ON THE SIMPLIFICATION
OF
BOOLEAN POLYNOMIALS

By

OREN WESLEY DIXON

A THESIS

submitted to

OREGON STATE UNIVERSITY

in partial fulfillment of
the requirements for the
degree of

MASTER OF SCIENCE

June 1962

APPROVED: [redacted]

Professor of Mathematics

In Charge of Major

[redacted]

Chairman of Department of Mathematics

[redacted]

Chairman of School Graduate Committee

[redacted]

Dean of Graduate School

Date thesis is presented April 10, 1962

Typed by Marlene McDonald

ACKNOWLEDGMENT

The author wishes to express his appreciation to Dr. H. E. Goheen for his encouragement and advice.

On the Simplification of Boolean Polynomials

As a result of Shannon's application of Boolean algebra to relay and switching circuits (6,p.713-723) the problem of simplifying Boolean polynomials has become important in that branch of engineering which deals with the design of such circuits. The problem has both commercial and technical significance, for in simplifying the representation of a given Boolean polynomial, the designer is able to reduce the number of components which occur in the corresponding logical circuit.

In a practical sense the problem is not solved. Indeed, if the number of independent variables is even moderate, for example, if the function is defined on ten independent variables, the time required for a modern high speed digital computer to determine a simplest representation is prohibitive. (2,p.210-212)

In the following pages, we first construct a model of a finite Boolean algebra of n independent variables. We then define the problem in terms of the model. We develop an algorithm which in theory will solve the problem. From the derivation of this algorithm we obtain methods which, for certain types of functions, lead to efficient solutions. In addition to giving a theoretical solution to the problem, the paper will reveal those difficulties which give explanation for the fact that, in the practical sense, the problem remains unsolved.

Our first goal is to construct a model of a finite Boolean algebra, of n independent variables.

<u>Definition 1</u>. Let S denote the set whose elements are 1, and 0, where 1 and 0 are real numbers.

<u>Definition 2</u>. We define two binary operations, $+$ and $\cdot$, and one unary operation $'$ on S:

$$\begin{array}{c|c|c|}
+ & 0 & 1 \\ \hline
0 & 0 & 1 \\ \hline
1 & 1 & 1
\end{array}
\qquad
\begin{array}{c|c|c}
\cdot & 0 & 1 \\ \hline
0 & 0 & 0 \\ \hline
1 & 0 & 1
\end{array}
\qquad
\begin{array}{l}
1' = 0 \\
0' = 1
\end{array}$$

S, along with the operations of Definition 2, is a Boolean algebra.

<u>Definition 3</u>. For each positive integer n, define a 2^n by n matrix D_n as follows:

$$D_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Having defined D_n, define D_{n+1} by:

$$D_{n+1} = \left(\begin{array}{c|c} \theta & D_n \\ \hline I & D_n \end{array}\right),$$ where θ is a 2^n by 1 column matrix, each element of which is 0, and where I is a 2^n by 1 column matrix, each element of which is 1.

<u>Definition 4</u>. For each positive integer n, let $\tilde{D}_n$ denote that set whose elements are the row n-tuples of the matrix D_n. Let $\tilde{B}_n$ denote that class of functions f, such that f is defined on $\tilde{D}_n$, and f maps $\tilde{D}_n$ into S.

Lemma 1.

Let A be any set of cardinality K, where K is any natural number. Let F denote that class of functions g, such that g maps A into S. Then the cardinality of F is 2^K.

Proof: For $n = 1$, there are two admissible functions which map a singleton into S. If a is the element of the singleton, the two functions are $a \rightarrow 0$, and $a \rightarrow 1$. Assume the proposition true for the natural number K. Let A be any set of cardinality $K + 1$, say $A = \{1, 2, \ldots K, K + 1\}$. For each function g defined on $\{1, 2, \ldots K\}$, which maps $\{1, 2, \ldots K\}$ into S, there correspond exactly two functions, h, and t, which map A into S. They are defined by:

$$h(j) = g(j) \quad \text{if} \quad 1 \leq j < K + 1$$
$$h(K + 1) = 0$$

$$t(j) = g(j) \quad \text{if} \quad 1 \leq j < K + 1$$
$$t(K + 1) = 1$$

By the induction hypothesis there are 2^K such functions g, defined on $\{1, 2, \ldots K\}$. Hence there are $2 \cdot 2^K = 2^{K+1}$ functions defined on A which map A into S.

Lemma 2.

The cardinality of $\tilde{B}_n$ is $2^{(2^n)}$.

Proof: By induction, it follows that no two rows of the matrix D_n are identical. Hence the cardinality of $\tilde{D}_n$ is 2^n. By Definition 4, and Lemma 1, the cardinality of $\tilde{B}_n$ is $2^{(2^n)}$.

<u>Definition 5</u>. We define two binary operations, $+$, and $\cdot$, and a unary operation $'$ on $\tilde{B}_n$

For each f and g in $\tilde{B}_n$ we define $f + g$ by $f + g\ (P) = f\ (P) + g\ (P)$, for each P in $\tilde{D}_n$.

We define $f \cdot g$ by $f \cdot g\ (P) = f\ (P) \cdot g\ (P)$ for each P in $\tilde{D}_n$.

We define f' by $f'\ (P) = [f\ (P)]'$ for each P in $\tilde{D}_n$ and f in $\tilde{B}_n$.

Comment: $\tilde{B}_n$, along with the operations of Definition 5, is a Boolean algebra. The elements of $\tilde{B}_n$ are functions f whose elements are ordered pairs of the form (P, a), where P is a row of D_n, and where a is an element of S. Let us agree to the following convention: we number the rows of the matrix D_n in the usual manner:

$$D_n = \begin{pmatrix} P_1 \\ P_2 \\ \cdot \\ \cdot \\ \cdot \\ P_{2^n} \end{pmatrix},$$ where each P_i is a 1 by n row matrix.

To represent a function f in $\tilde{B}_n$, we use the 2^n by 1 column matrix

$$\begin{pmatrix} f(P_1) \\ f(P_2) \\ \cdot \\ \cdot \\ \cdot \\ f(P_{2^n}) \end{pmatrix} .$$

<u>Definition</u> <u>6</u>. Let i be an integer, $1 \leq i \leq n$. The <u>independent</u> <u>variable</u> X_i is that function of $\tilde{B}_n$ which for each P in $\tilde{D}_n$, maps P onto the i^{th} coordinate of P. Hence, by the above convention, the independent variable X_i corresponds to the i^{th} column of the matrix D_n.

<u>Definition</u> <u>7</u>. A literal is a symbol X_i or X_i' $1 \leq i \leq n$.

Note that a literal always denotes a function in $\tilde{B}_n$, but not every representation of a function which admits a literal representation is a literal. For example, X_1 is a literal. The function X_1 is the function $X_1 + X_1$. That is $X_1 = X_1 + X_1$, but the symbol $X_1 + X_1$ is not a literal.

<u>Definition</u> <u>8</u>. A <u>clause</u> is a symbol of the form

$$\prod_{i=1}^{k} y_{s_i} \qquad \text{where}$$

1) $1 \leq s_i \leq n$.

2) y_{s_i} is a literal

3) $y_{s_i} = y_{s_j}$ if and only if $i = j$.

4) The product $y_{s_1} \cdot y_{s_2} \ldots y_{s_k}$ does not represent the zero element Φ in $\tilde{B}_n$.

Note that a clause denotes a function in $\tilde{B}_n$ but not every representation of a function which can be represented by a clause is a clause. For example, X_1X_2' is a clause. Further, the function X_1X_2' is the function

$$X_1X_2'X_3 + X_1X_2'X_3' \quad ,$$

but the symbol

$$X_1X_2'X_3 + X_1X_2'X_3'$$

is not a clause.

The <u>dimension</u> of the clause $\prod_{i=1}^{k} y_{s_i}$ is the integer k.

<u>Theorem 1</u>.

Let $\prod_{i=1}^{m} y_{s_i}$ be a clause of dimension m, and let f denote the function in $\tilde{B}_n$ which is represented by this clause. Then there are exactly 2^{n-m} elements P of $\tilde{D}_n$ such that $f(P) = 1$.

In order to prove the theorem, we need the following definition:

We define the product set S^n by induction.

$S^1 = \{ 0, 1 \}$, $S^{n+1} = \{ (q,a) \mid q \in S^n, a \in S \}$.

By induction, the cardinality of S^n is 2^n. Since no two rows of the matrix D_n are identical, it follows that $\tilde{D}_n = S^n$.

Proof of Theorem 1. An element P of $\tilde{D}_n$ is mapped onto 1 by f if and only if:

1) The $S_i{}^{th}$ coordinate of P is 1 if y_{s_i} is X_{s_i} or

2) The $S_i{}^{th}$ coordinate of P is 0 if y_{s_i} is X'_{s_i} .

With these m coordinates determined, there are $n - m$ coordinates of P which are not determined. The cardinality of the product set S^{n-m} is 2^{n-m}. But S^{n-m} is in $1 - 1$ correspondence with the set generated by holding fixed the m coordinates, and allowing the remaining to take on all possible values in S. Hence, there are exactly 2^{n-m} elements in $\tilde{D}_n$ which have the m determined coordinates. These, and only these, are mapped onto 1 by f.

<u>Corollary</u> <u>1-1</u>. Let $\prod_{i=1}^{n} y_i$ be a clause of dimension n, and let f be the function in $\tilde{B}_n$ represented by the clause. Then there is one and only one P in $\tilde{D}_n$ such that $f(P) = 1$. Conversely, let $P \in \tilde{D}_n$. Then there is one and only one clause of dimension n, such that the function f represented by this clause maps P onto 1.

Proof. The first part of the proposition follows immediately from Theorem 1. Let P be a given element of $\tilde{D}_n$. Construct a clause $\prod_{i=1}^{n} y_i$ of dimension n as follows:

if the i^{th} coordinate of P is 1, let y_i be X_i. If the i^{th} coordinate of P is 0, let y_i be X_i'. The function represented by the clause maps P onto 1. No other clause of dimension n can represent a function which maps P onto 1. For let $\prod_{i=1}^{n} z_i$ be any clause of dimension n. Write the clause $\prod_{i=1}^{n} y_i$ in the form $y_1 \cdot y_2 \cdot \ldots \cdot y_n$, and write $\prod_{i=1}^{n} z_i$ as $z_1 \cdot z_2 \cdot \ldots\ldots \cdot z_n$. Suppose the two clauses differ in the j^{th} factor. Then either

$$y_j = x_j \text{ and } z_j = x_j' \quad \text{or} \quad y_j = x_j' \text{ and } z_j = x_j.$$

In the first case, the j^{th} coordinate of P must be 1; and in the second case, the j^{th} coordinate of P must be 0. In either event, the function represented by the clause $\prod_{i=1}^{n} z_i$ does not map P onto 1. That is, P uniquely determines the factors of $\prod_{i=1}^{n} y_i$.

<u>Corollary 1-2</u>. Let f be an element of $\tilde{B}_n$, such that f is not the zero element. Then f uniquely determines a non-empty set of clauses, each clause of dimension n, such that the sum of these clauses represents f.

Proof: Consider the 2^n by 1 column matrix which represents f in $\tilde{B}_n$. If 1 appears in the i^{th} row of this matrix, there is a unique clause of dimension n which maps P_i, the i^{th} row of D_n onto 1. This follows immediately from Corollary 1-1. The sum of these clauses represents the function f. No clause of dimension n can be added to the collection of these clauses, for if one is added, then the sum of the clauses will represent a column matrix which for some row will have a 1 while 0 appears in that row of the matrix corresponding to f. No clause can be removed from the collection, since if one is removed, the sum of the remaining clauses will represent a column matrix which for some row has 0, while 1 appears in the corresponding row of the f-matrix.

<u>Definition 9</u>. Let $f \varepsilon \tilde{B}_n$. The rank of f, $\rho(f)$, is the number of occurrences of 1 in the corresponding

f - matrix.

Corollary 1-2 provides a canonical form for each non-zero element of $\tilde{B}_n$. That is, if f is an element of $\tilde{B}_n$ with rank j, $j \neq 0$, then f can be uniquely, up to the order of addition among the clauses, and up to the order of the factors of each clause, represented by the sum of j clauses, each clause of dimension n.

<u>Definition</u> <u>10</u>. Let $f \varepsilon \tilde{B}_n$, and assume that f can be represented as a sum of clauses, not necessarily all of the same dimension, where no clause appears more than once in the representation. This representation of f is called a <u>normal</u> <u>formula</u>.

For example, a normal formula of the function $X_1 + X_2$ is $X_1 + X_1' X_2$, as well as $X_1 + X_2$ itself.

Comment: Every non-zero element of $\tilde{B}_n$ has at least one representation which is a normal formula. This follows immediately from Corollary 1-2.

We are now in a position to define the simplification problem.

<u>Definition</u> <u>11</u>. Let f be a non-zero element of $\tilde{B}_n$. Let $\tilde{R}$ denote the class of normal formulas which represent f. Corresponding to each element ψ of $\tilde{R}$, there is an integer $o(\psi)$, where $o(\psi)$ is the sum of the number of occurrences of the operation $+$ in the representation ψ, and the number of occurrences of the operation $\cdot$ in the

representation ψ. The normal formula ψ_a is <u>a simplest normal representation</u> of the function f if and only if for every $\psi \in \tilde{R}$, $o(\psi_a) \leq o(\psi)$.

Example: Let $f = X_1 \cdot X_2 + X_1 \cdot X_3' + X_2' \cdot X_3'$, and let ψ_1 denote the representation $X_1X_2 + X_1X_3' + X_2'X_3'$. Another representation of f is $X_1 \cdot X_2 + X_2' \cdot X_3'$. Denote this representation by ψ_2. Then $o(\psi_1) = 5$ and $o(\psi_2) = 3$. We are not yet in a position to assert that ψ_2 is a simplest representation of f, but it is clearly simpler than ψ_1.

