $R = \{e_0, e_1, \ldots, e_{r-1}\}$, where e_i is 0 or 1. Thus, γ corresponds to a subalgebra of Boolean algebra on R. Every set partition of R defines one such subalgebra as follows. If e_a and e_b are in the same class of given set partition then they either both are 0 or both are 1 in each element of γ; γ contains all such elements (therefore 0 and $k-1$ in all cases). Every set partition (except the trivial one when all elements are in the same set) defines one subset γ corresponding to a B-maximal set. There are $\mathrm{Bl}(r)-1$ B-maximal sets in the central class R_5.

We denote the unary central relations by γ_j for $0 \leqslant j \leqslant \mathrm{Bl}(r)-2$, and the corresponding B-maximal sets by $C_j = \mathrm{Pol}\ \gamma_j$. The number of elements in γ_j is 2^m where m is the number of equivalence classes in the set partition, $1 \leqslant m \leqslant r-1$. For convenience, we let $\gamma_0 = \{0, k-1\}$ denote the first unary central relation and also let $C_0 = \mathrm{Pol}\ \gamma_0$ be the corresponding B-maximal set. For example, for $r = 3$ there exists four unary central relations which are preserved by all $[B]$ functions:

$$\gamma_0 = \{0, 7\}$$
$$\gamma_1 = \{0, 1, 6, 7\}$$
$$\gamma_2 = \{0, 2, 5, 7\}$$
$$\gamma_3 = \{0, 3, 4, 7\}$$

For $r = 2$ we have only one central relation $\gamma_0 = \{0, 3\}$.

The next theorem was obtained in [6, 12].

THEOREM 4.1 *For $k = 2^r$, P_k has exactly $\mathrm{Bl}(r) + k - 3$ B-maximal sets and $k-2$ C-maximal sets.*

In the search for S-maximal sets, one could use Theorem 3.4 as a mean to reduce the search. This was done in [6] for B-maximal sets.

Applying the general S-completeness theorem we obtain now the B-completeness and C-completeness criteria. A subset $F \subseteq P_k - [B]$ is B-complete in P_k if and only if $F - C_i \neq \emptyset$ for $1 \leqslant i \leqslant \mathrm{Bl}(r) + k - 3$ (see [6]). A function f is B-Sheffer in P_k if and only if $f \notin C_i$ for $1 \leqslant i \leqslant \mathrm{Bl}(r) + k - 3$. A subset $F \subseteq P_k - [C]$ is C-complete in P_k if and only if $F - E_i \neq \emptyset$ for $1 \leqslant i \leqslant k-2$ (see [12]). A function f is C-Sheffer in P_k if and only if $f \notin E_i$ for $1 \leqslant i \leqslant k-2$.

Once the completeness criteria are known, the intersection properties of S-maximal sets usually determine the S-complete sets, the S-bases and the S-Sheffer functions.

5. THE INTERSECTION PROPERTIES OF C-MAXIMAL SETS AND B-MAXIMAL SETS IN SET LOGIC

We begin by discussing intersection properties of C-maximal sets. The following theorem [5] enumerates the number of n-place r-valued set logic functions in each C-maximal set.

THEOREM 5.1 *We have* $|E_i(n)| = 2^{(r-t)2^{rn}} + t2^{tn}$.

The equivalence relation ε_i is maximal (minimal) if the binary representation of i has exactly one bit 0 (one bit 1, resp.). For example, for $r=3$, ε_5 is maximal and ε_2 is minimal. The intersection of all C-maximal sets defined by maximal relations only (or minimal relations only) gives the set of all $[C]$-functions (see [5]). Furthermore, the most interesting result from [5] is that the intersection of all $k-2$ C-maximal sets is exactly the set of all Boolean functions. This determines the place of the closed set of all Boolean set logic functions in the lattice of all closed sets of P_k.

Next we mention intersection properties of B-maximal sets. Properties given in [5] involve only equivalence relations; the maximal sets are C-maximal sets. Since the C-maximal sets are B-maximal sets defined by equivalence relation, then the intersection properties given in [5] are also applicable to all such B-maximal sets. Some properties involve B-maximal sets defined from unary central relations. In the sequel, we let B_s denote any B-maximal set for $1 \leqslant s \leqslant \mathrm{Bl}(r) + k - 3$. The following theorem (see [7, 41]) enumerates the number of n-place r-valued set logic functions in each B-maximal set defined by a unary central relation γ.

THEOREM 5.2 *We have* $|C(n)| = |\gamma|^{|\gamma|^n} \cdot (2^r)^{2^{rn} - |\gamma|^n}$.

If ε_i is a non-maximal equivalence relation and $\gamma_i = \{j | 0 \subseteq j \subseteq i\} \cup \{j | \bar{i} \subseteq j \subseteq k-1\}$ then Pol γ_i is B-maximal [41]. This result gives a procedure to construct central relations (corresponding to B-maximal sets) from equivalence relations. The equivalence classes $\{j | 0 \subseteq j \subseteq i\}$ and $\{\underline{i} | \bar{i} \subseteq j \subseteq k-1\}$ are always in the non maximal relation ε_i (for convenience, we say that γ_i is defined from ε_i). There are r maximal equivalence relations and thus, in general for $r \geqslant 3$, $k-2-r$ central relations corresponding to B-maximal sets are constructed in such a way. For example in P_{16} there are 14 central relations and

$10(=16-2-4)$ among them are defined from 10 nonmaximal equivalence relations corresponding to B-maximal sets. (It is easy to see that γ_i contains 2^{r-t+1} elements.)

Few properties in [41] involve only unary central relations. Such results are summarized here: if Pol γ and Pol γ' are B-maximal from distinct central relations then $\mathrm{Pol}(\gamma \cap \gamma')$ is also B-maximal and, $\mathrm{Pol}\,\gamma \cap \mathrm{Pol}\,\gamma' \subset \mathrm{Pol}(\gamma \cap \gamma')$.

Let $\cap B_s(n)$ be the intersection of all B-maximal sets. In [41] it is shown that $|\cap B_s(n)| = 2^{2n}$ and thus the size of the intersection is independent of r (or k). Moreover, for $n=1$ the four [B]-functions preserving all relations are: the constant function $f(x)=0$, the identity function $f(x)=x$, the complement function $f(x)=k-1-x$, and the constant function $f(x)=k-1$. Furthermore, $B_k(n)$, the set of all n-place [B]-functions in P_k, is contained in $\cap B_s(n)$ and then the intersection of all B-maximal sets gives the upper bound for the number of all Boolean functions without constants in P_k (cf. [41]).

