On Rine's View of Boolean Algebras

by

George Epstein
Computer Science Department
Indiana University
Bloomington, Indiana 47405

TECHNICAL REPORT NO. 102

ON RINE'S VIEW OF BOOLEAN ALGEBRAS

GEORGE EPSTEIN

DECEMBER, 1980

On Rine's View of Boolean Algebras

George Epstein

Indiana University

Bloomington, Indiana 47405

In [6] David Rine takes the view that there may be more connections between 2-valued logic and n-valued logic than we commonly suppose. He considers the n-valued case when n=2, then discusses the generalization of results for the 2-valued case to the n-valued case. His main example is the generalization of the symmetric difference x + y, which in the 2-valued case is the ring sum, the binary addition or exclusive-or of two binary variables x,y.

There has been some recent question about the relative character of investigations into 2-valued logic and n-valued logic [5]. It may be that an extension of Rine's observations will help to clarify this issue. The discussion which follows deliberately enters into major subject areas of switching theory with this in mind. These major subject areas include minimization, state assignment, and special functions.

It will be convenient to use the same n-valued system used by Rine, but it will be sufficient for purposes of illustration to limit the value of n to n=3. The resulting contrast between the binary case n=2 and ternary case n=3 is easily extended to higher values of n. The equation version of this system first appeared in [1]. An abbreviated form of this system for n=3 appeared in [2].

A common starting point for these systems is provided by lattices which are distributive and bounded. The zero bound is denoted by $\underline{z}$, the unit bound is denoted by $\underline{u}$. In the 3-valued case, there is a third intermediate constant denoted by $\underline{e}$. The lattice operations join $(x \vee y)$, meet $(x \wedge y)$ correspond to the logic operators OR, AND, respectively, each being distributive with respect to the other. Details about these operations within bounded distributive lattices may be found through the references.

The case n=2 and Boolean algebras arises through the introduction of an unary operator and two axioms:

$$\bar{x} \wedge x = \underline{z}$$

$$\bar{x} \vee x = \underline{u}.$$

The complementation operation $\bar{x}$ corresponds to the logic operator NOT. The first rule corresponds to the law of contradiction, the second to the law of excluded middle.

Before proceeding to the case n=3, it will be helpful to restate these two rules using two unary operators, one for complementation (C_0) and one for identity (C_1). The result is:

$$x = C_1(x)$$

$$C_0(x) \wedge C_1(x) = \underline{z}$$

$$C_0(x) \vee C_1(x) = \underline{u}.$$

Each of the 3 rules in this restatement will have a generalization for the case n=3. However, this will not suffice for n=3. Certain provable properties for n=2 such as $C_0(\underline{z}) = \underline{u}$ and $C_0(x \wedge y) = C_0(x) \vee C_0(y)$ must be included among the axioms when n=3. The result which follows shows a total of 10 axioms appearing within 5 groupings. The first 3 groupings generalize the above restatement; the second 2 groupings give the required additional properties.

$$x = (\underline{e} \;\; C_1(x)) \vee C_2(x)$$

$$C_0(x) \wedge C_1(x) = \underline{z};\; C_0(x) \wedge C_2(x) = \underline{z}\;;\; C_1(x) \wedge C_2(x) = \underline{z}$$

$$C_0(x) \vee C_1(x) \vee C_2(x) = \underline{u}$$

$$C_0(\underline{z}) = \underline{u}\;;\; C_1(\underline{e}) = \underline{u}\;;\; C_2(\underline{u}) = \underline{u}$$

$$C_0(x \wedge y) = C_0(x) \vee C_0(y)\;;\; C_2(x \vee y) = C_2(x) \vee C_2(y).$$

As the 2-valued system has illustration with 2-valued switching functions and truth tables, so can the 3-valued system be illustrated with 3-valued switching functions and corresponding truth tables. In the table which follows, the constant entries might alternately be written as logic values F (FALSE), I (INTERMEDIATE), T (TRUE).

x	$\underline{z}$	$\underline{e}$	$\underline{u}$	$C_0(x)$	$C_1(x)$	$C_2(x)$
	0	1	2	2	0	0
	0	1	2	0	2	0
	0	1	2	0	0	2

Many properties which hold in this system can be confirmed through the use of such tables. These include each of the axioms above, and further properties such as $C_0(x \vee y) = C_0(x) \wedge C_0(y)$ and $C_2(x \wedge y) = C_2(x) \wedge C_2(y)$.