Now the simplification problem can be stated in this manner:

Given a function f in $\tilde{B}_n$, such that f is not the zero element, and such that f is not the identity element, determine the class of normal representations $\tilde{S}$, such that $\psi \in \tilde{S}$ if and only if ψ is a simplest normal representation of f.

Comment: The set $\tilde{S}$ is non-void. For let $\tilde{R}$ denote the class of normal formulas which represent f, and let $\tilde{C} = \{o(\psi) \mid \psi \in \tilde{R}\}$.

By Corollary 1-2, $\tilde{R}$ is non-empty. Hence $\tilde{C}$ is non-empty. By Definition 11, $\tilde{C}$ is bounded below by 0. Since $\tilde{C}$ is well ordered, it follows that $\tilde{C}$ contains a least number, m. That is, there is at least one element ψ_a in $\tilde{R}$, such that $o(\psi_a) = m$. Hence $\psi_a \in \tilde{S}$.

We will now develop an algorithm which generates all simplest normal representations of a function in $\tilde{B}_n$.

<u>Definition</u> <u>12</u>. Let f be a given function in $\tilde{B}_n$, and let ψ be a normal representation of f. A clause φ of ψ is <u>superfluous</u> if the formula $\tilde{\psi}$ obtained by the deletion of φ from ψ also represents the function f. A literal y_j of a clause ξ of ψ is <u>superfluous</u> with respect to f, if the formula obtained from ψ by the deletion of y_j from ξ also represents the function f. The normal formula ψ is <u>irredundant</u> if it has no superfluous clauses and none of its clauses has superfluous literals.

Example: Let $f = X_1X_2 + X_1X_3' + X_2'X_3'$. The representation $X_1X_2 + X_1X_3' + X_2'X_3'$ is not irredundant since the clause X_1X_3' is superfluous.

Example: Let $f = X_1X_2 + X_1X_2'X_3 + X_1'X_2'X_3'$. The representation $X_1X_2 + X_1X_2'X_3 + X_1'X_2'X_3'$ is not irredundant since the literal X_2' of the clause $X_1X_2'X_3$ is superfluous. It is a simple matter to verify these statements with the machinery which we have already developed.

Now given a function f in $\tilde{B}_n$ it might be reasonable to suppose that an irredundant formula which represents the function would be a simplest normal representation. This however, is not so.

Example: Let $f = X_1X_2' + X_1'X_2 + X_2X_3' + X_2'X_3$. The representation $X_1X_2' + X_1'X_2 + X_2X_3' + X_2'X_3$ is irredundant. There are however two simpler representations:

$$f = X_1X_2' + X_1'X_3 + X_2X_3'$$

$$f = X_1'X_2 + X_1X_3' + X_2'X_3$$

Quine (5,Vol59,p.521-531) has established a necessary condition that a normal formula must satisfy if it is to be a simplest normal representation of a given function. We now state and prove Quine's condition of necessity.

__Definition 13__. Let f and g be functions in $\tilde{B}_n$. Let F be that subset of $\tilde{D}_n$ consisting of all P such that $f(P) = 1$. Let G be that subset of $\tilde{D}_n$ consisting of all P such that $g(P) = 1$. The function f is said to __imply__ the function g if and only if $F \subset G$.

Lemma 3.

Let $f = \varphi\, y_i + \psi$ where ψ represents some function in $\tilde{B}_n$, φ is a clause, and y_i is a literal. Then $f = \varphi + \psi$ if and only if φ implies $y_i + \psi$.

Proof: Let $F = \{P \in \tilde{D}_n \mid [\varphi\, y_i + \psi]\,(P) = 1\}$

$$G = \{P \in \tilde{D}_n \mid [\varphi + \psi]\,(P) = 1\}$$

Let $g = \varphi + \psi$. Now $f = g$ if and only if $f(P) = g(P)$ for all P in $\tilde{D}_n$. That is, $f = g$ if and only if $F = G$.

We will show that $F = G$ if and only if φ implies $y_i + \psi$.

First, suppose φ implies $y_i + \psi$. Let $P \varepsilon G$. Then either $\varphi(P) = 1$, or (inclusive), $\psi(P) = 1$. If $\psi(P) = 1$, then $P \varepsilon F$. If $\psi(P) = 0$, then $\varphi(P) = 1$. But φ implies $y_i + \psi$ and $\psi(P) = 0$. Therefore $y_i\,(P) = 1$, $\varphi\, y_i\,(P) = 1$, and $P \varepsilon F$. Hence, if φ implies $y_i + \psi$, then $G \subset F$. Let $P \varepsilon F$. Then either $\varphi\, y_i\,(P) = 1$, in which case $\varphi\,(P) = 1$ and $P \varepsilon G$, or (inclusive), $\psi\,(P) = 1$ and $P \varepsilon G$. Hence $F \subset G$. Thus if φ implies $y_i + \psi$, then $F = G$. Next, suppose $F = G$ Let $\varphi\,(P) = 1$. Then $P \varepsilon G$ and hence $P \varepsilon F$. But since $P \varepsilon F$, either $\varphi\, y_i\,(P) = 1$, in which event $y_i\,(P) = 1$ or, (inclusive), $\psi\,(P) = 1$, and in any event, φ implies $\varphi\, y_i + \psi$. Therefore $\varphi\, y_i + \psi = \varphi + \psi$ if and only if φ implies $y_i + \psi$.

The above lemma provides a test which determines if a literal is superfluous in a clause of a given representation.

<u>Definition</u> <u>14</u>. Let φ and ξ be clauses. φ is said to <u>subsume</u> ξ if every literal which appears in ξ also appears in φ.

Example: The clause X_1X_2 subsumes the clause X_1. The clause X_1X_2 subsumes the clause X_2.

<u>Definition</u> <u>15</u>. Let f be a function in $\tilde{B}_n$. Let φ

be a clause. The clause φ is a <u>prime</u> <u>implicant</u> of f if

1) φ implies f and
2) φ subsumes no clause of smaller dimension than φ which also implies f.

<u>Theorem</u> <u>2</u>. (Quine's result)

Let f be a function in $\tilde{B}_n$, $f \neq \Phi$, where Φ is the zero element in $\tilde{B}_n$. Let ψ denote a simplest normal representation of f. Then each clause which appears in ψ is a prime implicant of f.

Proof: Suppose one clause which appears in ψ is not a prime implicant of f. Denote this clause by φ. Denote the representation ψ by $\varphi + \tilde{\psi}$, where $\tilde{\psi}$ is the formula obtained by the deletion of φ from ψ. Now every clause of ψ implies f. In particular, φ implies f. Since φ is not a prime implicant of f, φ subsumes a clause φ_1 of smaller dimension which also implies f. There is at least one literal y_i which appears in φ and which does not appear in φ_1. Write $f = \varphi_2 \, y_i + \tilde{\psi}$, where φ_2 is the clause obtained by the deletion of y_i from φ. Either φ_2 is φ_1 or φ_2 subsumes φ_1, and in either event, φ_2 implies φ_1. But φ_1 implies f. Now implication as defined in Definition 13 is clearly transitive. Hence φ_2 implies f. That is, φ_2 implies $\varphi_2 \, y_i + \tilde{\psi}$. By Lemma 3, $f = \varphi_2 + \tilde{\psi}$. But $o\,[\varphi_2 + \tilde{\psi}] < o\,[\varphi_2 \, y_i + \tilde{\psi}] = o(\psi)$, which contradicts the fact that ψ is a simplest representation of f. This contradiction establishes the theorem.

Let S denote that representation consisting of the sum of all prime implicants of f. The representation S is called Quine's canonical form of the function f. It is clear that S represents f. For surely S implies f. But we have shown that f has at least one simplest representation ψ. Now f implies ψ, and by Theorem 2, ψ implies S. Hence f implies S and S implies f. Therefore S represents f. Note that S is not necessarily a simplest representation of f. However, any simplest representation of f must consist of a sum of clauses, each of which appears in S.

Let K be any integer, where $0 < K \leq n$. We will determine the number of clauses of dimension K.

Lemma 4.

Let $X_1, X_2, \ldots, X_j$ be a sequence of literals. Then exactly 2^j clauses of dimension j can be generated by the literals $X_1, X_2, \ldots, X_j, X_1', X_2' \ldots X_j'$.

Proof: If $j = 1$, there are exactly 2 admissible clauses of dimension 1 which can be generated by the literals X_1 and X_1'. They are X_1 and X_1'. Assume the proposition holds for j. Then corresponding to each clause $(y_1 y_2 \ldots y_j)$ which can be generated by the literals $X_1, X_2, \ldots, X_j, X_1', X_2' \ldots X_j'$, there are exactly two clauses which can be generated from the literals $X_1, X_2, \ldots X_j, X_{j+1}, X_1', X_2', \ldots, X_j', X_{j+1}'$. They are: $(y_1\ y_2 \ldots y_j)X_{j+1}$ and $(y_1\ y_2 \ldots y_j)X_{j+1}'$. By the induction hypothesis there are exactly 2^j distinct clauses $(y_1\ y_2 \ldots y_j)$. Therefore, there are exactly $2 \cdot 2^j = 2^{j+1}$ distinct clauses which can be generated by the literals

$$X_1, X_2, \ldots X_j, X_{j+1}, X_1', X_2' \ldots X_j', X_{j+1}'.$$

Lemma 5.

Let K be any integer where $0 < K \leq n$. There are exactly $2^K \binom{n}{K}$ clauses of dimension K which represent functions in $\tilde{B}_n$.

Proof: Consider the set $\{X_1, X_2, \ldots X_n\}$. There are exactly $\binom{n}{K}$ distinct sets $\{X_{s_1}\ X_{s_2} \ldots X_{s_k}\}$, each such set consisting of exactly K of the letters $X_1, X_2 \ldots X_n$. By Lemma 4, exactly 2^K clauses of dimension K can be generated from each set

$$\{X_{s_1}, X_{s_2} \ldots X_{s_k}\} \cup \{X_{s_1}', X_{s_2}' \ldots X_{s_k}'\}.$$

Every clause of dimension K is generated in this process. Hence there are exactly $2^K \binom{n}{K}$ clauses of dimension K.

Lemma 6.

There are exactly $3^n - 1$ clauses which represent functions in $\tilde{B}_n$.

Proof: Let $\sum$ denote the total number of clauses which represent functions in $\tilde{B}_n$. By Lemma 5,

$$\sum = 2^n \binom{n}{n} + 2^{n-1} \binom{n}{n-1} + \dots 2^1 \binom{n}{1}$$

But

$$3^n = (2 + 1)^n = \sum + 1. \text{ Hence } 3^n - 1 = \sum .$$

Let $f \in \tilde{B}_n$, $f \neq \Phi$. Let Q denote the set of prime implicants of f. Q is clearly finite since the total number of clauses which represent functions in $\tilde{B}_n$ is finite. Once the set Q is determined, in theory it is a simple matter to determine the set of simplest representations of f. For suppose $Q = \{\varphi_1, \varphi_2, \dots \varphi_m\}$. We next consider the class P(Q) of all non-empty subsets of Q. P(Q) is just the power set of Q without the void set and therefore has cardinality $2^m - 1$. We sum over each element of P(Q) to obtain $2^m - 1$ representations, each of which implies f. We form the set $\tilde{S}'$, consisting of those representations of the $2^m - 1$ representations, which are implied by f. Again, $\tilde{S}'$ is non-empty since $\sum_{i=1}^{m} \varphi_i$ is an element of $\tilde{S}'$.

Next, we determine the set $\tilde{C}$: $\tilde{C} = \{o(\psi) \mid \psi \in \tilde{S}'\}$. Again, $\tilde{C}$ is non-empty, and has a least element r. The set $\tilde{S} = \{\psi \in \tilde{S}' \mid o(\psi) = r\}$ is the set of all simplest representations of f.

Given a representation ψ of a function f in $\tilde{B}_n$, in theory it is a simple matter to determine the prime implicants of f. To do so, we need only follow the steps outlined below.

Step 1. Generate the matrix D_n. For each clause y of dimension 1, compute and save the corresponding y-matrix. From the formula ψ, compute the f-matrix.

Step 2. For each clause y of dimension 1, compare the y-matrix to the f-matrix. y implies f if and only if for every row of the y-matrix in which 1 appears, 1 also appears in the corresponding row of the f-matrix. The clauses of dimension 1 which imply f are the prime implicants of dimension 1.

Step 3. Having determined the prime implicants of dimension K, consider all the clauses of dimension $K + 1$. For each clause δ of dimension $K + 1$, compute and save the corresponding δ-matrix. (It is possible that the δ-matrix of a clause of dimension $K + 1$ might be computed at Step 1 when the f-matrix is determined. In this event, there is no need to compute it again here.) Eliminate from consideration each clause of dimension $K + 1$ which subsumes any clause ξ of dimension less than $K + 1$, if ξ implies f. Again, ξ implies f if and only if for every row of the ξ-matrix in which 1 appears, 1 also appears in the corresponding row of the f-matrix. Those, and only those remaining clauses φ of dimension $K + 1$ <u>which imply f</u> are the prime implicants of dimension $K + 1$.

Step 4. If the clauses of dimension n have not been subjected to the process, repeat Step 3. If the clauses of dimension n have been subjected to the process, terminate the algorithm. The clauses so determined are the prime implicants of f.