6. CLASSIFICATIONS AND ENUMERATIONS OF n-PLACE SET LOGIC FUNCTIONS

Intersection properties are used as a theoretical mean to classify all set logic functions according to the S-maximal sets they belong to. The functions in P_k may be classified by their membership in the S-maximal sets, leading to a natural classification of S-bases into equivalence classes (called aggregates). [7] described algorithms for facilitating the classification and enumeration of (set logic) functions based on the given completeness criteria.

Let $M_1,\ldots,M_m$ be m S-maximal sets determined by the S-completeness criteria. As mentioned previously, a subset F of P_k is S-complete in P_k if and only if for each i, $1 \leqslant i \leqslant m$, $F - M_i \neq \emptyset$ or in other words, there is $f \in F$ such that $f \notin M_i$. Define the map $\phi: P_k \to \{0,1\}^m$ by setting $\phi(f) = d_1 \cdots d_m$, where $d_i = 0$ if $f \in M_i$ and $d_i = 1$ if $f \notin M_i$. We call $\phi(f)$ the characteristic vector of f. If $\phi(f) = \phi(g)$, then we write $f \equiv g$. Clearly $\equiv$ is an equivalence relation on P_k and so it partitions P_k into pairwise disjoint equivalence classes. Note that for $f \equiv g$ we have either $f, g \in M_j$ or $f, g \notin M_j$ for all $1 \leqslant i \leqslant m$. We write AB for $A \cap B$, A^0 for A and A^1 for $\bar{A}$ (where $A, B \subseteq P_k$). Clearly, each class is of the form

$M_1^{d_1} \ldots M_m^{d_m}$, where $d_1, \ldots, d_m \in \{0,1\}$. If a set F is S-complete, then replacing a function f in F by any function in its equivalence class results in another S-complete set. Therefore, the study of the characteristic vectors provides information on the intersection properties of families of S-maximal sets.

In general, the number of possible classes (characteristic vectors) of (set logic) functions is 2^m, where m is the number of S-maximal sets in P_k. So, for large m (say, $m \geqslant 8$) it is better to use a computer program to generate most classes and enumerate (or generate) the functions in each class, in order to find the intersection properties in a fast way and also to seek for complete sets (S-bases and S-Sheffer functions) (cf. [41]). For such large m, classifying functions using only the intersection properties is truly tedious. Moreover, we must find the intersection properties of S-maximal sets first, which is not very easy for large m. In the other hand, using computer programs we can obtain at the same time the classes of functions and also those intersections properties which decide which classes are empty.

In [5, 11] the set logic functions are classified according to the C-maximal sets to which they belong to. In [5] specific results are obtained for $r=2$, while in [11] similar results are obtained for $r=3$. Computational results for both cases are included in [41], which agree with the theoretical results in [5, 11].

The following combinatorial result is presented in [5, 41, 11].

THEOREM 6.1 *There exist* 4 (29) *classes of functions of* 2-*valued* (3-*valued, resp.*) *set logic under compositions with* C-*functions.*

Another combinatorial result included in [5, 31] is

THEOREM 6.2 *There exist* 8 (200) *classes of functions of* 2-*valued* (3-*valued, resp.*) *set logic under compositions with* B-*functions.*

7. CLASSIFICATIONS AND ENUMERATIONS OF S-BASES AND S-SHEFFER (SET LOGIC) FUNCTIONS IN P_K

Once we have all the nonempty classes of (set logic) functions, we can discuss the S-completeness properties in P_k in terms of these classes instead of individual functions; if a set is S-complete, then by replacing a function in the set by any function in the corresponding equivalence class yields another S-complete set. If $f \in F \subseteq P_k$ and $f \equiv g$, then clearly F is S-complete in P_k if and only if $(F - \{f\}) \cup \{g\}$ *is* S-complete in P_k. In other words, it suffices to study the S-completeness in P_k up to the equivalence $\equiv$. It is easy to see that $F \subseteq P_k$ is S-complete in P_k if and only if $s = \Sigma\{\phi(f) | f \in F\}$ (bitwise or operation) has all coordinates equal to 1 (i.e., in decimal, $s = 2^m - 1$, where m is the number of S-maximal sets in P_k). Once we know all the characteristic vectors, we can find all S-complete sets in P_k and all S-bases by a direct combinatorial check. If we associate $A = \{i | s_i = 1\}$ to $s_1 \ldots s_m \in \{0,1\}^m$ and if $A_1, \ldots, A_t$ are the subsets of $\{1, \ldots, m\}$ corresponding to the characteristic vectors, the S-completeness problem is reduced to the listing of subsets of $A_1, \ldots, A_t$ covering $\{1, \ldots, m\}$ and the basis problem to the listing of such coverings which are irredundant (no proper subset covers $\{1, \ldots, m\}$).

Let m be the number of S-maximal sets in P_k. A set of (set logic) functions $F = \{f_1, \ldots, f_s\}$ is called *nonredundant in* P_k, if for each i, $1 \leqslant i \leqslant s$, there exists an S-maximal set M_j, in P_k, $1 \leqslant j \leqslant m$ which does not contain f_i while all the other functions f_t $(1 \leqslant t \leqslant s,\ t \neq i)$ are elements of S_j. In other words, F is nonredundant if and only if for each $g \in F$ the sum $s = \Sigma\{\phi(f) | f \in F - \{g\}$ (bitwise or operation) has at least one 0 coordinate (nonredundancy condition). We call nonredundant incomplete sets simply *addable.* The *rank* of a addable set is the number of its elements. A class of S-bases, also called an *aggregate* is the set of all S-bases having the same set of characteristic vectors. For an S-base, its class of S-bases is the set of classes of functions for functions belonging to the S-base.