For what follows, it is important to observe that C_0 is a pseudo-complement operator which generalizes 2-valued complementation. It has just been notified that C_0 satisfies both DeMorgan's laws. The table below shows all three operators which complements the extremal value 0,2. The operator N is the strong 3-valued negation of Lukasiewicz---it is given in this system by $N(x) = (\underline{e} \wedge C_1(x)) \vee C_0(x)$.

x	$C_0(x)$	$N(x)$	$C_0(C_2(x))=A_1(x)$
0	2	2	2
1	0	1	2
2	0	0	0

This last table is helpful in considering the generalization of results in the 2-valued case when the complementation operation $\overline{x}$ is involved. In particular, conclusions which are reached for one of these three operators need not extend to the other operators. Thus, for example, while both of DeMorgan's laws hold for each of these three operators, the law of double negation holds only for the operator N. It is easy to verify that $N(N(x)) = x$. However, $C_0(C_0(C_0(x)))=C_0(x)$ with a similar rule of triple negation for the third operator.

Rine begins by considering the generalization of the 2-valued symmetric difference $x + y = (\overline{x} \wedge y) \vee (x \wedge \overline{y})$ to the 3-valued $x + y = (C_0(x) \wedge y) \wedge y) \vee (x \wedge C_0(y))$. He shows that the associative feature of the former $w + (x + y) = (w + x) \quad y$ is <u>not</u> preserved by the latter generalization. He mentions further that associativity is also lost using the 3-valued generalization $x + y = (C_0(x) \wedge C_2(y)) \vee (C_2(x) \wedge C_0(y))$. It is straightforward to use the above tables to verify these statements. For each generalization just stated, associativity is lost when the variables assume the values w=1, x=1, y=2.

Other generalizations might be considered. The generalization $x +_3 y = (A_1(x) \wedge y) \vee (x \wedge A_1(y)$ is another candidate which is not associative. This can be seen by using the values w=1, x=2, y=2. The generalization $x +_4 y = (N(x) \wedge y) \vee (x \wedge N(y))$ succeeds in being associative, but may not be considered successful in other regards. There is no solution s to the equation $1 +_4 s = 0$, so that 1 does not have an additive inverse and the operation $+_4$ does not yield an additive group. Finally, the generalization

$$x +_5 y$$

$$= (\underline{e} \wedge ((C_0(x) \wedge C_1(y)) \vee (C_1(x) \wedge C_0(y)) \vee (C_2(x) \wedge C_2(y)))$$
$$\vee (C_0(x) \wedge C_2(y)) \vee (C_2(x) \wedge C_0(y)) \vee (C_1(x) \wedge C_1(y))$$

is another operation which is associative. This latter operation also yields an additive Abelian group, and corresponds to the ternary addition of two ternary variables x,y, where the ternary digits are 0, 1, 2.

Thus it would be difficult to generalize work on the 2-valued associative symmetric difference using any of the first three of the above five candidates. Either of the last two candidates, however, could lead to success, depending on the nature of the problem at hand.

There is a somewhat different situation for the problem of state assignments. It is known that this problem is sensitive to the kind of stable devices under use and the nature of their defining equations. Hence the state assignment problem for flip-flops which are inherently bistable without easy generalization beyond the case n=2, such as JK flip-flops, is essentially a

2-valued problem rather than a multiple-valued problem. Other flip-flops, such as D flip-flops, have defining equations which are not easily given in n-valued form as in 2-valued form. They are easily programmed in an integer format restricting the integer values to n=0,1, ... , n-1. The naming or assignment for each state can use the binary digits 0,1 or the ternary digits 0,1,2. The actual design procedure for the assignment of names can apply to either the binary or ternary case. Here the assignment of names to each state could as well be done for the case n=3 as for the case n=2. The binary formulation is usually most convenient because of its conciseness.

With regard to minimization, it will be enough to give two examples of simplification problems. Each of these are easy, well known examples from 2-valued switching theory.