Example 1: Let $n = 3$, and consider the function $f = X_1X_2' + X_1'X_2 + X_2X_3' + X_2'X_3$. We will determine the prime implicants of f. By Definition 3 we have:

$$D_1 = \begin{matrix} 0 \\ 1 \end{matrix}, \qquad D_2 = \begin{matrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix} \quad \text{and} \quad D_3 = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{matrix}$$

Here we omit the parentheses of the matrices for notational convenience.

By Definition 6 we have:

$$X_1 = \begin{matrix}0\\0\\0\\0\\1\\1\\1\\1\end{matrix}, \qquad X_2 = \begin{matrix}0\\0\\1\\1\\0\\0\\1\\1\end{matrix}, \quad \text{and} \quad X_3 = \begin{matrix}0\\1\\0\\1\\0\\1\\0\\1\end{matrix}$$

By Definition 5 we have:

$$X_1' = \begin{matrix}1\\1\\1\\1\\0\\0\\0\\0\end{matrix}, \qquad X_2' = \begin{matrix}1\\1\\0\\0\\1\\1\\0\\0\end{matrix}, \quad \text{and} \quad X_3' = \begin{matrix}1\\0\\1\\0\\1\\0\\1\\0\end{matrix}$$

Again, by Definition 5,

$$X_1X_2' = \begin{matrix}0\\0\\0\\0\\1\\1\\1\\1\end{matrix} \cdot \begin{matrix}1\\1\\0\\0\\1\\1\\0\\0\end{matrix} = \begin{matrix}0\\0\\0\\0\\1\\1\\0\\0\end{matrix}, \qquad X_1'X_2 = \begin{matrix}1\\1\\1\\1\\0\\0\\0\\0\end{matrix} \cdot \begin{matrix}0\\0\\1\\1\\0\\0\\1\\1\end{matrix} = \begin{matrix}0\\0\\1\\1\\0\\0\\0\\0\end{matrix}$$

$$X_2X_3' = \begin{matrix}0\\0\\1\\1\\0\\0\\1\\1\end{matrix} \cdot \begin{matrix}1\\0\\1\\0\\1\\0\\1\\0\end{matrix} = \begin{matrix}0\\0\\1\\0\\0\\0\\1\\0\end{matrix}, \quad \text{and} \quad X_2'X_3 = \begin{matrix}1\\1\\0\\0\\1\\1\\0\\0\end{matrix} \cdot \begin{matrix}0\\1\\0\\1\\0\\1\\0\\1\end{matrix} = \begin{matrix}0\\1\\0\\0\\0\\1\\0\\0\end{matrix}$$

Note that these multiplications resemble the "logical and" instruction which is found on most digital computers.

Again, by Definition 5,

$$
\begin{array}{l}
X_1X_2' + X_1'X_2 + X_2X_3' + X_2'X_3 = \\
\end{array}
\begin{array}{ccccccccccc}
0 & + & 0 & + & 0 & + & 0 & = & 0 & = & f \\
0 & & 0 & & 0 & & 1 & & 1 & & \\
0 & & 1 & & 1 & & 0 & & 1 & & \\
0 & & 1 & & 0 & & 0 & & 1 & & \\
1 & & 0 & & 0 & & 0 & & 1 & & \\
1 & & 0 & & 0 & & 1 & & 1 & & \\
0 & & 0 & & 1 & & 0 & & 1 & & \\
0 & & 0 & & 0 & & 0 & & 0 & &
\end{array}
$$

Note that these additions resemble the "logical or" instruction which is found on most digital computers.

We have now completed Step 1 of the algorithm. According to Step 2, we must now determine all clauses of dimension 1 which imply the function f. For convenience of comparison, we write the following table:

TABLE 1.

X_1	X_2	X_3	X_1'	X_2'	X_3'	f
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	1	0	0	1
1	0	0	0	1	1	1
1	0	1	0	1	0	1
1	1	0	0	0	1	1
1	1	1	0	0	0	0

We find that no clause of dimension 1 implies f; X_1, X_2, and X_3 are eliminated by the occurrence of 0 in row 8 of the f-matrix; X_1', X_2', and X_3' are eliminated by the occurrence of 0 in row 1 of the f-matrix. Hence, no prime implicant of f is of dimension 1.

According to Step 3, we must first generate all clauses of dimension 2. By Lemma 5, there are exactly $2^2\binom{3}{2} = 12$ such clauses. Note that Lemmas 4 and 5 give constructive proofs; they describe how these clauses can be generated.

According to Lemma 5, we determine all subsets of cardinality two of the set $\{X_1, X_2, X_3\}$. These are $\{X_1, X_2\}$, $\{X_1, X_3\}$ and $\{X_2, X_3\}$.

According to Lemma 4, from each of these three sets we generate $2^2 = 4$ clauses of dimension 2. For example, from the set $\{X_1, X_2\}$, we generate the clauses:

$$\begin{array}{l} X_1 X_2 \\ X_1 X_2' \\ X_1' X_2 \\ X_1' X_2' \quad . \end{array}$$

Similarly, from $\{X_1, X_3\}$ we generate

$$\begin{array}{l} X_1 X_3 \\ X_1 X_3' \\ X_1' X_3 \\ X_1' X_3' \quad , \end{array}$$

and from $\{X_2, X_3\}$ we generate

$$\begin{array}{l} X_2 X_3 \\ X_2 X_3' \\ X_2' X_3 \\ X_2' X_3' \quad . \end{array}$$

According to Lemma 5, these are all the clauses of dimension 2. Next, we compute for each clause of dimension 2, its corresponding matrix. Of course we have done this for the clauses X_1X_2', $X_1'X_2$, X_2X_3' , and $X_2'X_3$. We compute the remaining matrices and for convenience of comparison, we write TABLE 2.

Next, we must eliminate from consideration all clauses of dimension 2 which subsume any clause ξ of dimension 1, if ξ implies f. But, in this example, there is no clause of dimension 1 which implies f. Therefore, we must consider all clauses of dimension 2 which imply f.

TABLE 2

X_1X_2	X_1X_2'	$X_1'X_2$	$X_1'X_2'$	f
0	0	0	1	0
0	0	0	1	1
0	0	1	0	1
0	0	1	0	1
0	1	0	0	1
0	1	0	0	1
1	0	0	0	1
1	0	0	0	0

X_1X_3	X_1X_3'	$X_1'X_3$	$X_1'X_3'$	f
0	0	0	1	0
0	0	1	0	1
0	0	0	1	1
0	0	1	0	1
0	1	0	0	1
1	0	0	0	1
0	1	0	0	1
1	0	0	0	0

X_2X_3	X_2X_3'	$X_2'X_3$	$X_2'X_3'$	f
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
1	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	1	0	0	1
1	0	0	0	0

From TABLE 2 we immediately determine all clauses of dimension 2 which imply f. They are: X_1X_2', $X_1'X_2$, X_1X_3', $X_1'X_3$, X_2X_3', and $X_2'X_3$. These and only these are the prime implicants of dimension 2. Note that: We found that no clause of dimension 1 implies f. From this we could have immediately concluded, without further use of the algorithm, that the given clauses were prime implicants of f. For the given clauses, X_1X_2', $X_1'X_2$, X_2X_3', and $X_2'X_3$, are all of dimension 2, and clearly no given clause can subsume a clause of dimension 1 which implies f, the latter set being empty. But in addition to the given clauses, the algorithm has generated other prime implicants of dimension 2, namely, X_1X_3' and $X_1'X_3$.

Conclusion: <u>Once enough prime implicants have been determined to recover the function, it does not necessarily follow that all of the prime implicants have been determined</u>. In what follows, when we refine the algorithm, we will prove a theorem which is concerned with this fact.

At present however, we can not terminate the algorithm. Hence we proceed with Step 4. According to Step 4, we we return to Step 3. At this point, the index K of step 3 has the value 2. Hence we must consider all clauses of dimension 3. According to Lemma 5, we consider the set

$\{X_1, X_2, X_3\}$. According to Lemma 4, we consider, say, the clauses

$$X_1 \; X_2$$
$$X_1 \; X_2'$$
$$X_1' \; X_2$$
$$X_1' \; X_2' \quad .$$

From these we generate eight clauses:

$$X_1 \; X_2 \; X_3$$
$$X_1 \; X_2' \; X_3$$
$$X_1' \; X_2 \; X_3$$
$$X_1' \; X_2' \; X_3$$
$$X_1 \; X_2 \; X_3'$$
$$X_1 \; X_2' \; X_3'$$
$$X_1' \; X_2 \; X_3'$$
$$X_1' \; X_2' \; X_3'$$

According to Lemma 5, these are all the clauses of dimension 3 in $\tilde{B}_3$. According to Step 3, we compute the following matrices:

TABLE 3

$X_1'X_2'X_3'$	$X_1'X_2'X_3$	$X_1'X_2X_3'$	$X_1'X_2X_3$	$X_1X_2'X_3'$	$X_1X_2'X_3$	$X_1X_2X_3'$	$X_1X_2X_3$
1	0	0	0	0	0	00	0
0	1	0	00	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1

We now eliminate each clause of dimension 3 which subsumes any clause ξ of dimension 1 if ξ implies f. But, as we have seen, no clause of dimension 1 implies f, and therefore we are left with all eight clauses of dimension 3. Next, we eliminate each clause of dimension 3 which subsumes any clause ξ of dimension 2 if ξ implies f. Now we have already determined the clauses of dimension 2 which imply f. They are: X_1X_2', $X_1'X_2$, X_1X_3', $X_1'X_3$, X_2X_3', and $X_2'X_3$. Of course, in this particular example, the clauses of dimension 2 which imply f are the prime implicants of dimension 2. However, in general, this will not be true. We write the following table:

TABLE 4

Clauses of Dimension Three	Clauses of Dimension less Than Three Which Imply f
$X_1\ X_2\ X_3$	$X_1\ X_2'$
$X_1\ X_2'\ X_3$	$X_1'\ X_2$
$X_1'\ X_2\ X_3$	$X_1\ X_3'$
$X_1'\ X_2'\ X_3$	$X_1'\ X_3$
$X_1\ X_2\ X_3'$	$X_2\ X_3'$
$X_1\ X_2'\ X_3'$	$X_2\ X_3$
$X_1'\ X_2\ X_3'$	
$X_1'\ X_2'\ X_3'$	

From TABLE 4 we see that the following clauses of dimension 3 subsume clauses of dimension less than three which imply f:

$$X_1\ X_2'\ X_3$$
$$X_1'\ X_2\ X_3$$
$$X_1'\ X_2'\ X_3$$
$$X_1\ X_2\ X_3'$$
$$X_1\ X_2'\ X_3'$$
$$X_1'\ X_2\ X_3'$$

Hence we eliminate these from further consideration. We are left with two clauses: $X_1 X_2 X_3$, and $X_1' X_2' X_3'$. Again we write the following table for convenience:

$X_1 X_2 X_3$	$X_1' X_2' X_3'$	f
0	1	0
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1
1	0	0

We immediately see that the two remaining clauses do not imply f, and according to Step 4 we terminate the process: Hence the prime implicants of f are: X_1X_2', $X_1'X_2$, $X_1'X_3'$, $X_1'X_3$, X_2X_3' and $X_2'X_3$.

The algorithm as developed at this point is complete. That is, we have developed sufficient machinery to execute the four basic steps and generate the prime implicants of a given function. However, as Example 1 illustrates, the algorithm is not particularly efficient. Therefore we will next prove some theorems which will increase its efficiency.

The following lemma makes it possible to by-pass the recursive definition in the generation of D_n.

Lemma 7.

The K^{th} column of the matrix D_n can be partitioned into 2^K 2^{n-K} by 1 sub-matrices Q_i, where i is the row index, such that if i is odd, each element of Q_i

is 0, and if i is even, each element of Q_i is 1.

Proof:

$D_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, and the proposition holds for $n = 1$.

Assume the proposition holds for D_n and consider D_{n+1}.

By Definition 3,

$$D_{n+1} = \left(\begin{array}{c|c} \theta & D_n \\ \hline I & D_n \end{array}\right), \quad \text{where}$$

θ is a 2^n by 1 column matrix and I is a 2^n by 1 column matrix. Therefore the proposition is true for column 1 of D_{n+1}.

If $1 < K \leq n + 1$, then the K^{th} column of D_{n+1} is the $K-1$ column of $\begin{pmatrix} D_n \\ D_n \end{pmatrix}$. By the induction hypothesis, the $K-1$ column of D_n can be written as: $\begin{matrix} Q_1 \\ Q_2 \\ \cdot \\ \cdot \\ \cdot \\ Q_{2^{k-1}} \end{matrix}$,

where each Q_i is a 2^{n-K+1} by 1 matrix such that if i is odd, each element of Q_i is 0, and if i is even, each element of Q_i is 1. Therefore the $K-1$ column of $\left(\frac{D_n}{D_n}\right)$ can be written as

Q_1
Q_2
.
.
.
$Q_{2^{K-1}}$
Q_1
Q_2
.
.
.
$Q_{2^{K-1}}$

This is the K^{th} column of D_{n+1}. There are $2 \cdot 2^{K-1} = 2^K$ partitions of this matrix, and each sub-matrix has $2^{(n+1)-K}$ elements. Further, each element of the sub-matrix with row index i is 0 if i is odd, and 1 if i is even. This establishes the proposition.

Let ψ denote a normal representation of a function f in $\tilde{B}_n$. The representation ψ is called a <u>developed normal formula</u> if for each literal y_i which occurs in one clause of the representation, the literal y_i or the literal y_i' occurs in every clause of the representation. That is, for each letter which occurs in one clause of the representation, we require that the letter occur in every clause of the representation. For example, $X_1X_2 + X_1'X_2'$ is a developed normal formula. It follows immediately from Corollary 1-2, that every function $f \in \tilde{B}_n$, $f \neq \bar{\Phi}$, has at least one developed normal formula.