Conditions of S-completeness and nonredundancy of a set F can be conveniently expressed by using characteristic vectors of set logic functions belonging to F. From the definitions of S-completeness, nonredundancy and S-bases, it follows that there are two conditions for $F \subseteq P_k$ to be an S-base: S-completeness and nonredundancy. An S-base corresponds to a minimal cover of $(1, \ldots, 1)$ (unit vector), and an nonredundant set corresponds to a minimal cover of some non-unit vector in which some 0's may occur (except null vector). Using the bitwise *or* operation $\vee$ for characteristic vectors

$$(a_1, \ldots, a_m) \vee (b_1, \ldots, b_m) = (a_1 \vee b_1, \ldots, a_m \vee b_m),$$

criteria for the S-completeness and irredundancy of a set $\{c_1, \ldots, c_t\}$ of characteristic vectors are respectively the following (cf. [42]):

$$c_1 \vee \ldots \vee c_t = (1,\ldots,1) \text{ } S\text{-completeness}$$

$$c_1 \vee \ldots \vee c_t \neq c_1 \vee \ldots \vee c_{j-1} \vee c_{j+1} \vee \ldots c_t \text{ irredundancy}$$

If we have a complete list of characteristic vectors for non empty classes of (set logic) functions of a set, we can enumerate all its aggregates. As an example, assume a set M contains 4 S-maximal sets M_1, M_2, M_3, M_4 and 6 classes of (set logic) functions:

0011 0100 1000 0010 0001 0000

For instance, the first class is the set $M_1 M_2 \overline{M_3 M_4}$, where $\bar{M}_i = M - M_i$. M has exactly two aggregates $\{1,2,3\}$ and $\{2,3,4,5\}$. Bitwise or operation for the set $\{1,2,3\}$ results in 1111 ((S-) completeness). Bitwise or operation for the set $\{1,2\}$ results in 0111, for the set $\{1,3\}$ results in 1011 and for the set $\{2,3\}$ results in 1100 (nonredundancy). The set $\{1,3,4\}$ is redundant, because applying the bitwise or operation to the sets $\{1,3,4\}$ and $\{1,3\}$ give equal results (1011).

On the basis of a backtrack procedure for lexicographic enumeration of all subsets of a set of n elements, algorithms using the S-completeness and irredundancy conditions are described in [7,42]. These algorithms determine all S-bases consisting of n-place functions from a given S-complete set in P_k and enumerate all classes of S-bases in P_k. The lexicographic algorithm enumerates classes of S-bases for every rank at the same time. Moreover the maximal ranks of S-bases are automatically given as a result. They are applied in the determination of C-bases and B-bases of P_4 and P_8 (cf. [7]); some computational results thus obtained for the case $S = C$ coincide with some theoretical results given in (5,11).

Classification and enumeration of (classes of) S-bases can be achieved also by a purely theoretical way. However as for classification and enumeration of functions, it is much more difficult for large m particularly when enumerating S-Sheffer functions for P_k by applying the principle of inclusion and exclusion. This is also true in finding theoreticaly the maximum rank of a S-base given large m. However, from Theorem 3.1 it follows that the maximum rank of any S-base in P_k is less than or equal to m. The rank of any C-base of P_4 (P_8) is between 1 and 2 (1 and 4, resp.), cf. [5,11] (in [7] these results are verified using a computer program). Also from [7], the rank of any B-base of P_4 (P_8) is between 1 and 3 (1 and 7, respectively).

Suppose we have m S-maximal sets $M_1,\ldots,M_m$, then an n-place function f is S-Sheffer for P_k if and only if $f\notin M_i$ for $1\leqslant i\leqslant m$, that is, iff $f\in P_k(n)-(M_1(n)\cup\ldots\cup M_m(n))$. The number of n-place S-Sheffer functions is $|P_k(n)-(M_1(n)\cup\ldots\cup M_m(n))|$. If m is small ($m\leqslant 7$) and the cardinality of the M_i's is known, the number of S-Sheffer functions for P_k can be usually calculated using the principle of inclusion and exclusion and the known knowledges on the intersection properties of these S-maximal sets. For large m (say $m\geqslant 8$) it may be hard to use this principle because of the large number of intersections to consider. In that case, it may be sufficient to estimate the number of S-Sheffer functions, that is, find a lower bound and an upper bound on that number.

In [11] the following result is obtained.

THEOREM 7.1 *The number of C-bases of rank two containing n-place functions in P_4 is*

$$2^{2^{2n+1}+2^{n+1}}-2^{2^{2n}+2^{n+1}+2^n+1}+2^{2^{n+2}}.$$

The number of n-place C-Sheffer functions in P_4 is

$$2^{2^{2n+1}}-2^{2^{2n}+2^n+1}+2^{2^n+1}.$$

In [5] a similar result is obtained for P_8.

THEOREM 7.2 *The number of n-place C-Sheffer functions in P_8 is:*

$$8^{8^n}-3\cdot 2^{8^n}4^{4^n}-3\cdot 4^{8^n}2^{2^n}+6\cdot 2^{2^n}2^{4^n}2^{8^n}+3\cdot 4^{4^n}2^{2^n}-6\cdot 2^{4^n}2^{2^{n+1}}+2\cdot 8^{2^n}.$$

It is easy to see that the ratio of n-place C-Sheffer functions over all n-place set logic functions approaches 1 as n grows. Therefore almost all set logic functions in P_4 and P_8 are C-Sheffer functions.

In general, the number of S-bases of P_k consisting of n-place (set logic) functions in an aggregate can be calculated as product of the numbers of n-place functions in the classes of functions determined by the characteristic vectors. Summing these numbers for all aggregates for a rank we obtain corresponding data for the S-bases of the rank and finally the number of all S-bases consisting of n-place set logic functions. In [7] the following tables for $r=2$ and $r=3$, respectively, are constructed; they show the number of C-bases of P_4 and P_8 containing one-place functions for a given rank.

Rank	1	2
Number of C-bases	144	2304

Rank	1	2	3	4
Number of C-bases in P_8	15894144	2020225024	458900576	889192448

In [7, 41] the following result concerning B-Sheffer functions is obtained.

THEOREM 7.3 *The number of n-place B-Sheffer functions for P_4 is*

$$4^{4^n} - 2^{4^n+2^n+1} - 2^{2^n} \cdot 4^{4^n-2^n} + 2^{2^n+1} + 2^{4^n+1} - 2^{2^n}.$$

In P_8 it becomes hard to apply the principle of inclusion and exclusion to find the number of 3-valued B-Sheffer functions. This is due to the large number of intersections of B-maximal sets: there are ten B-maximal sets in P_8 and then 1013 intersections. However, in [7, 41] an estimation the number of B-Sheffer functions by an interval is given, that is, a lower bound and an upper bound are found. To compute the upper bound and the lower bound they obtain the number of functions which are in no B-maximal sets defined from central relations on L_8. Next, they consider the number of functions which are in no B-maximal sets defined from equivalence relations on L_8; this number is given in [5] as the number of n-place C-Sheffer functions of 3-valued set logic (as we have explained, a C-maximal set from [5] is also a B-maximal set). The upper bound and the lower bound can then be determined using these two quantities. Let $C(n)$ denote the set of n-place functions preserving unary central relations on L_8 (i.e., $\cup C_j(n), j = 0, 1, 2, 4$) and let $E(n)$ denote the set of n-place functions preserving equivalence relations on L_8 (i.e., $\cup E_i(n), i = 1, 2, 3, 4, 5, 6$). Also we let $\overline{C(n)}$ ($\overline{E(n)}$) be the set of set logic functions not preserving any unary central relations (equivalence relations, respectively). $\overline{E(n)}$ is simply the number of n-place C-Sheffer functions for P_8.