First, consider the 2-valued simplification $\overline{x} \vee (x \wedge y) = \overline{x} \vee y$. Verification by truth tables is immediate. The simplification follows directly from distributivity: $\overline{x} \vee (x \wedge y) = (\overline{x} \vee x) \wedge (\overline{x} \vee y) = \underline{u} \wedge (\overline{x} \vee y) = \overline{x} \vee y$. Evident 3-valued expressions are below:

$$C_0(x) \vee (x \wedge y)$$

$$N(x) \vee (x \wedge y)$$

$$A_1(x) \vee (x \wedge y).$$

Neither of the first two has such simplification. The last does simplify: $A_1(x) \vee (x \wedge y) = (A_1(x) \vee x) \wedge (A_1(x) \vee y) = \underline{u} \wedge (A_1(x) \vee y) = A_1(x) \vee y$. This leads to a modification of the first expression which allows a similar type of simplification. The expression $C_0(x) \vee (C_0(C_0(x)) \wedge y)$ may be obtained by replacing

$\overline{x}$ with $C_0(x)$ and x with $C_0(C_0(x))$. This new expression has the same type of simplification to $C_0(x) \vee y$.

Second, consider the 2-valued simplification $(w \wedge y) \vee (x \wedge \overline{y}) \vee (w \wedge x) = (w \wedge y) \vee (x \wedge \overline{y})$. Verification by truth tables is again immediate. Simplification follows from the partial ordering whereby $a \vee b = a$ stands for $b \leq a$ stands for $a \wedge b = b$. Using the right hand expression for a and $w \wedge x$ for b, $((w \wedge y) \vee (x \wedge \overline{y}))\ (w \wedge x) = (w \wedge x \wedge y) \vee (w \wedge x \wedge \overline{y}) =$ $(w \wedge x) \wedge (y \vee \overline{y}) = w \wedge x$. It is clear from this simplification that again replacement of $\overline{y}$ with either $C_0(y)$ or $N(y)$ will fail to generalize, but replacement of $\overline{y}$ with $A_1(y)$ will. It is again possible to create a new expression involving C_0 and having a parallel simplification. This 3-valued simplification is given below:

$$(v \wedge C_2(y)) \vee (w \wedge C_1(y)) \vee (x \wedge C_0(y)) \vee (v \wedge w \wedge x)$$
$$= (v \wedge C_2(y)) \vee (w \wedge C_1(y)) \vee (x \wedge C_0(y)).$$

Thus the possibilities for n-valued simplifications appear quite rich. The bounds on such simplifications can become severe, however, as the number of variables and the value of n increases. For example, Karnaugh maps for 6 or fewer variables are feasible in the binary case n=2, but for higher values of n are only feasible for 3 or fewer variables.

Finally, there are important classes of switching functions in the 2-valued case which can generalize to the n-valued case. The symmetric difference may be considered as an example of an arithmetic function. Other important functions are the symmetric, unate, threshold, and fan-out free functions. To illustrate, consider the two variable symmetric functions. The 2-valued symmetric difference is one such function. For n=2 there are three fundamental symmetric functions:

$$f_0 = \overline{x} \wedge \overline{y}$$

$$f_1 = (\overline{x} \wedge y) \vee (x \wedge \overline{y})$$

$$f_2 = x \wedge y.$$

Note that generalizations of f_1 alone were discussed in a different context at the beginning of this paper. The interest here is in developing a decomposition theory for symmetric functions, and so generalizations which occur here may be quite different. There are many different generalizations for f_0, f_1, f_2, and many of these have occurred within the literature. The 2-valued decomposition theory in [0] was recently generalized to the n-valued case in [3, 4]. The difficulty in making this generalization lay with the factorization theory, not with the generalization of the fundamental symmetric functions. Nevertheless, it is instructive to note the generalization to the case n=3. There are 6 resulting fundamental symmetric functions, and they are listed below. The decompositions are from [3]. Each fundamental symmetric function is a conjunction of terms C_i, for certain i, where the arguments in two variables are $x \vee y$ and $x \wedge y$. This generalizes the 2-valued result in [0], where a similar decomposition occurs. These 6 functions and their decompositions are:

$$C_0(x) \wedge C_0(y) = C_0(x \vee y)$$

$$C_1(x) \wedge C_1(y) = C_1(x \wedge y) \wedge C_1(x \vee y)$$

$$C_2(x) \wedge C_2(y) = C_2(x \wedge y)$$

$$(C_0(x) \wedge C_1(y)) \vee (C_1(x) \wedge C_0(y)) = C_0(x \wedge y) \wedge C_1(x \vee y)$$

$$(C_1(x) \wedge C_2(y)) \vee (C_2(x) \wedge C_1(y)) = C_1(x \wedge y) \wedge C_2(x \vee y)$$

$$(C_0(x) \wedge C_2(y)) \vee (C_2(x) \wedge C_0(y)) = C_0(x \wedge y) \wedge C_2(x \vee y).$$