Now one advantage of our algorithm is that we are not required to start with a developed normal formula. Indeed, the method of the last example is completely general, in the sense that ψ need not even be a normal formula, for given any representation of the function f, we can determine the corresponding f-matrix by the methods of the last example. But for this generality we pay the price of computing the matrix of each literal which appears in ψ, in order to combine the matrices according to ψ, and thereby determine the matrix of the function which ψ represents. Now suppose that ψ is, in fact, a normal formula. We do not assume ψ to be developed, we just assume that ψ is normal. Is there a more efficient way to determine the matrix of the function which ψ represents?

Suppose $(y_{s_1} y_{s_2} \cdots y_{s_j})$ is a clause of ψ. By Theorem 1, there are exactly 2^{n-j} rows of D_n which are mapped onto 1 by the function in $\tilde{B}_n$ represented by the clause $y_{s_1} y_{s_2} \cdots y_{s_j}$. Hence there are exactly 2^{n-j} rows of the matrix of that function in which 1 appears, and the row number of each such row can be determined by the matrix D_n. For example, consider the clause X_1X_3' which represents a function f in $\tilde{B}_3$.

	X_1	X_2	X_3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

The two rows of D_3 which are mapped onto 1 by X_1X_3' are row 5 and row 7. Hence 1 appears only in rows 5 and 7 of the f-matrix and we have

$$f = \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{matrix}$$

Therefore, if we are given ψ as a normal formula, we can determine the matrix of the corresponding function in $\tilde{B}_n$ without computing the individual matrices of the clauses of ψ. For example, consider, the representation $X_1X_2 + X_2X_3$. We compute the matrix of the function of $\tilde{B}_3$ represented by $X_1X_2 + X_2X_3$ as follows:

	X_1	X_2	X_3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

The two rows of D_3 which are mapped onto 1 by X_1X_2 are rows 7 and 8. The two rows of D_3 which are mapped onto 1 by X_2X_3 are rows 4 and 8. Hence the only rows of the matrix of the function represented by $X_1X_2 + X_2X_3$, in which 1 appears, are rows 4, 7, and 8. Therefore this matrix is:

1	0
2	0
3	0
4	1
5	0
6	0
7	1
8	1

Now assuming that ψ is given as any normal representation of the function f, with the aid of Lemma 6 and the above method, we write Step 1 of the algorithm in the following manner:

Step 1. Determine D_n. From D_n and ψ, compute the f-matrix. Compute $\rho(f)$. (By Definition 9, the rank of f, $\rho(f)$, is the number of occurrences of 1 in the f-matrix.)

Suppose f is of rank m, where $m < 2^{n-K}$. Since each clause of $\tilde{B}_n$ of dimension K has rank 2^{n-K}, it follows that no clause of dimension less than or equal to K can imply f. Hence no clause of dimension less than or equal to K can be a prime implicant of f. Hence, in determining the prime implicants of f, we can ignore all clauses of dimension less than or equal to K. Hence Step 2 of the algorithm can be written:

Step 2. Determine the smallest integer j such that $0 < j \leq n$, and $2^{n-j} \leq \rho(f)$. Each clause φ of dimension j which implies f is a prime implicant of f. The clauses so determined are all the prime implicants of f of dimension j. No clause of dimension less than j is a prime implicant of f.

There is one case in which the conditions $0 < j \leq n$ and $2^{n-j} \leq \rho(f)$ can not both be satisfied. If f is the identity element in $\tilde{B}_n$, then $\rho(f) = 2^n$. Thus if $2^{n-j} \leq \rho(f) = 2^n$, j can not satisfy $0 < j \leq n$. But in this case, any representation of the form $X_i + X_i'$, $(i = 1, 2, \ldots n)$ will represent the identity element, and the case is trivial, since each such representation is a simplest representation of the identity element.

Now given the f-matrix, where $\rho(f) = m$, and given j such that $0 < j \leq n$ and $2^{n-j} \leq \rho(f)$, we wish to determine all clauses of dimension j which imply f, and yet avoid computing the individual matrices of the clauses. Further, we wish to minimize the number of candidates to be considered.

By Corollary 1-1 each row of D_n uniquely determines a clause of dimension n such that 1 appears in the corresponding matrix of the function which that clause represents. For example, row 3 of D_4 is (0, 0, 1, 0) and this row determines the clause $X_1'X_2'X_3X_4'$. Hence the matrix corresponding to the clause $X_1'X_2'X_3X_4'$ must have a 1 in its fourth row.

Now given the r^{th} row of D_n, let φ be the clause of dimension n determined by that row, such that 1 appears in the r^{th} row of the φ-matrix. Now φ subsumes exactly $\binom{n}{j}$ clauses of dimension j, and for each of these clauses, 1 must appear in the r^{th} row of its corresponding matrix. Hence, it follows that if 1 does not appear in the r^{th} row of the f-matrix, none of these clauses of dimension j imply f.

However, if 1 does appear in the r^{th} row of the f-matrix, each of these clauses is a <u>candidate</u>. Assuming then that 1 appears in the r^{th} row of the f-matrix, let ξ denote one of the $\binom{n}{j}$ clauses of dimension j which are subsumed by φ. Then 1 appears in the r^{th} row of the ξ-matrix. But, unless $j = n$, we can not assume from this that ξ implies f, for there are 2^{n-j} rows of D_n which are mapped by ξ onto 1, and hence there are 2^{n-j} corresponding rows of the ξ-matrix in which 1 appears. Denote their row numbers by $S_1, S_2, .. r, ... S_{2^{n-j}}$. Then ξ implies f if and only if 1 appears in rows $S_1, S_2, ... r, ... S_{2^{n-j}}$ of the f-matrix.

Hence we have the following process for determining the clauses of dimension j which imply f:

Case I. $\rho(f) \geq 2^{n-1}$. In this event, we must start the process at $j = 1$. To determine the clauses of dimension 1 which imply f we note that:

If in each row of column i of D_n in which 1 appears, 1 also appears in the corresponding row of the f-matrix, X_i implies f; if not, X_i does not imply f. Assuming that f is not the identity element, if X_i implies f then X_i' does not imply f. If in each row of column i of D_n in which 1 appears, 0 appears in the corresponding row of the f-matrix, then X_i' implies f; if not, X_i' does not imply f.

Case II. $\rho(f) = m$ and $1 < j$.

1. Let $S_1, S_2 \ldots S_m$ be an increasing sequence such that 1 appears in rows $S_1, S_2, \ldots$ and S_m of the f-matrix.

2. Ignore all rows of D_n except rows $S_1, S_2, \ldots S_m$. Let the index i have value 1.

3. For the row S_i of D_n let φ_1 denote the uniquely determined clause of dimension n which maps row S_i of D_n onto 1. For each clause $\xi_{i,t}$ ($t = 1, 2, \ldots \binom{n}{j}$), subsumed by φ_i, determine the 2^{n-j} rows of D_n which $\xi_{i,t}$ maps onto 1. Denote

their row numbers by $\mu_{t,1}, \mu_{t,2}, \ldots\ldots \mu_{t,2^{n-j}}$. If 1 appears in rows $\mu_{t,1}, \mu_{t,2}, \ldots \mu_{t,2^{n-j}}$ of the f-matrix, then $\xi_{i,t}$ implies f; if not, eliminate $\xi_{i,t}$ from consideration.
With i fixed, the above process is to be performed for $t = 1, 2, \ldots\ldots \binom{n}{j}$.

4. If 3) has not been performed for $i = m$, increase i by 1 and repeat 3). If 3) has been performed for $i = m$, terminate the process. The clauses so determined are all the clauses of dimension j which imply f.

It should be noted that for distinct values of i, say i_1 and i_2, it is possible that $\xi_{i_1,t}$ and $\xi_{i_2,t}$ can denote the same clause. Thus, if $i_1 < i_2$ and $\xi_{i_1,t}$ and $\xi_{i_2,t}$ denote the same clause, and $\xi_{i_1,t}$ has been found to either imply f or not imply f, it of course is not necessary to again test $\xi_{i_2 t}$.

Example: Consider the function f in $\widetilde{B}_3$ which is represented by $X_1'X_2X_3 + X_1X_2X_3' + X_1X_2X_3$. Suppose that we wish to determine all clauses of dimension 2 which imply f.

	X_1	X_2	X_3	f
1	0	0	0	
2	0	0	1	
3	0	1	0	
4	0	1	1	1
5	1	0	0	
6	1	0	1	
7	1	1	0	1
8	1	1	1	1

We immediately determine the f-matrix. Then from row 4, there are three candidates: $X_1'X_2$, $X_1'X_3$, and X_2X_3. Of these, $X_1'X_2$ is eliminated by row 3, and $X_1'X_3$ is eliminated by row 2. The remaining clause X_2X_3 implies f. From row 7 we obtain the candidates X_1X_2, X_1X_3' and X_2X_3'. X_1X_2 implies f; X_1X_3' and X_2X_3' are eliminated. From row 8, X_1X_2, X_1X_3 and X_2X_3 are candidates. We have determined already that X_1X_2 and X_2X_3 imply f. The remaining clause X_1X_3 is eliminated by row 6. Hence the clauses of dimension 2 which imply f are: X_1X_2 and X_2X_3.

According to Definition 15, a clause φ is a prime implicant of the function f if φ implies f, and does not subsume any clause of smaller dimension which also implies f. Thus in Step 3 of the algorithm, when we determine the prime implicants of dimension $K + 1$, for each clause φ of dimension $K + 1$ which implies f, we must be assured that φ subsumes no clause of smaller dimension which implies f. Hence, in Step 3, we test φ against every clause ξ of dimension less than $K + 1$, where ξ implies f. We then eliminate φ from consideration if φ subsumes any such clause ξ. That this procedure can be simplified is suggested by the following results:

Lemma 8.

The relation subsume is of a transitive nature: if φ subsumes ξ, and ξ subsumes δ, then φ subsumes δ. The proof is by direct application of Definition 14.

Lemma 9.

Let f be a given function in $\tilde{B}_n$, $f \neq \underline{\Phi}$. Let φ be any clause of dimension j such that φ implies f. Then φ is a prime implicant of f if φ subsumes no <u>prime implicant</u> of f of dimension less than j.

Proof: If φ is of dimension 1, and if φ implies f, then φ is a prime implicant of f. Assume the proposition holds for dimensions $1, 2, \ldots K$. Let φ have dimension $K + 1$, such that φ implies f and φ subsumes no prime implicant of f of dimension less than $K + 1$. Suppose φ is not a prime implicant of f. Then φ subsumes some clause ξ of dimension less than $K + 1$, where ξ implies f. But φ subsumes no prime implicant of f of dimension less than $K + 1$. Now either ξ is a prime implicant of f, in which case we have a contradiction, or ξ is not a prime implicant of f. But if ξ is not a prime implicant of f, by the induction hypotheses, it follows that ξ must subsume a prime implicant δ of f, where dimension δ < dimension ξ. Then φ subsumes ξ, and ξ subsumes δ. By Lemma 8, φ subsumes δ, and dimension δ < dimension ξ < dimension φ, another contradiction.

Combining Definition 15 and Lemma 9, we have the following theorem:

<u>Theorem</u> <u>3</u>:

Let $f \in \tilde{B}_n$, $f \neq \Phi$. Let φ be any clause of $\tilde{B}_n$. A necessary and sufficient condition that φ is a prime implicant of f is that:

1. φ implies f and
2. φ subsumes no prime implicant of f with dimension less than that of φ.

Proof: Sufficiency is by Lemma 9, necessity is an immediate consequence of Definition 15.

The above results suggest that Step 3 should be modified in the following manner: Assuming that all the prime implicants of a function f have been determined up to and including those of dimension K, we wish to determine the prime implicants of dimension $K + 1$. Now for each candidate φ, we immediately eliminate φ from consideration if φ subsumes one of the prime implicants of dimension less than $K + 1$, this step being justified by the second condition of Theorem 3. If φ does not subsume any of the prime implicants of dimension less than $K + 1$, we next determine if φ implies f. If φ then implies f, φ is a prime implicant of f. The order of the above two steps can be reversed. That is, given the candidate φ, we can first determine if φ implies f. If φ does not imply f, of course φ is not a prime implicant. If φ implies f, we determine if φ subsumes any of the prime implicants of dimension less than $K + 1$. If then φ does not subsume one of the prime implicants of dimension less than $K + 1$, φ is a prime implicant. The significance of either method, of course, is that in the process of determining the clauses of dimension $K + 1$, as a result of the previous steps in the algorithm which determine all the prime implicants of dimension less than $K + 1$, we are not required to test each candidate φ against every <u>clause</u> ξ of dimension less than $K + 1$ which implies f to insure that φ does not subsume ξ; we are only required to test φ against each <u>prime implicant</u> of dimension less than $K + 1$.

Although it is not our specific purpose to develop a computer-oriented algorithm, it is obvious that for even a moderate number of independent variables, it would be impractical to use the methods so far described without a computer. Now the algorithm so far developed is at least partially computer-oriented. In particular, the operations $\cdot$, $+$, and $'$ of our model are easily applied to a computer. However the basic definition of subsumes has visual connotations. For example, the clause $X_1X_2X_3$ subsumes the clause X_1X_3, and we determine this by actually observing that the symbols X_1 and X_3 appear in the clause $X_1X_2X_3$. Now it is certainly true that computer systems in the present state of the technology can distinguish and work with symbols. But in general, the basic computer can not. Hence it is necessary to convert a symbol, by some suitable code, to a form with which the computer can work. But this form is usually a sequence of the digits 1 and 0. Thus, it would require extra programming for the computer to determine that $X_1X_2X_3$ subsumes X_1X_3, but this extra programming would not be necessary for the computer to determine that $X_1X_2X_3$ implies X_1X_3, if the matrices

$X_1X_2X_3$	X_1X_2
0	0
0	0
0	0
0	0
0	0
0	0
0	1
1	1

were stored in the memory of the computer. The following theorem, Theorem 4, might therefore have practical applications, in addition to the theoretical applications for which we will find it useful.