THEOREM 7.4 (in [7, 41]). *The number of n-place 3-valued set logic functions which are in no B-maximal sets defined from unary central relations on L_8 is*

$$\overline{|C(n)|} = P_8(n) - C(n) = 8^{8^n} - 2^{2^n} \cdot 8^{8^n-2^n} - 3 \cdot 4^{4^n} \cdot 8^{8^n-4^n} + 3 \cdot 2^{2^n} \cdot 4^{4^n-2^n} \cdot 8^{8^n-4^n}.$$

In the same references the following result is shown:

LEMMA 7.5 *Let Sh(n) denote the set of n-place B-Sheffer functions in P_8. We have*

$$\overline{|C(n)|} - |E(n)| < |Sh(n)| < \overline{|C(n)|} < \overline{|E(n)|}.$$

The ratio of all n-place B-Sheffer functions for P_4 over $|P_4|$ and the ratios

$$\frac{\overline{|C(n)|} - |E(n)|}{|P_8(n)|} \text{ and } \frac{\overline{|C(n)|}}{|P_8(n)|}$$

approach 1 when $n \to \infty$ and therefore, almost all 2-valued and 3-valued set logic functions are B-Sheffer functions.

In [7] the following tables for $r = 2$ and $r = 3$ are given, respectively. The first table shows the number of B-bases of P_4 containing one place functions for a given rank. The second one shows the number of aggregates for a given rank in P_8.

Rank	1	2	3
Number of B-bases	108	5184	1728

Rank	1	2	3	4	5	6	7
Number of aggregates	1	4301	150973	230632	40166	1496	9

8. APPROXIMATION PROPERTIES IN SET LOGIC

Boolean set logic functions are important because they can be realized using binary standard electronic circuitry. Therefore, we need to be able to decide whether a set logic function is Boolean and, in the negative case, which is the most likely situation, to be able to approximate portions of the graph of non Boolean functions by graphs of Boolean functions. This offers the possibility of building hybrid circuits, that is, circuits that use binary electronic components and non-binary optical or biological components which correspond to Boolean and non-Boolean functions, respectively.

8.1. The Equivalence of a Set Logic Function

Results concerning Boolean approximation of set logic functions in [9] seek to identify partitions of the graphs of set logic functions such that the blocks of the partitions give the maximal interpolation domains of these functions by Boolean set logic functions. The authors determined an upper bound on the complexity of bio-circuits that realize set logic functions. This bound is based on an equivalence attached to a set logic function such that the classes of the quotient set of the definition domain with respect to such an equivalence coincide with the sets on which they can evaluate the function by computing a value of a Boolean function. The reference [9] contains a list of one-place set logic functions based on the number of maximal interpolation classes. It might be possible to obtain S-completeness criteria of sets of set logic functions based on parameters of the attached Boolean interpolation sets. That is, if a given set F contains non-Boolean functions, we can first replace (i.e., approximate) all or some of them by Boolean functions (from corresponding interpolation sets) and then determine the S-completeness criteria of the new set F' which contains the Boolean approximations of functions from F. The problem here is that if we decide to approximate all non-Boolean functions of F, the new set F' may not be complete as it contains only Boolean functions. The interesting question is how to achieve functional completeness or S-completeness for such sets.

If f is a one-place Boolean (set logic) function then, for any $X, Y \in 2^R$ have $f(X) \oplus f(Y) \subseteq X \oplus Y$. This property of Boolean function was obtained in [43] and generalized in [17].

A function $F \in P_k(1)$ generates an equivalence $\sim_F$ on 2^R. Namely, whenever $F(X) \oplus F(W) \subseteq X \oplus W$ if and only if $F(Y) \oplus F(W) \subseteq Y \oplus W$ for every $W \in 2^R$, then $X \sim_F Y$. The equivalence class of X with respect to $\sim_F$ will be denoted by $[X]_F$. Let $F, G \in P_k(1)$. F and G are equivalent if the one-place set logic function h given by $h(X) = F(X) \oplus G(X)$ is a Boolean function. We denote this by $F \equiv G$ and also denote the equivalence class of F by $[F]_{\equiv}$.

The role of the equivalence classes of sets is clarified by the next result from [9].

THEOREM 8.1 *Let* $F: 2^R \rightarrow 2^R$. *For every set* $X \in 2^R$ *there exists a Boolean function* $b_X: 2^R \rightarrow 2^R$ *such that* $F(Y) = b_X(Y)$ *for every* $Y \in [X]_F$.

Further, if $F \equiv G$ then $[X]_F = [X]_G$ for every $X \in 2^R$.

The equivalence classes $[X]_F$ of the quotient set $2^R/\sim_F$ constitute the sets on which we can approximate any function F of $P_k(1)$ by Boolean functions. The set of equivalence classes $[F]_\equiv$ determines a partition of $P_k(1)$ where each class $[F]_\equiv$ of functions gives in fact the maximal interpolation domain of these functions (from $[F]_\equiv$) by Boolean functions.

Consider a set logic function F and its corresponding carrier computing circuit C. The results in [9] allow the design of hybrid set logic circuits (biological and digital, for instance) equivalent to C, in which the Boolean components of F are implemented with binary circuits while we retain carrier computing switching devices (e.g., bio-devices) for computing the non-Boolean components of F. This enables the reduction of the implementation complexity of F.

Using an algorithm, the authors give a complete catalog of all 256 functions of $P_4(1)$ grouped according to their common equivalence relations. Consider, for instance, the function F given by $F(0)=3$, $F(1)=2$, $F(2)=1$ and $F(3)=2$. Applying the definition of $\sim_F$, the classes of the equivalence relation $2^{\{e_0,e_1\}}/\sim_F$ are then 0, 1, 2, 3.

8.2. Characterization of Boolean Collections

Boolean collections of sets are collections of subsets of a given set whose characteristic function is a Boolean set logic function. These collections are introduced and studied in [13], where combinatorial aspects of these collections are also explored. Boolean collections appear to play a role in the approximation of non-Boolean set logic functions by Boolean functions and, therefore, are relevant in the study of carrier computing circuits, where set logic functions are used. These results are useful for characterizing certain testing conditions involving set logic functions.