While the whole set of 6 is the generalization of the previous set of 3, the first 3 of these 6 appear to generalize f_0 and f_2, and the latter 3 appear to generalize f_1. The last was already discussed in the earlier context.

Other generalizations using other operators can lead to decompositions which are quite complex or lengthy. For any given class of functions, it may be a difficult matter to determine the best generalization beyond the case n=2.

REFERENCES

0. Epstein, G., Synthesis of Electronic Circuits for Symmetric Functions, IRE Trans. on Elec. Computers, EC-7, #1, 57-59, March, 1958.

1. Epstein, G., An Equational Axiomatization for the Disjoint System of Post Algebras, IEEE Trans. on Computers, C-22, #4, 422-423, April 1973.

2. Epstein, G., Frieder, G., Rine, D., The Development of Multiple-valued Logic as Related to Computer Science, COMPUTER, 20-32, September 1974.

3. Epstein, G., Miller, D. M., Muzio, J., Some Preliminary Views on the General Synthesis of Electronic Circuits for Symmetric and Partially Symmetric Functions, Proc. 7th Int'l Symp. on Multiple-valued Logic, University of North Carolina, Charlotte, 29-34, May, 1977.

4. __________, Selecting Don't-Care Sets for Symmetric Many-valued Functions, Proc. 10th Int'l Symp. on Multiple-valued Logic, Northwestern University, Evanston, 219-225, June, 1980.

5. Newsletter of the IEEE Technical Committee on Multiple-valued Logic, January, 1981.

6. Rine, D. C., There is More to Boolean Algebra Than You Would Have Thought, Reports on Math. Logic, 2, 25-32, 1974.

On Rine's View of Boolean Algebras

George Epstein
Indiana University
Bloomington, Indiana 47401

In [6] David Rine takes the view that there may be more connections between 2-valued logic and n-valued logic than we commonly suppose. He considers the n-valued case when n=2, then discusses the generalization of results for the 2-valued case to the n-valued case. His main example is the generalization of the symmetric difference $x \oplus y$, which in the 2-valued case is the ring sum, the binary addition or exclusive-or of two binary variables x,y.

There has been some recent question about the relative character of investigations into 2-valued logic and n-valued logic [5]. It may be that an extension of Rine's observations will help to clarify this issue. The discussion which follows deliberately enters into major subject areas of switching theory with this in mind. These major subject areas include minimization, state assignment, and special functions.

It will be convenient to use the same n-valued system used by Rine, but it will be sufficient for purposes of illustration to limit the value of n to n=3. The resulting contrast between the binary case n=2 and ternary case n=3 is easily extended to higher values of n. The equational version of this system first appeared in [1]. An abbreviated form of this system for n=3 appeared in [2].

A common starting point for these systems is provided by lattices which are distributive and bounded. The zero bound is denoted by $\underline{z}$, the unit bound is denoted by $\underline{u}$. In the 3-valued case, there is a third intermediate constant denoted by $\underline{e}$. The lattice operations join $(x \vee y)$, meet $(x \wedge y)$ correspond to the logic operators OR, AND, respectively, each being distributive with respect to the other. Details about these operations within bounded distributive lattices may be found through the references.

The case n=2 and Boolean algebras arises through the introduction of an unary operator and two axioms:

$$\bar{x} \wedge x = \underline{z}$$
$$\bar{x} \vee x = \underline{u}.$$

The complementation operation $\bar{x}$ corresponds to the logic operator NOT. The first rule corresponds to the law of contradiction, the second to the law of excluded middle.