Lemma 10.

Let φ and ξ be clauses of $\tilde{B}_n$. Then a necessary condition that φ subsumes ξ is that φ implies ξ.

Proof: Let φ be a clause of dimension j, and let ξ be a clause of dimension K, and assume that φ subsumes ξ. (We can assume that $j > K$, for if $j = K$, then $\varphi = \xi$ and there is nothing to prove.) Let φ be the clause $y_{s_1} y_{s_2} \ldots y_{s_k}$. Let $P \varepsilon \tilde{D}_n$ such that $\varphi(P) = 1$. Then the following condition is satisfied:

If y_{s_i} is X_{s_i} then the S_i^{th} coordinate of P is 1. If y_{s_i} is X'_{s_i} then the S_i^{th} coordinate of P is 0. But this condition, since φ subsumes ξ, is sufficient to insure that $\xi(P) = 1$. By Definition 13, φ implies ξ.

The condition of Lemma 10 is also sufficient. That is, if φ and ξ are clauses, and φ implies ξ, then φ subsumes ξ. We state this in Theorem 4 below after some preliminary lemmas.

Definition 16. Let $P \varepsilon \tilde{D}_n$ and $q \varepsilon \tilde{D}_n$. P and q are equivalent by the deletion of X_K if P and q differ only in the K^{th} coordinate. We write $(P \sim q)$ by X_K to denote this relation.

<u>Definition</u> <u>17</u>. Let $f \varepsilon \tilde{B}_n$. The function f is <u>independent</u> of the variable $\underline{X_K}$ provided that for every P in $\tilde{D}_n$, $f(P) = f(q)$ if $(P \sim q)$ by X_K.

Lemma 11.

Let φ and ξ be clauses in $\tilde{B}_n$. Then a sufficient condition that φ subsumes ξ is that φ implies ξ.

Proof: Let φ have dimension K and let ξ have dimension j, and assume that φ implies ξ. Clearly $K \geq j$, for if $K < j$, since there are 2^{n-K} P in $\tilde{D}_n$ such that $\varphi(P) = 1$, and 2^{n-j} P in $\tilde{D}_n$ such that $\xi(P) = 1$, it would follow that for some P for which $\varphi(P) = 1$, $\xi(P) = 0$. Hence $K \geq j$. Now if $K = j$, $\varphi = \xi$ and φ subsumes ξ. Suppose $K > j$. Now suppose that φ does not subsume ξ. Then either a literal y_m appears in ξ and the literal y_m' appears in φ, or a literal y_m appears in ξ and neither y_m or y_m' appear in φ. The first case is clearly impossible since φ implies ξ. In the second case, φ is indpendent of X_m. Let $\varphi(P) = 1$. Then $\xi(P) = 1$, and hence the m^{th} coordinate of P is 1 if y_m is X_m, or the m^{th} coordinate of P is 0 if y_m is X_m'. Consider the element q of $\tilde{D}_n$ such that $(P \sim q)$ by X_m. Then, since φ is independent of X_m, $\varphi(q) = 1$. But $\xi(q) = 0$. Hence φ does not imply q. This contradiction establishes the proposition.

Combining the Lemmas 10 and 11, we have the following Theorem:

Theorem 4.

Let φ and ξ be clauses of $\tilde{B}_n$. Then a necessary and sufficient condition that φ subsumes ξ is that φ implies ξ.

Remark: It must be emphasized that φ and ξ are clauses in the statement of Theorem 4.

Lemma 12.

Let $f \varepsilon \tilde{B}_n$, where f is not the identity element, and assume that f is independent of X_K, $1 \leq K \leq n$. Let φ be a clause which implies f such that the literal y_K, $(y_K = X_K$ or $y_K = X_K')$ appears in φ. Then φ is not a prime implicant f.

Proof: φ can not be of dimension 1. For suppose φ is of dimension 1. Then φ is the literal y_K. Now f is not the identity element. Therefore, there exists $P \varepsilon \tilde{D}_n$ such that $f(P) = 0$. Now φ implies f and φ is y_K. Hence $y_K(P) = 0$. Let q be that element of $\tilde{D}_n$ such that $(P \sim q)$ by X_K. Then $\varphi(q) = y_K(q) = 1$. But f is independent of X_K, and therefore $f(q) = 0$. This contradicts the fact that φ implies f. Thus φ is not of dimension 1. Therefore let ξ be the clause obtained by the deletion of the literal y_K from φ. Then ξ is independent of X_K. Let $P \varepsilon \tilde{D}_n$ such that

$\xi(P) = 1$. If $y_K(P) = 1$, then $\varphi(P) = 1$, and, since φ implies f, $f(P) = 1$. If $y_K(P) = 0$, let q be that element of $\tilde{D}_n$ such that $(P \sim q)$ by X_K. Then $\xi(q) = 1$ since ξ in independent of X_K. But $y_K(q) = 1$. Hence $\varphi(q) = 1$ and $f(q) = 1$. But f is independent of X_K, and $(P \sim q)$ by X_K. Hence $f(P) = 1$. Therefore if $\xi(P) = 1$, then $f(P) = 1$. That is, ξ implies f. Since φ subsumes ξ, φ is not a prime implicant of f.

At this point, we again consider, Example 1. In Example 1 we found that the prime implicants X_1X_2', $X_1'X_2$, X_2X_3' and $X_2'X_3$ were enough to recover the function f, where one representation of f is:

$$f = X_1'X_2'X_3 + X_1'X_2X_3' + X_1'X_2X_3 + X_1X_2'X_3' + X_1X_2'X_3 + X_1X_2X_3'.$$

We found in addition to X_1X_2', $X_1'X_2$, and X_2X_3', two other prime implicants of dimension 2: X_1X_3' and $X_1'X_3$. However we found that no clauses of dimension 3 were prime implicants, and indeed, we were required to complete what turned out to be extraneous work in order to verify the latter remark. That this work was unnecessary is shown by the following theorem:

<u>Theorem 5</u>.

Let f be an element of $\tilde{B}_n$, $n > 2$, such that f is not the zero element and f is not the identity element. Let $\tilde{P}$ denote the set of <u>all</u> prime implicants of f of dimension 2 or less. Then if $\sum_{\varphi \in \tilde{P}} \varphi = f$, no clause of dimension greater than 2 is a prime implicant of f.

Proof: Let $\widetilde{P} = \{\varphi_1, \varphi_2, \dots \varphi_m\}$ be the set of all prime implicants of f of dimension 2 or less, and assume that $f = \varphi_1 + \varphi_2 + \dots + \varphi_m$. Let M be the set of all integers j such that:

1) $n \geq j > 2$ and

2) Every clause of $\widetilde{B}_n$ with dimension j which implies f subsumes at least one clause in $\widetilde{P}$.

Let φ be a clause of dimension n such that φ implies f. By Corollary 1 - 1 there is exactly one element P in $\widetilde{B}_n$ such that $\varphi(P) = 1$. But φ implies $f = \varphi_1 + \varphi_2 + \dots + \varphi_m$, and hence there is at least one φ_i, $1 \leq i \leq m$, such that $\varphi_i(P) = 1$. Therefore φ implies φ_i. By Theorem 4, φ subsumes φ_i. Hence $n \varepsilon M$.

Let j be any integer such that $n \geq j > 4$, and assume that $j \varepsilon M$. Let φ be a clause of dimension $j - 1$, such that φ implies f. Now $n > j-1$, and therefore not all n letters can appear in φ. Then there is some literal X_K, where X_K does not appear in φ and X_K' does not appear in φ. Consider the clause φX_K. The clause φX_K implies f and is of dimension j. By the induction hypothesis, φX_K subsumes some clause φ_i of $\widetilde{P}$. Next consider the clause $\varphi X_K'$. The clause $\varphi X_K'$

implies f and is of dimension j. Hence $\varphi X_K'$ subsumes some clause φ_t of $\tilde{P}$. Now clearly the literal X_K' does not appear in φ_i since φX_K subsumes φ_i, and therefore φX_K implies φ_i. Therefore, if the literal X_K does not appear in φ_i, every literal which does appear in φ_i appears in φ, and φ subsumes φ_i.

Suppose that the literal X_K does appear in φ_i. Then consider φ_t.

Clearly the literal X_K does not appear in φ_t, since $\varphi X_K'$ subsumes φ_t, and hence $\varphi X_K'$ implies φ_t. If then the literal X_K' does not appear in φ_t, every literal which appears in φ_t appears also in φ and φ subsumes φ_t. Suppose that the literal X_K' does appear in φ_t. Then X_K appears in φ_i and X_K' appears in φ_t.

Claim: The dimension of φ_i is 2 and the dimension of φ_t is 2. Indeed: It can not be that the dimension of φ_t is 1 and the dimension of φ_i is 1. For if so, then $\varphi_i = X_K$ and $\varphi_t = X_K'$. But then $\varphi_i + \varphi_t = I$, where I is the identity element. But $(\varphi_i + \varphi_t)$ implies f, and this would mean that f is the identity element, contrary to the hypothesis. Also, it can not be that φ_i is of dimension 2 and φ_t is of dimension 1. For if so, $\varphi_i = YX_K$ and $\varphi_t = X_K'$. Then

$$\varphi_i + \varphi_t = yX_K + X_K' = X_K' + y.$$

But then the literal y implies $\varphi_i + \varphi_t$ and $\varphi_i + \varphi_t$ implies f. Hence y implies f, and φ_i is not a prime implicant of f, another contradiction. Similarly, it can not be that φ_t is of dimension 2 and φ_i is of dimension 1. Therefore both φ_i and φ_t are of dimension 2, where X_K appears in φ_i and X_K' appears in φ_t.

Write $\varphi_i = y_1 X_K$ and $\varphi_t = y_2 X_K'$.

Then φX_K subsumes $y_1 X_K = \varphi_i$ and

$\varphi X_K'$ subsumes $y_2 X_K' = \varphi_t$.

Hence φ subsumes y_1 and

φ subsumes y_2

Then φ subsumes $y_1 y_2$.

Claim: $y_1 y_2$ implies f. For suppose that $y_1 y_2(P) = 1$. Then $y_1(P) = 1$ and $y_2(P) = 1$. Either $X_K(P) = 1$, in which case $\varphi_i(P) = 1$ and $f(P) = 1$, or $X_K(P) = 0$, in which case $X_K'(P) = 1$, $\varphi_t(P) = 1$ and $f(P) = 1$. Thus $y_1 y_2$ implies f. y_1 can not imply f since $\varphi_1 = y_1 X_K$ is a prime implicant of f; similarly, y_2 can not imply f. Hence $y_1 y_2$ implies f and subsumes no clause of smaller dimension which implies f. Thus, $y_1 y_2$ is a prime implicant of f of dimension 2. Hence $y_1 y_2 \varepsilon \tilde{P}$, and φ subsumes $y_1 y_2$. Therefore, in all eventualities,

φ subsumes some clause of $\tilde{P}$. Hence if $j \varepsilon M$, $j - 1 \varepsilon M$, for $j = n, n - 1, \ldots 4$, which completes the proof.

There is an immediate generalization of Theorem 5 which comes to mind, but which, unfortunately, is not true. Let $f \varepsilon \tilde{B}_n$, f not the zero element, and f not the identity element. Let $\tilde{P}$ denote the set of all the prime implicants of f of dimension K or less. Now suppose that $\sum_{\varphi \varepsilon \tilde{P}} \varphi = f$. Then if $K < n$, a reasonable question is: "Is every prime implicant of f an element of $\tilde{P}$?" The answer is, "not necessarily." The proof of Theorem 5 is almost applicable in this case. For if $K < n$, by the argument of Theorem 5 we can show that no clause of dimension n is a prime implicant of f. However, the trouble arises when we attempt to get from n to $n - 1$. Let φ be a clause of dimension $n - 1$ which implies f. Then, as in Theorem 5, we determine two clauses φ_i and φ_t of dimension n such that

$$\varphi X_K \text{ subsumes } \varphi_i \text{ and}$$
$$\varphi X_K' \text{ subsumes } \varphi_t \quad .$$

Now if X_K appears in φ_i and X_K' appears in φ_t, we have $\varphi_i = \xi_1 X_K$ and $\varphi_t = \xi_2 X_K'$. And again we have that φ subsumes $\xi_1 \xi_2$. However we can not conclude that $\xi_1 \xi_2$ is of dimension K, nor can we conclude that

$\xi_1\xi_2$ is not a prime implicant of f. It is at this point that the proof of Theorem 5 fails.

Example: Let $f = X_1X_2X_5' + X_3X_4X_5$.

Then $X_1X_2X_5'$ and $X_3X_4X_5$ are prime implicants of f, and these are all the prime implicants of f of dimension 3 or less. But $X_1X_2X_3X_4$ is also a prime implicant of f.

There is however, one immediate Corollary of Theorem 5:

Corollary 5-1. Let $f \varepsilon \tilde{B}_n$, f not the identity element, and f not the zero element. Let $\tilde{P}$ be the set of all the clauses of dimension 1 which imply f, and suppose that $\sum_{y \varepsilon \tilde{P}} y = f$. Then every prime implicant of f is an element of $\tilde{P}$.

Proof: By Theorem 5, no clause of dimension greater than 2 can be a prime implicant of f. Let $y_K y_j$ be a clause of dimension 2 which implies f.