Formally, a nonempty collection of sets $\mathscr{C} \subseteq (2^R)^n$ is a *Boolean collection* if there exists a n-place Boolean set logic function f such that $\mathscr{C} = \mathscr{C}_f$, where $\mathscr{C}_f = \{X \in (2^R)^n \mid f(X) = \emptyset\}$.

Any singleton set $\{L\}$ is a Boolean collection, since $\{L\} = \mathscr{C}_f$, where $f(X) = X \oplus L$ for $X \in 2^R$. The set $(2^R)^n$ is a Boolean collection defined by the Boolean set logic function $f(X_1, \ldots, X_n) = \emptyset$ for $X_1, \ldots, X_n \in 2^R$. Note that for any set $H \in 2^R$ and any Boolean set logic function f the collection $\mathscr{C}=\{X\in(2^R)^n | f(X)=H\}$ is Boolean since we can write $\mathscr{C}=\{X\in(2^R)^n | \ g(X)=\emptyset\}$, where g is a set logic function given by $g(X)=f(X)\oplus H$ for all $X\in(2^R)^n$.

The set of Boolean collections on $(2^R)^n$ is closed with respect to intersection. Indeed, if f and g are two Boolean set logic functions, it is easy to verify that $\mathscr{C}_f\cap\mathscr{C}_g=\mathscr{C}_{f\cup g}$, where $(f\cup g)(X)=f(X)\cap g(X)$ for every $X\in(2^R)^n$. This allows for the consideration of the least Boolean collections that contains an arbitrary collection of sets. Also, note that if f, g are two Boolean set logic functions, then their equalizing collection $\mathscr{C}_{f=g}=\{X\in(2^R)^n \mid f(X)=g(X)\}$ is also a Boolean collection since $\mathscr{C}_{f=g}=\mathscr{C}_{f\oplus g}$, where $(f\oplus g)(X)=f(X)\oplus g(X)$ for $X\in(2^R)^n$.

A collection of sets $\mathscr{C}\subseteq 2^R$ is Boolean if and only if it is a set interval $[P, Q]=\{X\in 2^R | P\subseteq X\subseteq Q\}$, where $P, Q\in 2^R$.

There exists a bijection between Boolean collections on 2^R and pairs of sets (A, B) such that $A\subseteq B$. Therefore, there are 3^r distinct Boolean collections in 2^R.

General Boolean collections are characterized by Theorem 8.2.

THEOREM 8.2 *Let $f\colon 2^R\to 2^R$ be a Boolean function given by*

$$f(X_1,\ldots,X_n)=\sum\{A^n_{j_1\ldots j_n}X_1^{j_1}\cdots X_n^{j_n}|(j_1,\ldots,j_n)\in\{0,1\}^n\}.$$

The collection C_f is non-empty if and only if

$$A^n_0\subseteq\bigcup\{A^n_i|1\leqslant i\leqslant 2^n-1\}$$

and it consists of all tuples $X=(X_1,\ldots,X_n\}$ that satisfy

$$\frac{A^m_0-A^m_2-\cdots-A^m_{2^m-2}\subseteq X_m}{\subseteq(A^m_0\oplus A^m_1)-(A^m_2\oplus A^m_3)-\cdots-(A^m_{2^m-2}\oplus A^m_{2^m-1})}$$

for $1\leqslant m\leqslant n$, where

$$A^m_i=A^{m+1}_{2i}\oplus A^{m+1}_{2i+1}X_{m+1}$$

for $1\leqslant m\leqslant n-1$.

COROLLARY 8.3 *There are exactly $(2^{2^n}-1)^r$ non-empty Boolean collections. For each such collection $\mathscr{C}$, the Boolean function $f\colon(2^R)^n\to 2^R$ such that $\mathscr{C}=\{(X_1,\ldots,X_n)|f(X_1,\ldots,X_n)=\emptyset\}$ is uniquely determined.*

Consider a collection of sets $\mathscr{C} \subseteq (2^R)^n$. Let F be an n-place set logic function. It is interesting to examine the collections of sets $C_{F,f} \subseteq (2^R)^n$ for which a Boolean function $f:(2^R)^n \rightarrow 2^R$ can be found such that $F(X) = f(X)$ for every $X \in \mathscr{C}$. In other words, the problem is to determine the portions of $(2^R)^n$ where the computation of the function F can be replaced by the computation of a Boolean function f. Consider the graph $G(F)$ of F defined as $G(F) = \{(X_1, \ldots, X_n, X_{n+1}) \in (2^R)^{n+1} \mid F(X_1, \ldots, X_n) = X_{n+1}\}$. Then, $G(F)$ is a Boolean collection determined by the Boolean function h given by $h(X_1, \ldots, X_n, X_{n+1}) = f(X_1, \ldots, X_n) \oplus X_{n+1}$, where $f(X_1, \ldots, X_n) = F(X_1, \ldots, X_n)$. It is clear that $C_{F,f}$ is a fragment of $G(F)$.

The maximal Boolean collections contained in the graph of non Boolean functions give the maximal sets on which we can approximate these non Boolean functions by Boolean ones. Therefore, S-completeness properties of non Boolean set logic functions based on the maximal Boolean collections contained by these graphs present a great interest for implementation of hybrid circuits (biological and digital, for example).

9. BIO-ALGEBRAS

A class of algebras called bio-algebras were introduced in [10]. They represent, in a certain sense, an extension of Boolean rings. The operations of these algebras are inspired by the works on biological computing. The authors defined and axiomatized the bio-algebra and prove that the finite algebras that satisfy the system of proposed axioms are functionally complete and incorporate the expected properties of the bio-pass and the bio-output. Thus, bio-algebras are primal algebras that are capable of model computation done by bio-circuits: bio-pass (BP), bio-output (BO) and bio-complement (BC).

If $R = \{0, \ldots, r-1\}$ is the set of fundamental values of an r-valued set logic, then every j, $0 \leqslant j \leqslant r-1$ represents a pair substratum-enzyme (assuming that we deal with r distinct enzymes). The bio-circuits mentioned above operate on the power set 2^R. The bio-pass is a two-place function defined by $\mathrm{BP}(X, Y) = X - Y$, that is, by the set difference. The bio-complement is a unary function given by

$BC(X) = R - X$. The bio-output is a two-place function given by

$$\mathrm{BO}(X, Y) = \begin{cases} X & \text{if } Y = \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

The bio-pass and bio-complement functions are Boolean set logic functions while the bio-output is non Boolean. In order to obtain a natural algebra for this type of operations, in [10] a Boolean ring $\mathscr{R} = (I, 0, 1, +, \cdot)$ is introduced, that is a unitary, associative ring that consists of idempotent elements. Such a ring is commutative and of characteristic 2. In other words we have $x + x = 0$ and $x \cdot y = y \cdot x$ for every $x, y \in I$. The functions bc, bo and bp are defined by:

$$\mathrm{bp}(x, y) = x + xy \tag{1}$$

$$\mathrm{bo}(x, y) = \begin{cases} x & \text{if } y = 0, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\mathrm{bc}(x) = 1 + x \tag{3}$$

These functions are refered to as the bio-pass, the bio-output and the bio-complement functions on the Boolean ring R, respectively. It is not difficult to see that the bio-output over a Boolean ring is not a polynomial of the ring.