Before proceeding to the case n=3, it will be helpful to restate these two rules using two unary operators, one for complementation (C_0) and one for identity (C_1). The result is:

$$x = C_1(x)$$
$$C_0(x) \wedge C_1(x) = \underline{z}$$
$$C_0(x) \vee C_1(x) = \underline{u}.$$

Each of the 3 rules in this restatement will have a generalization for the case n=3. However, this will not suffice for n=3. Certain provable properties for n=2 such as $C_0(\underline{z}) = \underline{u}$ and $C_0(x \wedge y) = C_0(x) \vee C_0(y)$ must be included among the axioms when n=3. The result which follows shows a total of 10 axioms appearing within 5 groupings. The first 3 groupings generalize the above restatement; the second 2 groupings give the required additional properties.

$$x = (\underline{e} \;\; C_1(x)) \vee C_2(x)$$

$$C_0(x) \wedge C_1(x) = \underline{z};\; C_0(x) \wedge C_2(x) = \underline{z}\;;\; C_1(x) \wedge C_2(x) = \underline{z}$$

$$C_0(x) \vee C_1(x) \vee C_2(x) = \underline{u}$$

$$C_0(\underline{z}) = \underline{u}\;;\; C_1(\underline{e}) = \underline{u}\;;\; C_2(\underline{u}) = \underline{u}$$

$$C_0(x \wedge y) = C_0(x) \vee C_0(y)\;;\; C_2(x \vee y) = C_2(x) \vee C_2(y).$$

As the 2-valued system has illustration with 2-valued switching functions and truth tables, so can the 3-valued system be illustrated with 3-valued switching functions and corresponding truth tables. In the table which follows, the constant entries might alternately be written as logic values F (FALSE), I (INTERMEDIATE), T (TRUE).

x	$\underline{z}$	$\underline{e}$	$\underline{u}$	$C_0(x)$	$C_1(x)$	$C_2(x)$
	0	1	2	2	0	0
	0	1	2	0	2	0
	0	1	2	0	0	2

Many properties which hold in this system can be confirmed through the use of such tables. These include each of the axioms above, and further properties such as $C_0(x \vee y) = C_0(x) \wedge C_0(y)$ and $C_2(x \wedge y) = C_2(x) \wedge C_2(y)$.

For what follows, it is important to observe that C_0 is a pseudo-complement operator which generalizes 2-valued complementation. It has just been noted that C_0 satisfies both DeMorgan's laws. The table below shows all three operators which complements the extremal value 0,2. The operator N is the strong 3-valued negation of Lukasiewicz---it is given in this system by $N(x) = (\underline{e} \wedge C_1(x)) \vee C_0(x)$.

x	$C_0(x)$	$N(x)$	$C_0(C_2(x))=A_1(x)$
0	2	2	2
1	0	1	2
2	0	0	0

This last table is helpful in considering the generalization of results in the 2-valued case when the complementation operation $\overline{x}$ is involved. In particular, conclusions which are reached for one of these three operators need not extend to the other operators. Thus, for example, while both of DeMorgan's laws hold for each of these three operators, the law of double negation holds only for the operator N. It is easy to verify that $N(N(x)) = x$. However, $C_0(C_0(C_0(x)))=C_0(x)$ with a similar rule of triple negation for the third operator.

Rine begins by considering the generalization of the 2-valued symmetric difference $x \oplus y = (\overline{x} \wedge y) \vee (x \wedge \overline{y})$ to the 3-valued $x \underset{1}{\triangle} y = (C_0(x) \wedge y) \vee (x \wedge C_0(y))$. He shows that the associative feature of the former $w \oplus (x \oplus y) = (w \oplus x) \oplus y$ is not preserved by the latter generalization. He mentions further that associativity is also lost using the 3-valued generalization $x \underset{2}{\triangle} y = (C_0(x) \wedge C_2(y)) \vee (C_2(x) \wedge C_0(y))$. It is straightforward to use the above tables to verify these statements. For each generalization just stated, associativity is lost when the variables assume the values w=1, x=1, y=2.