If either of y_K or y_j is in $\tilde{P}$, then $y_K y_j$ is not a prime implicant of f. Suppose that $y_K \notin \tilde{P}$ and $y_j \notin \tilde{P}$. Then $y_K' \notin \tilde{P}$. For if $y_K' \varepsilon \tilde{P}$, then since $y_K' + y_K y_j = y_K' + y_j$, y_j would imply f, and would therefore be an element of $\tilde{P}$. Hence y_K and y_K' are not in $\tilde{P}$. Hence f is independent of y_K, and by Lemma 12, $y_K y_j$ is not a prime implicant of f.

Therefore no clause of dimension 2 is a prime implicant of f. It follows that every prime implicant of f is an element of $\tilde{P}$.

Theorem 5 and Corollary 5-1 justify the following termination criteria:

Let $f \varepsilon \tilde{B}_n$, where f is not the identity element, and f is not the zero element. Then,

1) If $\tilde{P}$ is the set of all the clauses of dimension 1 which imply f, and if the sum over $\tilde{P}$

represents f, then every prime implicant of f is an element of $\tilde{P}$, and the algorithm can be terminated once these clauses have been determined.

2) If $\tilde{P}$ is the set of all the prime implicants of f of dimension 2 or less, and if the sum over $\tilde{P}$ represents f, then every prime implicant of f is an element of $\tilde{P}$, and the algorithm can be terminated once these clauses have been determined.

3) If $\tilde{P}$ is the set of all the prime implicants of f of dimension $n-1$ or less, and if the sum over $\tilde{P}$ represents f, then every prime implicant of f is an element of $\tilde{P}$, and the algorithm can be terminated once these clauses have been determined.

We can now write the modified algorithm in this manner: Given the representation ψ of the function f in $\tilde{B}_n$, where f is not the zero element,

Step 1. From ψ, determine the f-matrix. If f is the identity element, terminate the algorithm. Each representation $X_i + X_i'$, $i = 1, 2, \ldots, n$, is a simplest representation of f. If f is not the identity element, compute the rank $\rho(f)$ and proceed to step 2.

Step 2. Determine the smallest integer j such that $0 < j \leq n$, and $2^{n-j} \leq \rho(f)$. Each candidate φ, of dimension j which implies f is a prime implicant of f. The clauses so determined are all of the prime implicants of f of dimension j. No clause of dimension less than j is a prime implicant of f. If $j = 1$, proceed to A). If $j = 2$, proceed to B). If $j > 2$, let the index K of Step 3 have value j and proceed to Step 3.

A) $j = 1$. Each clause φ of dimension 1 which implies f is a prime implicant of f. If the sum of the clauses of dimension 1 which imply f represents f, terminate the algorithm. The clauses so determined are all of the prime implicants of f. If the sum of the clauses so determined does not represent f, consider the candidates of dimension 2. Eliminate from consideration each candidate of dimension 2 which subsumes a prime implicant of dimension 1. The remaining candidates of dimension 2 which imply f are all of the prime implicants of dimension 2. If the sum of all of the prime implicants of dimension 1 and all the prime implicants of dimension 2 represents f, then every prime implicant of f is of either dimension 1 or dimension 2. Hence all prime implicants have been determined. Terminate the algorithm. If the sum of all the prime implicants of dimension 1 and all prime implicants of dimension 2 does not represent f, let the index K of Step 3 have value 3, and proceed to Step 3.

B) $j = 2$. Each clause of dimension 2 which implies f is a prime implicant of f. If the sum of the clauses of dimension 2 which imply f represents f, terminate the algorithm. Every prime implicant of f is of dimension 2, and has been determined. If the sum of the clauses of dimension 2 which imply f does not represent f, let the index K of Step 3 have value 3 and proceed to Step 3.

Step 3. Consider the candidates of dimension K. Eliminate from consideration each candidate of dimension K which subsumes a prime implicant of dimension less than K. The remaining candidates which imply f are all of the prime implicants of dimension K. Proceed to Step 4.

Step 4. If the value of j of Step 2 is n, or if the clauses of dimension $n - 1$ have been subjected to the process, proceed to C). If the value of the index j of Step 2 is not n, and the clauses of dimension $n - 1$ have not been subjected to the process, increase the index K of Step 3 by 1 and proceed to Step 3.

C) If the clauses of dimension n have been subjected to the process, proceed to E). If the clauses of dimension n have not been subjected to the process, proceed to D).

D) If the sum of the prime implicants already determined represents f, terminate the algorithm. No prime implicant has dimension n. Hence all prime implicants have been determined. If the sum of the prime implicants already determined does not represent f, let the index K of Step 3 have value n, and proceed to Step 3.

E) Terminate the algorithm. All prime implicants have been determined.

Example 2. We again consider the function of Example 1:

$$f = X_1X_2' + X_1'X_2 + X_2X_3' + X_2'X_3.$$

We first compute the f-matrix:

	X_1	X_2	X_3	f
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	0

From this, we have $\rho(f) = 6$, and j in Step 2 is 1, for $2^{3-1} = 4 < 6$. We proceed to A). We see that no clause of dimension 1 implies f, X_1, X_2, and X_3 are eliminated by row 1; X_1', X_2', and X_3' are eliminated by row 8. Next, we determine the clauses of dimension 2 which imply f. These are: $X_1'X_3$, $X_2'X_3$, $X_1'X_2$, X_2X_3', X_1X_2', and X_1X_3'. These surely recover the function, and we terminate the algorithm.

Comparing Example 2 with Example 1, it is evident that the refined algorithm does indeed possess some advantages over the original version.

It is perhaps appropriate here to examine briefly the theory from the view-point of actual application. The basic concept of the algorithm is simply this: The algorithm determines a sequence of tests by which certain clauses are eliminated. The remaining clauses are the prime implicants of the given function. Now one of the main difficulties in the application of the theory can be attributed to the number of tests which must be completed in order that the process can be terminated. And in general, the time required, even for a high speed computer, to complete the sequence of tests is prohibitive.

In computer applications there is a certain type of problem, the so-called real-time problem, in which the time required to solve the problem is critical. Now quite often in the solution of a real-time problem, the computing system is programmed to function not only as a computing unit, but also as a library system. For example, in a given trajectory problem, it might be necessary to perform some type of arithmetical computation involving the number $\sin X_o$. Now given the number X_o, the computer can surely compute an approximation to the number $\sin X_o$. However, it might not be expedient for the computer to actually compute the number.

Instead, the following alternative might very well be used:

Before the computations begin, a table of pairs $(X, \sin X)$ is stored in the memory of the computer. Thus, given the number X_o, the computer performs a table look-up and an interpolation to determine the approximation to $\sin X_o$.

Is it possible to apply a similar table look-up procedure in our simplification problem? Consider the following possibility:

We first determine one simplest representation for each function in $\tilde{B}_n$. We index each function in $\tilde{B}_n$, so that for each such function there corresponds exactly one integer i, $1 \leq i \leq (2)^{2^n}$. We sequence the representations of the functions according to the index of each function. The representations are then stored, in their sequential order in a memory device of a computing system. We program the computer in a manner such that: Given a representation ψ of a function in $\tilde{B}_n$, we in-put ψ to the computer. The computer determines from ψ, say, the corresponding function matrix. From this matrix, the computer determines the index of the function, and from this index, the computer determines the location in memory where the representation is stored. This representation is printed as out-put.

For what order n of the algebra $\tilde{B}_n$ would the above procedure be workable? The cardinality of $\tilde{B}_4$ is 65, 536, and the procedure would probably be workable for $\tilde{B}_4$. But the cardinality of $\tilde{B}_5$ is 4, 294, 967, 296. Thus, with the existing state of computer technology, it is improbable that a complete table look-up procedure would be workable for any higher algebra than $\tilde{B}_4$. And this of course means that the individual functions will have to be simplified as the need arises.

There is a restricted class of functions in $\tilde{B}_n$ whose prime implicants can be determined without subjecting the functions to the algorithm. Before considering these functions we state the dualization laws:

If $f \varepsilon \tilde{B}_n$ and $g \varepsilon \tilde{B}_n$, then

$$(f + g)' = f' \cdot g' \quad .$$

We also need the law of involution:

$$(f')' = f.$$

It is easily verified that these relations are true in the model with which we are working, since they clearly hold in the algebra of Definition 1.

Suppose that we are given a function f in $\tilde{B}_n$ such that $\rho(f) = 1$. By Theorem 1, no clause of dimension less than n implies f. By Corollary 1-2, there is exactly one clause of dimension n which implies f. This clause is therefore the only prime implicant of f. Further, by Corollary 1-1, the row of D_n which corresponds to the row of the f-matix in which 1 appears determines this one prime implicant of f.

Now suppose that f has rank $2^n - 1$. In this event, there is only one row of the f-matrix in which 1 does not appear. Hence there is exactly one row of the f'-matrix in which 1 appears. That is, the rank of f' is 1. Thus there is exactly one prime implicant of f'. Again, this prime implicant of f' is of dimension n, and is determined by the row of D_n which corresponds to the row of the f'-matrix in which 1 appears. Denoting this clause by $(y_1 \cdot y_2 \ldots y_i \ldots y_n)$ we have

$$f' = y_1 \cdot y_2 \cdot\cdot y_i \ldots \cdot y_n.$$ Using the dualization laws, we have

$$(f')' = (y_1 \cdot y_2 \ldots y_i \ldots y_n)' = y_1' + y_2' + y_i' \ldots + y_n' .$$

Next, by the involution law $(f')' = f$, and therefore

$$f = y_1' + y_2' + \ldots y_i' + \ldots + y_n'.$$

Hence, f can be represented by a sum of clauses of dimension 1. It is clear that no other clause of dimension 1 can imply f, for if, say, y_i also implies f, we would have

$$f = y_1' + y_2' + \ldots. y_i' + \ldots + y_n' + y_i ,$$

and

$$f' = y_1 y_2 \ldots y_i \ldots y_n \cdot y_i' = \bar{\Phi} ,$$

and hence f would be the identity element, contrary to the hypothesis that $\rho(f) = 2^n - 1$. We then have the following lemma:

Lemma 13.

Let $f \in \tilde{B}_n$ such that f has rank $2^n - 1$. Then there are exactly n clauses of dimension 1 which imply f. Further, the sum of these clauses represents f.

From Lemma 13 and Corollary 5-1, we have:

<u>Theorem</u> <u>6</u>.

A function f in $\tilde{B}_n$ with rank $2^n - 1$ has exactly n prime implicants. Each of these prime implicants is of dimension 1.

Example: Let $f = X_1'X_2' + X_1'X_2 + X_2'X_3 + X_1X_2$. Then we have

	X_1	X_2	X_3	f
1	0	0	0	1
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

From row 5, the prime implicants of f are X_1', X_2, and X_3.

The method of the last example is also applicable to functions of rank $2^n - 2$. For let $f \in \tilde{B}_n$, such that f has rank of $2^n - 2$. Then there are exactly two rows of the f-matrix in which 1 does not appear. Hence, the function f' has rank 2 and can be represented as the sum of two clauses of dimension n. Denote these clauses by $y_1 y_2 \cdots y_n$ and $z_1 z_2 \cdots\cdots z_n$. Then

$$f' = (y_1 \cdot y_2 \cdot \cdots\cdots \cdot y_n) + (z_1 \cdot z_2 \cdot \cdots\cdots \cdot z_n)$$

Hence $f = (y_1' + y_2' + \cdots\cdots y_n') + (z_1' + z_2' + \cdots\cdots z_n') =$

$$\sum_{i=1}^{n} \left[\sum_{j=1}^{n} y_i' \, z_j' \right] .$$

Hence we have the following lemma:

Lemma 14:

Every function f in $\tilde{B}_n$ with rank $2^n - 2$ can be represented as the sum of clauses, each clause of dimension 2 or less.

Combining Lemma 14 and Theorem 5, we have:

<u>Theorem 7</u>:

Let f be a function in $\tilde{B}_n$ such that f has rank $2^n - 2$. Then no prime implicant of f has dimension greater than 2.

Theorem 7 suggests the following method of determining the prime implicants of a function f of rank $2^n - 2$: Consider the two rows of the f-matrix in which 1 does not appear. Let P_1 and P_2 denote the corresponding rows of D_n. From P_1 and P_2 we first determine the clauses of dimension 1 which imply f. We next determine, again from P_1 and P_2, the clauses of dimension 2 which do not imply f. From the $2^2\binom{n}{2} = 2n(n-1)$ clauses of $\tilde{B}_n$ of dimension 2, we first eliminate those clauses which do not imply f. Next, we eliminate those clauses which subsume clauses of dimension 1 which imply f. The remaining clauses, along with the clauses of dimension 1 which imply f, are the prime implicants of f.

Example 3. Consider the function f of $\tilde{B}_4$ where

$$f' = X_1X_2'X_3'X_4' + X_1X_2X_3X_4.$$

Now f' has rank 2. Hence f has rank 14. Consider the two rows of $\tilde{B}_4$,

X_1	X_2	X_3	X_4	f
1	0	0	0	0
1	1	1	1	0

From this, we see that the only clause of dimension 1 which implies f is X_1'. Also, the only clauses of dimension 2 which do not imply f are: X_1X_2', X_1X_3', X_1X_4', $X_2'X_3'$, $X_2'X_4'$, $X_3'X_4'$, X_1X_2, X_1X_3, X_1X_4, X_2X_3, X_2X_4, and X_3X_4. We immediately determine all of the prime implicants of f:

X_1X_2	X_1X_3	X_1X_4	X_2X_3	X_2X_4	X_3X_4
X_1X_2'	X_1X_3'	X_1X_4'	X_2X_3'	X_2X_4'	X_3X_4'
$X_1'X_2$	$X_1'X_3$	$X_1'X_4$	$X_2'X_3$	$X_2'X_4$	$X_3'X_4$
$X_1'X_2'$	$X_1'X_3'$	$X_1'X_4'$	$X_2'X_3'$	$X_2'X_4'$	$X_3'X_4'$

The remaining clauses of dimension 2, along with X_1' are the prime implicants of f. That is, the prime implicants of f are X_1', X_2X_3', $X_2'X_3$, X_2X_4', $X_2'X_4$, X_3X_4', and $X_3'X_4$.