In [10] the definition of a bio-algebra is introduced. A *bio-algebra* is an algebra $A = (I, 0, 1, \mathrm{bp}, \mathrm{bo})$, where I is the carrier of the algebra, 0, 1 are two zero-ary operations and bp, bo are two binary operations linked by the following axioms:

$$\begin{aligned}
&\mathrm{BP}i) && \mathrm{bp}(x, x) = 0, \\
&\mathrm{BP}ii) && \mathrm{bp}(x, 0) = x, \\
&\mathrm{BP}iii) && \mathrm{bp}(x, \mathrm{bp}(1, y)) = \mathrm{bp}(y, \mathrm{bp}(1, x)), \\
&\mathrm{BP}iv) && \mathrm{bp}(x, \mathrm{bp}(y, z)) = \\
& && \mathrm{bp}(1, \mathrm{bp}(\mathrm{bp}(1, \mathrm{bp}(x, y)), \\
& && \mathrm{bp}(x, \mathrm{bp}(1, z)))), \\
&\mathrm{BP}v) && \mathrm{bp}(\mathrm{bp}(x, y), z) = \mathrm{bp}(\mathrm{bp}(x, z), y), \\
&\mathrm{BO}i) && \mathrm{bo}(x, 0) = x,
\end{aligned}$$

BO*ii*) if $y \neq 0$ then $\text{bo}(1, y) = 0$,
BPOi) $\text{bo}(x, y) = \text{bp}(x, \text{bp}(1, \text{bo}(1, y)))$,

for every $x, y, z \in I$.

The finite algebras that satisfy this system of axioms are functionally complete and incorporate the expected properties of the bio-pass and the bio-output (cf. [10]). They can be regarded as Boolean algebras equipped with the supplementary operation bo.

10. POWER ALGEBRAS

The reference [4] deals with the question of how much structure 2^E inherits from E, and how. The three kinds of structure he considered are algebraic, relational and topological. In each case, a specific power construction is applied in order to obtain the power algebra, power structure and power space, defined over the power set. The constructions work by defining for every function a corresponding power function over 2^E, for any relation, a power relation, and for any topology a power topology. (There is also an important intermediate case, which defines for any relation a power function.)

Algebraic structure is given by functions over the elements of E. An algebra $\langle E, F \rangle$ is just a set E endowed with some n-place functions from a set F. A relational structure is given by relations between the elements of E. A relational structure $\langle E, R \rangle$, often just called a structure, is defined just like an algebra, except that each R element is a n-ary relation. Since any n-place function is a $(n+1)$-ary relation, the concept of structure subsumes that of algebra. More generally, a structure may have both functions and relations. A topological structure is given by a topology, which is a subset of 2^E (containing at least the empty set $\emptyset$ and E itself) constrained to be closed under finite intersections and arbitrary unions. The elements of the topology are called open subsets of E. Note that whereas algebraic and relational structure is given between elements, topological structure is already given in terms of subsets of E.

Each of the power construction in [4] succeeds in lifting significant part (i.e., properties) of the structure of E to 2^E.

A power algebra, as it is conceived, is simply a set logic algebra. Call the algebra $\langle R, F \rangle$ a base algebra and let its power algebra be

$\langle 2^R, 2^F \rangle$ where 2^F is the set of power functions defined over the power set 2^R. In [4] certain natural constructions are defined. For example, if we can multiply any elements x and y of R, then we can also multiply any elements X and Y of 2^R as:

$$X \cdot Y = \{x \cdot y \mid x \in X, y \in Y\}.$$

In fact, we can in this way lift any function f over R to a power function 2^f over 2^R. This example suggests the following definitions:

1. a) For any set R and any function $f: R^n \to R (n \geqslant 1)$, the power function $2^f: (2^R)^n \to 2^R$ is defined by:

$$2^f(X_1, \ldots, X_n) = \{f(x_1, \ldots, x_n) \mid x_i \in X_i \text{ for } 1 \leqslant i \leqslant n\},$$

   for any $X_1, \ldots, X_n \subseteq R$.
2. For any set R and any constant function $f: R^0 \to R$, the power constant function $2^f: (2^R)^0 \to 2^R$ is defined by: $2^f(\emptyset) = \{f(\emptyset)\}$.
3. For any algebra $\langle R, F = \{f_1, \ldots, f_s\} \rangle$, its power algebra is defined by $\langle 2^R, 2^F = \{2^{f_1}, \ldots, 2^{f_s}\} \rangle$.

Note that any power algebra, being based on a power set, is automatically a Boolean algebra under the usual set-theoretic operations. An alternative notion of power algebras incorporates this feature into the definition: for any base algebra $\langle R, F \rangle$, its power algebra is $\langle 2^R, \{\cup, \cap, ^-\} \cup 2^F \rangle$. A study of power algebras defined in this way would then involve interactions between the power functions and the set-theoretic functions. Power algebras, therefore can also be viewed as Boolean algebras with operators, a concept introduced and studied in [44,45].

From the definition of a power function we can say that a power algebra is same as a set logic algebra. However the method of construction of a power algebra (which is not unique) may provide an answer to the following question: how to construct set logic functions of an S-complete set? This question can be stated in another way: if a set F is complete, is 2^F S-complete (or, how can we construct 2^F such that it is S-complete, given $S \subseteq \{\cup, \cap, ^-\}$)? Such questions are not studied in previous papers on set logic algebra.