Other generalizations might be considered. The generalization $x \underset{3}{\triangle\!\!\!+} y = (A_1(x) \wedge y) \vee (x \wedge A_1(y)$ is another candidate which is not associative. This can be seen by using the values $w=1$, $x=2$, $y=2$. The generalization $x \underset{4}{\triangle\!\!\!+} y = (N(x) \wedge y) \vee (x \wedge N(y))$ succeeds in being associative, but may not be considered successful in other regards. There is no **solution** s to the equation $1 \underset{4}{\triangle\!\!\!+} s = 0$, so that 1 does not have an additive inverse and the operation $\underset{4}{\triangle\!\!\!+}$ does not yield an additive group. Finally, the generalization $x \underset{5}{\triangle\!\!\!+} y$

$$= (\underline{e} \wedge ((C_0(x) \wedge C_1(y)) \vee (C_1(x) \wedge C_0(y)) \vee (C_2(x) \wedge C_2(y)))$$

$$\vee (C_0(x) \wedge C_2(y)) \vee (C_2(x) \wedge C_0(y)) \vee (C_1(x) \wedge C_1(y))$$

is another operation which is associative. This latter operation also yields an additive Abelian group, and corresponds to the ternary addition of two ternary variables x,y, **where** the ternary digits are 0,1,2.

Thus **it would** be **difficult** to generalize work on the 2-valued associative symmetric difference using any of the first three of the above five candidates. Either of the last two candidates, however, **could** lead to success, depending on the nature of the problem at hand.

There is a somewhat **different** situation for the problem of state assignments. It is known that this problem is sensitive to the kind of stable devices under use and the nature of their defining equations. Hence the state assignment problem for flip-flops which are inherently bistable without easy generalization beyond the case n=2, such as JK flip-flops, is essentially a 2-valued problem rather than a multiple-valued problem. Other flip-flops, such as D flip-flops, have defining equations which are as easily given in n-valued form as in 2-valued form. They are easily programmed in an integer format restricting the integer values to n=0,1, ··· , n-1. The naming or assignment for each state can use the binary digits 0,1 or the ternary digits 0,1,2. The actual design procedure for the assignment of names can apply to either the binary or ternary case. Here the assignment of names to each state could as well be done for the case n=3 as for the case n=2. The binary formulation is usually most convenient because of its conciseness.

With regard to minimization, it will be enough to give two examples of simplification problems. Each of these are easy, well known examples from 2-valued switching theory.

First, consider the 2-valued simplification $\overline{x} \vee (x \wedge y) = \overline{x} \vee y$. Verification by truth tables is immediate. The simplification follows directly from distributivity: $\overline{x} \vee (x \wedge y) = (\overline{x} \vee x) \wedge (\overline{x} \vee y)$ $= \underline{u} \wedge (\overline{x} \vee y) = \overline{x} \vee y$. Evident 3-valued expressions are below:

$$C_0(x) \vee (x \wedge y)$$

$$N(x) \vee (x \wedge y)$$

$$A_1(x) \vee (x \wedge y).$$

Neither of the first two has such simplification. The last does simplify: $A_1(x) \vee (x \wedge y) = (A_1(x) \vee x) \wedge (A_1(x) \vee y)$ $= \underline{u} \wedge (A_1(x) \vee y) = A_1(x) \vee y$. This leads to a modification of the first expresion which allows a similar type of simplification. The expression $C_0(x) \vee (C_0(C_0(x)) \wedge y)$ may be obtained by replacing $\overline{x}$ with $C_0(x)$ and x with $C_0(C_0(x))$. This new expression has the same type of simplification to $C_0(x) \vee y$.

Second, consider the 2-valued simplification $(w \wedge y) \vee (x \wedge \overline{y}) \vee (w \wedge x) = (w \wedge y) \vee (x \wedge \overline{y})$. Verification by truth tables is again immediate. Simplification follows from the partial ordering whereby $a \vee b = a$ stands for $b \leq a$ stands for

$a \wedge b = b$. Using the right hand expression for a and $w \wedge x$ for b,
$((w \wedge y) \vee (x \wedge \overline{y})) \wedge (w \wedge x) = (w \wedge x \wedge y) \vee (w \wedge x \wedge \overline{y})$
$= (w \wedge x) \wedge (y \vee \overline{y}) = w \wedge x$. It is clear from this simplification that again replacement of $\overline{y}$ with either $C_0(y)$ or $N(y)$ will fail to generalize, but replacement of $\overline{y}$ with $A_1(y)$ will. It is again possible to create a new expression involving C_0 and having a parallel simplification. This 3-valued simplification is given below:

$$(v \wedge C_2(y)) \vee (w \wedge C_1(y)) \vee (x \wedge C_0(y)) \vee (v \wedge w \wedge x)$$

$$= (v \wedge C_2(y)) \vee (w \wedge C_1(y)) \vee (x \wedge C_0(y)).$$

Thus the possibilities for n-valued simplifications appear quite rich. The bounds on such simplifications can become severe, however, as the number of variables and the value of n increases. For example, Karnaugh maps for 6 or fewer variables are feasible in the binary case n=2, but for higher values of n are only feasible for 3 or fewer variables.