There is another type of function in $\tilde{B}_n$ whose prime implicants can be determined under a reduced number of tests. By Lemma 12, if $f \varepsilon \tilde{B}_n$, such that f is not the identity element, and if f is independent of X_K, $1 \leq K \leq n$, then no clause in which X_K or X_K' appears is a prime implicant of f. The next theorem might be useful in working with such functions.

Theorem 8.

Let n be an integer, $n > 1$ and let $\bar{\tilde{B}}$ be that subset of elements of $\tilde{B}_n$ consisting of all the functions in $\tilde{B}_n$ which are independent of the variable X_K, where $1 \leq K \leq n$. Then $\bar{\tilde{B}}$, as embedded in $\tilde{B}_n$ is isomorphic to $\tilde{B}_{n-1}$.

Proof: For every $P \in \tilde{D}_n$ there is exactly one $q \in \tilde{D}_n$ such that $(P \sim q)$ by X_K. There are then $2^n/2 = 2^{n-1}$ such equivalent pairs in $\tilde{D}_n$. Denote the set of these equivalent pairs by $\bar{D}_{n-1} = \{(P,q)_1, (P,q)_2, \ldots. (P,q)_{2^{n-1}}\}$. Note that $\bar{D}_{n-1}$ and $\tilde{D}_{n-1}$ are in 1-1 correspondence. Let r_i be the $(n-1)$-tuple obtained from $(P,q)_i \in \bar{D}_{n-1}$ by the deletion of the K^{th} coordinate of P. Every function in $\bar{\tilde{B}}$ associates with each element of $\bar{D}_{n-1}$ an element of $s = \{0,1\}$. For if $\bar{f} \in \bar{\tilde{B}}$ and $(P,q) \in \bar{D}_{n-1}$, then $\bar{f}$ associates with (P,q) the element $f(P) = f(q)$ in S.

Conversely, any mapping of $\bar{D}_{n-1}$ into S corresponds to a function in $\bar{\tilde{B}}$. Hence $\bar{\tilde{B}}$ can be considered as the set of all functions which map $\bar{D}_{n-1}$ into S. By Lemma 2 the cardinality of $\bar{\tilde{B}}$ is $(2)^{2^{n-1}}$. Hence $\bar{\tilde{B}}$ and

$\tilde{B}_{n-1}$ are in 1-1 correspondence . To determine the isomorphism, we associate with each function $\bar{f}$ in $\bar{\tilde{B}}$ which maps $(P,q)_i$ onto $a \varepsilon S$, the function f in $\tilde{B}_{n-1}$ which maps r_i onto a, for $i = 1, 2, \ldots n-1$. Denote this correspondence by F. Then F is the desired isomorphism. To prove this last statement, first note that $\bar{\tilde{B}}$ is closed under the operations +, •, and ' in $\tilde{B}_n$. For let $\bar{\alpha}$ and $\bar{\beta}$ be elements of $\bar{\tilde{B}}$. Then for each n-tuple $(b_1 \ldots b_K, \ldots b_n)$ in $\tilde{D}_n$,

$\bar{\alpha}\,(b_1, \ldots b_K, \ldots b_n) = \bar{\alpha}\,(b_1, \ldots b'_K, \ldots b_n)$ and
$\bar{\beta}\,(b_1, \ldots b_K, \ldots b_n) = \bar{\beta}\,(b_1, \ldots b'_K, \ldots b_n)$. Hence
$[\bar{\alpha} \dotplus \bar{\beta}]\,(b_1, \ldots b_K, \ldots b_n) =$
$\bar{\alpha}\,(b_1, \ldots b_K, \ldots b_n) \dotplus \bar{\beta}\,(b_1, \ldots b_K, \ldots b_n) =$
$\bar{\alpha}\,(b_1, \ldots b'_K, \ldots b_n) \dotplus \bar{\beta}\,(b_1, \ldots b'_K, \ldots b_n) =$
$[\alpha \dotplus \beta]\,(b_1, \ldots b'_K, \ldots b_n)$, and $\alpha \dotplus \beta$ are independent of X_K. Hence $\bar{\tilde{B}}$ is closed under the multiplication and addition of $\tilde{B}_n$.

Further,

$\bar{\alpha}'\,(b_1, \ldots b_K, \ldots b_n) =$
$[\bar{\alpha}\,(b_1, \ldots b_K, \ldots b_n)]' = [\bar{\alpha}\,(b_1, \ldots b'_K, \ldots b_n)]' =$
$\bar{\alpha}'\,(b_1, \ldots b'_K, \ldots b_n)$, and $\bar{\tilde{B}}$ is closed with respect to the operation ' in $\tilde{B}_{n+1}$.

Let $F(\bar{\alpha}) = \alpha$ and $F(\bar{\beta}) = \beta$.

Let $(b_1, \ldots b_{K-1}, b_{K+1}, \ldots b_n) \varepsilon \tilde{D}_{n-1}$, and suppose that

$\alpha(b_1, .. b_{K-1}, b_{K+1}, .. b_n) = a$ and

$\beta(b_1, .. b_{K-1}, b_{K+1}, .. b_n) = b.$

Then $[\alpha + \beta](b_1, .. b_{K-1}, b_{K+1}, .. b_n) = a + b$

But $\overline{\alpha}(b_1, .. b_{K-1}, b_K, b_{K+1}, .. b_n) = a$ and

$\overline{\beta}(b_1, .. b_{K-1}, b_K, b_{K+1}, .. b_n) = b.$

Therefore, $[\overline{\alpha} + \overline{\beta}](b_1, .. b_{K-1}, b_K, b_{K+1}, .. b_n) = a + b,$ and hence $[F(\overline{\alpha} + \overline{\beta}](b_1, .. b_{K-1}, b_{K+1}, .. b_n) = a + b =$

$\alpha(b_1, .. b_{K-1}, b_{K+1}, .. b_n) + \beta(b_1, .. b_{K-1}, b_{K+1}, .. b_n)$

$= [F(\overline{\alpha})](b_1, .. b_{K-1}, b_{K+1}, .. b_n)$

$+ [F(\overline{\beta})](b_1, .. b_{K-1}, b_{K+1}, .. b_n).$

Hence $F(\overline{\alpha}) + F(\overline{\beta}) = F(\overline{\alpha} + \overline{\beta})$. Similarly,

$F(\overline{\alpha}) \cdot F(\overline{\beta}) = F(\overline{\alpha} \cdot \overline{\beta}).$

Further, let

$[F(\overline{\alpha}')](b_1, .. b_{K-1}, b_{K+1}, .. b_n) = C.$ Then

$C = \overline{\alpha}'(b_1, .. b_{K-1}, b_K, b_{K+1}, .. b_n) =$

$[\overline{\alpha}(b_1; .. b_{K-1}, b_K, b_{K+1}, .. b_n)]' =$

$[\,[F(\overline{\alpha})](b_1, .. b_{K-1}, b_{K+1}, .. b_n)\,]' =$

$[F(\overline{\alpha})]'(b_1, .. b_{K-1}, b_{K+1}, .. b_n)$. Hence

$F(\overline{\alpha}') = [F(\overline{\alpha})]'$. Hence $\overline{\tilde{B}}$ as embedded in $\tilde{B}_n$, is isomorphic to $\tilde{B}_{n-1}$.

Note that $\overline{\alpha}$ implies $\overline{\beta}$ if and only if $F(\overline{\alpha})$ implies $F(\overline{\beta})$. For if $\overline{\alpha}$ implies $\overline{\beta}$, let $\overline{\gamma} = \overline{\alpha}'\overline{\beta}$. Then $\overline{\gamma} \varepsilon \overline{\tilde{B}}$ and $\overline{\alpha} + \overline{\gamma} = \overline{\alpha} + \overline{\alpha}'\overline{\beta} = \overline{\alpha} + \overline{\beta} = \overline{\beta}$. Hence

$F(\overline{\alpha} + \overline{\gamma}) = F(\overline{\beta}) = F(\overline{\alpha}) + F(\overline{\gamma})$ and $F(\overline{\alpha})$ implies $F(\overline{\beta})$.

Next, if $F(\overline{\alpha})$ implies $F(\overline{\beta})$, then there exists $F(\overline{\gamma})$ such that $F(\overline{\alpha}) + F(\overline{\gamma}) = F(\overline{\beta})$. Hence

$F^{-1}[F(\overline{\alpha}) + F(\overline{\gamma})] = \overline{\beta} = \overline{\alpha} + \overline{\gamma}$ and $\overline{\alpha}$ implies $\overline{\beta}$.

Theorem 9.

Let $\overline{f}$ be a function in $\tilde{B}_n$, $n > 1$, where $\overline{f}$ is not the zero element and $\overline{f}$ not the identity element. Denote the independent variables of $\tilde{B}_n$ by $\overline{X}_i$ $i = 1, 2, \ldots n$. Assume that $\overline{f}$ is independent of $\overline{X}_K$. Let $\overline{\tilde{B}}$ be that subset of $\tilde{B}_n$ consisting of those functions in $\tilde{B}_n$ which are independent of $\overline{X}_K$. Let $F: \overline{\tilde{B}} \to \tilde{B}_{n-1}$ be the isomorphism of Theorem 8. Let $F(\overline{f}) = f$. Then:

1) Every prime implicant of $\overline{f}$ is an element of $\overline{\tilde{B}}$ and

2) If φ is a clause of $\tilde{B}_{n-1}$, then φ is a prime implicant of f if and only if $F^{-1}(\varphi)$ is a prime implicant of $\overline{f}$.

Proof:

1) Every prime implicant of $\overline{f}$ is an element of $\overline{\tilde{B}}$. $\overline{f}$ is not the identity element. Since $\overline{f}$ is independent of $\overline{X}_K$, it follows from Lemma 12 that the literal $\overline{X}_K$ does not appear in any prime implicant of $\overline{f}$, and the literal $\overline{X}_K'$ does not appear in any prime implicant of $\overline{f}$. Hence every prime implicant of $\overline{f}$ is independent of $\overline{X}_K$, and is therefore an element of $\overline{\tilde{B}}$.

2) If φ is a clause of $\tilde{B}_{n-1}$, then φ is a prime

implicant of f if and only if $F^{-1}(\varphi)$ is a prime implicant of $\overline{f}$.

Suppose φ is a clause of $\tilde{B}_{n-1}$ and assume that φ is a prime implicant of f. Denote $F^{-1}(\varphi)$ by $\overline{\varphi}$. Then $\overline{\varphi}$ must be a prime implicant of $\overline{f}$. For suppose not. Clearly $\overline{\varphi}$ implies $\overline{f}$, since φ implies f, and F^{-1} is an isomorphism. Then $\overline{\varphi}$ subsumes some clause $\overline{\xi}$ of ~~smaller~~ dimension, where $\overline{\xi}$ implies $\overline{f}$. Now $\overline{\varphi} = F^{-1}(\varphi)$, and $\varphi \varepsilon \overline{\tilde{B}}$. Hence $\overline{\varphi}$ is independent of $\overline{X}_K$. Thus $\overline{\xi}$ is independent of $\overline{X}_K$ since $\overline{\varphi}$ subsumes $\overline{\xi}$. Then $\overline{\xi} \varepsilon \tilde{B}$. Hence $F(\overline{\xi})$ is defined. Let $F(\overline{\xi}) = \xi$. Then $\overline{\varphi}$ implies $\overline{\xi}$ and $\overline{\xi}$ implies $\overline{f}$. Hence

$$F(\overline{\varphi}) = \varphi \text{ implies } F(\overline{\xi}) = \xi \text{ implies } F(\overline{f}) = f.$$

By Theorem 4, φ subsumes ξ. But dimension $\overline{\varphi}$ = dimension φ and dimension $\overline{\xi}$ = dimension ξ. Hence φ is not a prime implicant of f. Contradiction.

Next, suppose that φ is a clause of $\tilde{B}_{n-1}$ such that $F^{-1}(\varphi)$ is a prime implicant of $\overline{f}$. Denote $F^{-1}(\varphi)$ by $\overline{\varphi}$. Then φ must be a prime implicant of f. For suppose not. Then φ subsumes a clause ξ of smaller dimension which implies f. Denote $F^{-1}(\xi)$ by $\overline{\xi}$. By Theorem 4, φ implies ξ. Hence by the isomorphism F^{-1}, $\overline{\varphi}$ implies $\overline{\xi}$ and $\overline{\xi}$ implies $\overline{f}$. By Theorem 5, $\overline{\varphi}$ subsumes $\overline{\xi}$ and $\overline{\varphi}$ is not a prime implicant of $\overline{f}$. Contradiction. Thus, φ is a prime implicant of f if and only if $F^{-1}(\varphi)$ is a prime implicant of $\overline{F}$.

Example 4. Consider the function f in $\tilde{B}_4$:

$$f = X_1 + X_1'X_2X_3X_4 + X_1'X_2X_3X_4' + X_1'X_2X_3'X_4 + X_1'X_2X_3'X_4' + X_1'X_2'X_3X_4 + X_1'X_2'X_3'X_4$$

X_1	X_2	X_3	X_4	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

From D_4 we see that f is independent of X_3. By the isomorphism of Theorem 8, we have:

X_1	X_2	X_4	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

and from this image, we have

$$X_1 + X_2 + X_4 = f$$

Hence the prime implicants of f are X_1, X_2, and X_4.