11. OPEN PROBLEMS AND FURTHER STUDIES IN SET LOGIC ALGEBRA

1. Let $B_k(n)$ be the set of all n-place [B]-functions in P_k. Determine $|B_k(n)|$, i.e., the exact number of such functions in P_k. In [7, 41] only an upper bound was obtained.
2. An interesting question is to determine all clones in set logic containing all Boolean functions (In [6, 11] only maximal clones are found).
3. There are a number of remaining questions concerning S-completeness criteria in set logic if certain functions are considered cheap. Assuming first that S is an arbitrary, nonempty subset of $\{\cup, \cap, ^{-}, \text{constants}\}$, we are going to determine the S-completeness criteria in set logic for subsets S other than C and B, the most interesting cases being $S = \{\cup, \cap\}$, $S = \{^{-}, \text{constants}\}$, and $S = \{^{-}\}$. Second, S-completeness criteria in set logic for any given set S, not necessarily Boolean, is left as an open problem.
4. Listing all classes of functions and classification and enumeration of C-bases and B-bases, C-Sheffer and B-Sheffer functions for r-valued set logic when $r \geqslant 4$ remains also an open problem for further study.
5. Recent developments in universal algebra (see [4]) have stimulated an interest in power structures (power algebras, in particular), that is, in structures defined on power domains where operations result by extending the usual operations on the base domain. It is interesting to explore the possibility of transferring S-completeness results from operations on sets to similar operations on power domains. Results from this area should be of interest for engineers who use functions defined on power sets to model properties of carrier computing systems.
6. There are open problems on approximation theory in set logic. In order to design the most economical hybrid circuits (that is circuits that combine set logic biological or optical components – carrier computing devices – with standard binary electronic components) it is necessary to extend the previous classification of one-place set logic functions based on the number of maximal interpolation classes (see [9]) to n-place functions, and to characterize

almost – Boolean set logic switching functions, that is, functions that have a small number of Boolean interpolation classes. It is interesting to investigate the metric properties of collections (classes) of set logic functions, that is, properties that can be expressed by using suitable metrics. Special attention should be paid to the possibility of metric characterizations of maximal clones of S-complete sets of set logic functions.

7. It is also important to design fast (parallel) algorithms for the search of S-maximal sets in P_k, for the classification and enumeration of functions and S-bases of set logic, and for cataloging set logic functions based on their Boolean interpolation properties. Such algorithms are particularly necessary for $r \geqslant 4$ or $n \geqslant 2$.
8. The study of the complexity of set logic functions asks the question of how to construct them. The complexity of a function is defined as the least number of non Boolean components needed by a set logic circuit that computes the function. The problem is then to find an S-complete set such that the set logic circuit of the function have a minimum number of non Boolean components.

References

[1] Aoki, T., Kameyama, M. and Higuchi, T. (1989). Design of a Highly Parallel Set Logic Network Based on a Bio-Device Model, *Proc. 19th IEEE Int. Symp. Multiple-Valued Logic*, pp. 360–367.

[2] Conrad, M. (1986). The Lure of Molecular Computing, *IEEE Spectrum*, **23**, 55–60.

[3] Kaminuma, T. and Matsumoto, G. (1991). Biocomputers: The Next Generation from Japan, Hitachi Information Research Institute, Japan Institute.

[4] Brink, C. (1992). Power Structures RR121, University of Cape Town, and Ph.D. thesis, Rand Africaans University.

[5] Demetrovics, J., Reischer, C., Simovici, D. A. and Stojmenovic, I. (1994). Enumeration of Functions and Bases of 3-Valued Set Logic under Composition with Boolean Functions, *Proc. 24th IEEE Int. Symp Multiple-Valued Logic*, pp. 164–171.

[6] Demetrovics, J., Ronyai, L., Rosenberg, I. G. and Stojmenovic, I. (1996). On Clones and Maximal Sets in Set Logic Containing All Boolean Functions, *Acta Scientarium Mathematicarum*, **62**, 345–357.

[7] Ngom, A. (1994). Set Logic Foundation of Carrier Computing, M.Sc., Thesis, University of Ottawa, Ottawa.

[8] Kameyama, M. and Higuchi, T. (1988). Prospects of Multiple-Valued Bio-Information Processing Systems, *Proc. 18th IEEE Int. Symp. Multiple-Valued Logic*, pp 237–242.

[9] Reischer, C. and Simovici, D. A. (1991). On the Implementation of Set Logic Non Boolean Switching Functions, *Proc. 21st IEEE Int. Symp. Multiple-Valued Logic*, 1991, pp. 166–172.

[10] Reischer, C. and Simovici, D. A. (1990). Bio-Algebras, *Proc. 20th IEEE Int. Symp. Multiple-Valued Logic*, pp. 48–53.

[11] Simovici, D. A., Stojmenovic, I. and Tosic, R. (1993). Functional Completeness and Weak Completeness in Set Logic, *Proc. 23rd IEEE Int. Symp. Multiple-Valued Logic*, pp. 251–256.

[12] Stojmenovic, I. (1994). Completeness Criteria in Many-Valued Set Logic Under Compositions with Boolean Functions, *Proc. 24th IEEE Int. Symp. Multiple-Valued Logic*, pp. 177–183.

[13] Tosic, R., Stojmenovic, I., Simovici, D. A. and. Reischer, C. (1992). On Set Logic Functions and Boolean Collections, *Proc. 22nd IEEE Int. Symp. Multiple-Valued Logic*, pp. 250–254.

[14] Yuminaka, Y., Aoki, T. and Higuchi, T. (1993). Design of Set Logic Networks for Wave-Parallel Computing, *Proc. 23rd IEEE Int. Symp. Multiple-Valued Logic*, pp. 277–282.

[15] Yuminaka, Y., Aoki, T. and Higuchi, T. (1991). Design of a Set Logic Network Based on Frequency Multiplexing and Its Applications to Image Processing, *Proc. 21st IEEE Int. Symp. Multiple-Valued Logic*, pp. 8–15.

[16] White, K. C. (1981). The Prospects for Multivalued Logic: A Technology and Applications View, *IEEE Trans. Computers*, **C-30**(9), 619–634.

[17] Rudeanu, S. (1974). Booleans Functions and Equations, North-Holland, Amsterdam.

[18] Jablonskij, S. V. (1958). Functional Constructions in k-Valued Logic (Russian), *Trudy Mat. Inst. Steklov*, **51**, 5–142.

[19] Rosenberg, I. G. (1977). Completeness Properties of Multiple-Valued Logic Algebra, in D. C. Rine (ed.), Computer Science and Multiple-Valued Logic: Theory and Application, North-Holland, pp. 144–186.

[20] Post, E. (1921). Introduction to a General Theory of Elementary Propositions, *Amer, J. Math.*, **43**, 163–185.

[21] Kuznecov, A. V. (1961). Structures with Closure and Criteria of Functional Completeness (Russian) *Usp. Math. Nauk*, **16**, **2** **(98)**, 201–202.