Finally, there are important classes of switching functions in the 2-valued case which can generalize to the n-valued case. The symmetric difference may be considered as an example of an arithmetic function. Other important functions are the symmetric, unate, threshold, and fan-out free functions. To illustrate, consider the two variable symmetric functions. The 2-valued symmetric diffence is one such function. For n=2 there are three fundamental symmetric functions:

$$f_0 = \overline{x} \wedge \overline{y}$$

$$f_1 = (\overline{x} \wedge y) \vee (x \wedge \overline{y})$$

$$f_2 = x \wedge y.$$

Note that generalizations of f_1 alone were discussed in a different context at the beginning of this paper. The interest here is in developing a decomposition theory for symmetric functions, and so generalizations which occur here may be quite different. There are many different generalizations for f_0, f_1, f_2, and many of these have occurred within the literature. The 2-valued decomposition theory in [0] was recently generalized to the n-valued case in [3,4]. The difficulty in making this generalization lay with the factorization theory, not with the generalization of the fundamental symmetric functions. Nevertheless, it is instructive to note the generalization to the case n=3. There are 6 resulting fundamental symmetric functions, and they are listed below. The decompositions are from [3]. Each fundamental symmetric function is a conjunction of terms C_i, for certain i, where the arguments in two variables are $x \vee y$ and $x \wedge y$. This generalizes the 2-valued result in [0], where a similar decomposition occurs. These 6 functions and their decompositions are:

$$C_0(x) \wedge C_0(y) = C_0(x \vee y)$$

$$C_1(x) \wedge C_1(y) = C_1(x \wedge y) \wedge C_1(x \vee y)$$

$$C_2(x) \wedge C_2(y) = C_2(x \wedge y)$$

$$\big(C_0(x) \wedge C_1(y)\big) \vee \big(C_1(x) \wedge C_0(y)\big) = C_0(x \wedge y) \wedge C_1(x \vee y)$$

$$\big(C_1(x) \wedge C_2(y)\big) \vee \big(C_2(x) \wedge C_1(y)\big) = C_1(x \wedge y) \wedge C_2(x \vee y)$$

$$\big(C_0(x) \wedge C_2(y)\big) \vee \big(C_2(x) \wedge C_0(y)\big) = C_0(x \wedge y) \wedge C_2(x \vee y).$$

While the whole set of 6 is the generalization of the previous set of 3, the first 3 of these 6 appear to generalize f_0 and f_2, and the latter 3 appear to generalize f_1. The last was already discussed in the earlier context.

Other generalizations using other operators can lead to decompositions which are quite complex or lengthy. For any given class of functions, it may be a difficult matter to determine the best generalization beyond the case n=2.

REFERENCES

[0]. Epstein, G., Synthesis of Electronic Circuits for Symmetric Functions, IRE Trans. on Elec. Computers, EC-7, #1, 57-59, March, 1958.

[1]. Epstein, G., An Equational Axiomatization for the Disjoint System of Post Algebras, IEEE Trans. on Computers, C-22, #4, 422-423, April 1973.

[2]. Epstein, G., Frieder, G., Rine, D., The Development of Multiple-valued Logic as Related to Computer Science, COMPUTER, 20-32, September 1974.

[3]. Epstein, G., Miller, D.M., Muzio, J., Some Preliminary Views on the Genreal Synthesis of Electronic Circuits for Symmetric and Partially Symmetric Functions, Proc. 7th Int'l Symp. on Multiple-valued Logic, University of North Carolina, Charlotte, 29-34, May, 1977.

[4] ____________, Selecting Don't-Care Sets for Symmetric Many-valued Functions, Proc. 10th Int'l Symp. on Multiple-valued Logic, Northwestern University, Evanston, 219-225, June, 1980.

[5]. Newsletter of the IEEE Technical Committee on Multiple-valued Logic, January, 1981.

[6]. Rine, D.C., There is More to Boolean Algebra Than You Would Have Thought, Reports on Math. Logic, 2, 25-32, 1974.