[22] Rosenberg, I. G. (1965). La structure des fonctions a plusieurs variables sur un ensemble fini, *C. R. Acad. Sci. Paris*, **A-B**, **260**, 3817,–3819.

[23] Rosenberg, I. G. and Simovici, D. A. (1988). Simovici, Algebraic Aspects of Multiple-Valued Logic, *Proc. 18th IEEE Int. Symp. Multiple-Valued Logic*, pp. 266–275.

[24] Rosenberg, I. G. (1973). The Number of Maximal Closed Classes in the Set of Functions over a Finite Domain, *Journal of Combinatorial Theory* (*A*) **14**, 1–7.

[25] Aoki, T. Dreams for New Device-based superchips, *Proc. 23rd IEEE Int. Symp. Multiple-Values Logic*, 1993, pp. 140–149.

[26] Aoki, T. and Higuchi, T. Impact of Interconnection-Free Biomolecular Computing, *Proc. 23rd IEEE Int. Symp. Multiple-Valued Logic*, 1993, pp. 271–276.

[27] Akoi, T., Kameyama, M. and Higuchi, T. (1992). Interconnection-free Biomolecular Computing, *IEEE Computer*, **25**, November, 41–50.

[28] Aoki, T., Kameyama, M. and Higuchi, T. (1991). Design of Interconneciton Free Biomolecular Computing System, *Proc. 21st IEEE Int. Symp. Multiple-Valued Logic*, pp. 173–180.

[29] Aoki, T., Kameyama, M. and Higuchi, T. (1990). Interconnection-Free Set Logic Network Based on a Bio-Device Model, *IEEE Letters*, **26**, 1015–1016.

[30] Maeda, S., Aoki, T. and Higuchi, T. (1992). Set Logic Network Based on Optical Wavelength Multiplexing, *Proc. 22nd IEEE Int. Symp. Multiple-Valued Logic*, pp. 282–290.

[31] Baker, K. A. and Pixley, A. F. (1975). Polynomial Identities and the Chinese Remainder Theorem for Algebraic Systems, *Math. Z.* **143** **heft** **2**, 165–174.

[32] Brawley, J. V. and Carlitz, L. (1973). A Characterization of the $n \times n$ Matrices over a Finite Field, *Am. Math. Monthly*, **80** **MR** **47** # 5046 670–672.

[33] Froemke, J. A Note on Functionally Complete Algebras with No Non Trivial Subalgebras, reprint, 1971.
[34] Herstein, I. N. Noncommutative Rings, Wiley, New York, 1968.
[35] Lausch, H. and Nobauer, W. Algebra of Polynomials, North-Holland, Amsterdam, 1973.
[36] Maurer, W. D. and Rhodes, G. L. (1965). A Property of Finite Simple Non Abelian Groups, *Proc. Am. Math. Soc*, **16**, p. 552–554.
[37] Muzio, J. C. Concerning Completeness and Abelian Semigroups, *Z. Math, Logik Grundlagen Math.*, 1976
[38] Werner, H. (1973). Finite Simple Non Abelian Groups are Functionally Complete, *Notices Am. Math. Soc.*, **20**(7), ***73T A228.**
[39] Wesselkamper, T. C. Some Completeness Results for Abelian Semi Groups and Groups, *Proc. 4th IEEE Int. Symp. Multiple-Valued Logic*, Morgantown WVa, 1974, pp. 393–400.
[40] Tosic, R., Vojvodic, G., Masulovic, D., Doroslovacki, R. and Rosic, J. (1997). Two Examples of Relative Completeness, *Multiple-Valued Logic – an International Journal*, **2**(1), 67–78.
[41] Ngom, A., Reischer, C. and Stojmenovic, I. Classification of Functions and Enumeration of Bases of Set Logic under Boolean Compositions, *Proc. 25th IEEE Int. Sym. Multiple-Valued Logic*, to appear.
[42] Stojmenovic, I. Some Combinatorial and Algorithmic Problems in Many-Valued Logics, *Inst. Math. Uni. Novi Sad*, 1987.
[43] Scognamiglio, G. (1961). Interpolazione per le funzioni algebriche booleane, *Giron. Mat. Bataglini*, **89**, 14–41.
[44] Jonsson, B. and Tarski, A. (1952). Boolean Algebras with Operators II, *American Journal of Mathematics*, **74**, 127–167.
[45] Jonsson, B. and Tarski, A. (1951). Boolean Algebras with Operators I, *American Journal of Mathematics*, **73**, 891–939.
[46] Higuchi, T. and Aoki, T. (1996). A Multiplex Computing Paradigm, *Proceedings of IIZUKA, Japan*, pp. 95–100.
[47] Kurosaki, M., Takagi, N. and Nakashima, K. (1996). Implementation of Set-Valued Logic Functions by Hybrid Structure, *Notes on Multiple-Valued Logic in Japan*, vol. 19.
[48] Nakanishi, T., Aoki, T. and Higuchi, T. (1996). Set-Valued Logic Networks using pseudonoise Sequences, *Notes on Multiple-Valued Logic in Japan*, **19**(14), pp. 1–6.
[49] Pogosyan, G. and Nozaki, A. (1996). Completeness and Closed Classes of Boolean Functions under Composition with Feedback Loops, *Multiple-Valued Logic – an International Journal*, **1**(4), pp. 285–306.
[50] Takagi, N., Nakamura, Y. and Nakashima, K. (1997). Set-Valued Functions Monotonic in the Set-Theoretical Inclusion, *Multiple-Valued Logic – an International Journal*, to appear.
[51] Takagi, N., Nakamura, Y. and Nakashima, K. (1997). Set-Valued Functions and Regularity, *Proc. 27th IEEE ISMVL*, to appear.
[52] Simovici, D. A. and Reischer, C. (1996). Several Remarks on the Complexity of Set-Valued Switching Functions, *Proc. 26th IEEE ISMVL*, pp. 166–170.
[53] Rosenberg, I. G. (1996). An Algebraic Approach to Hyperalgebras. *Proc. 26th IEEE ISMVL*, pp. 203–207.
[54] Romov, B. A. (1996). Polynomial Completeness in Finite Boolean Algebras. *Proc. 26th IEEE ISMVL*, pp. 262–266.
[55] Ngom, A., Reischer, C., Simovici, D. A. and Stojmenovic, I. (1997). Completeness Criteria in Set-Valued Logic Under Compositions with Union and Intersection, *Proc. 27th IEEE ISMVL*, to appear.
[56] Romov, B. A. (1997). Hyperclones on a Finite Set, *Proc. 27th IEEE ISMVL*, to appear.