The following results lead to a theorem which might be useful in determining if a function is independent of an independent variable.

Lemma 15.

Let $P = (a_{i1}, a_{i2}, \ldots a_{in})$ be the i^{th} row of D_n. Let $q = (b_{j1}, b_{j2}, \ldots b_{jn})$ be the j^{th} row of D_n.

Let $W(P) = 1 + \sum_{r=1}^{n} 2^{n-r} a_{ir}$ and

let $W(q) = 1 + \sum_{r=1}^{n} 2^{n-r} b_{jr}$.

Then $P = q$ if and only if $W(P) = W(q)$.

Proof: If $P = q$, then clearly $W(P) = W(q)$.

Suppose that $W(P) = W(q)$. Then

$$\sum_{r=1}^{n} 2^{n-r} a_{ir} = \sum_{r=1}^{n} 2^{n-r} b_{jr} .$$

We proceed by induction on the index r.

First, $a_{i1} = b_{j1}$. For suppose not. Assume without loss of generality that $a_{i1} = 0$ and $b_{j1} = 1$. Then

$$\sum_{r=2}^{n} 2^{n-r} a_{ir} = 2^{n-1} + \sum_{r=2}^{n} 2^{n-r} b_{jr} .$$

The maximum value of the left side of the above equation is the sum of a geometric series of $n-1$ terms with first term 1 and common ratio 2. That is, the maximum value of the left side is $2^{n-1} - 1$. But the minimum value of the right side is 2^{n-1}. Contradiction.

Hence $a_{i1} = b_{j1}$. Suppose $a_{ir} = b_{jr}$ for all r such that $1 \leq r < K \leq n$. Then $a_{iK} = b_{iK}$

For since $\sum_{r=1}^{n} 2^{n-r} a_{ir} = \sum_{r=1}^{n} 2^{n-r} b_{jr}$,

then $\sum_{r=K}^{n} 2^{n-r} a_{ir} = \sum_{r=K}^{n} 2^{n-r} b_{jr}$. By the same argument as above, $a_{iK} = b_{iK}$. Hence $a_{ir} = b_{ir}$ $r = 1, 2, \ldots n$, which establishes the proposition.

Lemma 16.

Let i be any integer such that $1 \leq i \leq 2^n$. Then there is exactly one row $P = (a_{j1}, a_{j2}, \ldots a_{jn})$ of D_n such that $W(P) = i$, where

$$W(P) = 1 + \sum_{r=1}^{n} 2^{n-r} a_{jr} .$$

Proof: Clearly, $\min\{W(P) \mid P \text{ is a row of } D_n\}$ is 1 and $\max\{W(P) \mid P \text{ is a row of } D_n\}$ is 2^n. D_n has exactly 2^n rows, and by Lemma 15, W is 1-1 on the set of these rows. Hence W is 1-1 on the set of rows of D_n and onto $\{1, 2, \ldots 2^n\}$. Hence if $1 \leq i \leq 2^n$, there is exactly one row P of D_n that $W(P) = i$.

Lemma 16 states that if $1 \leq i \leq 2^n$, then there is some row P of D_n such that $W(P) = i$. Lemma 17 states that this in fact is the i^{th} row.

Lemma 17.

The i^{th} row, $P = (a_{i1}, a_{i2}, \ldots a_{in})$ is the only row of D_n such that:

$$\left[\sum_{j=1}^{n} 2^{n-j} a_{ij}\right] + 1 = i \ .$$

Proof is by induction on n.

$$D_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} .$$

For row 1: $2^{1-1}\ 0 + 1 = 1$.

For row 2: $2^{1-1}\ 1 + 1 = 2$.

Hence the statement holds for D_1. Assume that it holds for D_n.

Let i be any row number of any row of D_{n+1}.

By Definition 3, $D_{n+1} = \left(\begin{array}{c|c} \theta & D_n \\ \hline I & D_n \end{array}\right)$

Case 1. $i \leq 2^n$. Then the i^{th} row of D_{n+1} is of the form $(0, a_{i2}, a_{i3}, \ldots a_{in+1})$, where

$(a_{i2}, a_{i3}, \ldots a_{in+1})$ is the i^{th} row of D_n. By the induction hypothesis,

$$\left[\sum_{j=1}^{n} 2^{n-j} a_{ij+1}\right] + 1 = i$$

$$2^n \cdot 0 + \left[\sum_{j=1}^{n} 2^{n-j} a_{ij+1}\right] + 1 = i \ .$$

Case 2. $2^n < i \leq 2^{n+1}$

Let $i - 2^n = r$. Then the i^{th} row of D_{n+1} is of the form $(1, a_{r2}, a_{r3}, \ldots a_{rn+1})$ where $(a_{r2}, a_{r3}, \ldots a_{rn+1})$ is the r^{th} row of D_n. By the induction hypothesis,

$$\left[\sum_{j=1}^{n} 2^{n-j} a_{rj+1}\right] + 1 = r. \quad \text{Hence}$$

$$2^n \cdot 1 + \left[\sum_{j=1}^{n} 2^{n-j} a_{rj+1}\right] + 1 = 2^n + r = i .$$

Hence the i^{th} row $P = (a_{i1}, a_{i2}, \ldots a_{in})$ is such that

$$1 + \sum_{r=1}^{n} 2^{n-r} a_{ir} = i.$$ By Lemma 16, the i^{th} row is is the only row with this property.

<u>Theorem 10</u>.

Let P be the i^{th} row of D_n. Then if the K^{th} coordinate of P is 0, the row number of the row q, where $(P \simeq q)$ by X_K, is $i + 2^{n-K}$.

Proof: q is obtained from P by changing the K^{th} coordinate of P to 1 and leaving fixed all others.

Let $P = (a_{i1}, \ldots a_{iK-1}, 0, a_{iK+1}, \ldots a_{in})$. Then

$$q = (a_{i1}, \ldots a_{iK-1}, 1, a_{iK+1}, \ldots a_{in}).$$

By Lemma 17,

$$\left[\sum_{\substack{j=1\\P}}^{n} 2^{n-j} a_{ij}\right] + 1 = i.$$

Hence $\left[\sum_{\substack{j=1\\q}}^{n} 2^{n-j} a_{ij}\right] + 1 = i + 2^{n-K}$.

Now $1 < i + 2^{n-K} \leq 2^n$

By Lemma 16, there is exactly one row of $\widetilde{D}_n$ such that

$1 + \sum_{\substack{j=1\\q}}^{n} 2^{n-j} a_{ij} = i + 2^{n-K}$, and by Lemma 17,

this is row $i + 2^{n-K}$.

Once the prime implicants of a given function have been determined it is still necessary to determine the simplest representations of the function. In order to do this, we first construct a new model. For each function f in $\widetilde{B}_n$, let the f-set be that set of integers i such that 1 appears in the i^{th} row of the f-matrix. Now the collection of these sets is simply the power set of $\{1, 2, \ldots\ldots 2^n\}$. This power set forms, of course, a Boolean algebra, and the correspondence $f \longrightarrow$ f-set, $f \in \widetilde{B}_n$, is an isomorphism. It is clear that the following relations hold:

$$f \cdot g \leftrightarrow \text{f-set} \cap \text{g-set}$$

$$f + g \leftrightarrow \text{f-set} \cup \text{g-set}$$

$$f' \leftrightarrow \text{The complement of the f-set with respect to } \{1, 2, \ldots 2^n\}.$$

$$f \text{ implies } g \leftrightarrow (\text{f-set}) \subset (\text{g-set})$$

Now once the prime implicants of the function f have been determined the simplification problem is essentially that of determining the most efficient covering of the f-set with those sets which correspond to the prime implicants.

Example: Again consider the function f of Example 1:

$$f = X_1X_2' + X_1'X_2 + X_2X_3' + X_2'X_3$$

We first compute the f-set:

	X_1	X_2	X_3	f
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	1
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	0

In the new model, we have $f = \{2, 3, 4, 5, 6, 7\}$

The prime implicants of f have been determined. In the new model they are:

$$X_1X_2' = \{5, 6\}$$
$$X_1'X_2 = \{3, 4\}$$
$$X_1X_3' = \{5, 7\}$$
$$X_1'X_3 = \{2, 4\}$$
$$X_2X_3' = \{3, 7\}$$
$$X_2'X_3 = \{2, 6\}$$

Applying the technique of page 67, we find that:

1) No union of two of the above sets covers f.

2) $\{2, 3, 4, 5, 6, 7\} = \{2, 6\} \cup \{3, 4\} \cup \{5, 7\}$

$$\{2, 3, 4, 5, 6, 7\} = \{2, 4\} \cup \{3, 7\} \cup \{5, 6\}$$

We see that these are the only two ways in which $\{2, 3, 4, 5, 6, 7\}$ can be covered by a union of three of the above sets.

Hence there are exactly two simplest representations of f. They are:

$$X_2'X_3 + X_1'X_2 + X_1X_3' \quad \text{and}$$

$$X_1'X_3 + X_2X_3' + X_1X_2' \quad .$$

BIBLIOGRAPHY

1. Birkhoff, Garrett and Saunders MacLane. A survey of modern algebra. Rev. ed. New York, Macmillan, 1953. 472 p.

2. Brender, David M. The logical procedures needed for finding the minimals of a Boolean function on a digital computer. In: Summer Institute for Symbolic Logic Cornell University 1957. 2d ed. Princeton, N.J., Communications Research Division Institute for Defense Analysis, 1960. 210-212.

3. Hohn Franz E. Applied Boolean algebra, an elementary introduction. New York, Macmillan, 1960. 139 p.

4. Newman, M. H. A. Elements of the topology of plane sets of points. London, Cambridge University Press, 1939. 221 p.

5. Quine, W. V. The problem of simplifying truth functions. American Mathematical Monthly. 59:521-531. 1952.

6. Shannon, Claude E. A symbolic analysis of relay and switching circuits. American Institute of Electrical Engineers. 57:713-723. 1938.

APPENDIX

APPENDIX

In the set-theoretic model any function f of $\tilde{B}_n$, with the exception of the zero element can be considered as a sequence of natural numbers. This sequence is obtained by ordering the elements of the f-set by its natural ordering. A literal then corresponds to a sequence of exactly 2^{n-1} elements. The m^{th} term of the sequence corresponding to the literal y_j can be determined as follows: Let a be the row number of the first row of the y_j-matrix in which 1 appears. The y_j-matrix is partitioned into 2^j, 2^{n-j} by 1 sub matrices, where each sub matrix has elements either all zero, or all 1. Consider those sub matrices in which the elements are all 1. The first row of the first such sub matrix has row number a with respect to the y_j-matrix. The first row of the second sub matrix, whose elements are all 1, has row number $a + 2^{n-j+1}$ with respect to the y_j matrix. The row numbers, with respect to the y_j-matrix, of the first rows of the 2^{j-1} sub matrices in which only 1 appears, form an arithmetic progression with first term a and common difference 2^{n-j+1}. Hence the first row of the r^{th} sub matrix in which only 1 appears has row number $a + (r-1)2^{n-j+1}$ with respect to the y_j-matrix. Now given the integer m such that $1 \leq m \leq 2^{n-1}$, write $m = m_1 \cdot 2^{n-j} + m_2$, where $0 \leq m_2 < 2^{n-j}$. The integers m_1 and m_2 are, of course, uniquely determined by the division algorithm.

Case 1. $m_2 = 0$. In this event, the m_{th} term of the y_j-sequence is the row number, with respect to the y_j-matrix, of the last row of the sub matrix Q, where Q is the m_1st matrix of those sub matrices in which only 1 appears. Hence the m^{th} term of the y_j sequence is

$$a + (m_1 - 1)\, 2^{n-j+1} + 2^{n-j} - 1.$$

Denote the m^{th} term of the y_j sequence by $y_j(m)$. Now in this case, $m = m_1 \cdot 2^{n-j}$. Therefore,

$$y_j(m) = a + 2(m_1 \cdot 2^{n-j}) - 2 \cdot 2^{n-j} + 2^{n-j} - 1 =$$
$$a + 2m - 2^{n-j} - 1.$$

If y_j is X_j, then $a = 2^{n-j} + 1$.

If y_j is X_j', $a = 1$.

Hence if m is a multiple of 2^{n-j}, the m^{th} term of the X_j-sequence is $2m$, and the m^{th} term of the X_j'-sequence is $2m - 2^{n-j}$.

Case 2. $m = m_1 \cdot 2^{n-j} + m_2$, where $m_2 \neq 0$. In this event, the m^{th} term of the y_j sequence is the row number, with respect to the y_j-matrix, of the $m_2{}^{od}$ row of the matrix Q, where Q is the $(m_1 + 1)^{th}$ matrix of those sub matrices in which only 1 appears. Hence

$$y_j(m) = a + (m_1) \cdot 2^{n-j+1} + m_2 - 1$$

Therefore, if y_j is X_j,
$X_j(m) = 2^{n-j} + 1 + 2 \cdot 2m_1 \cdot 2^{n-j} + m_2 - 1 =$
$2^{n-j}(1 + 2m_1) + m_2$, and $X_j'(m) = m_1 \cdot 2^{n-j+1} + m_2$.

We summarize these results as follows:

The m^{th} term of the sequence corresponding to X_j of $\tilde{B}_n$ is given by: $X_j(m) = 2m$ if m is a multiple of 2^{n-j}. $X_j(m) = 2^{n-j}(1 + 2m_1) + m_2$, if $m = m_1 \cdot 2^{n-j} + m_2$, where $0 < m_2 < 2^{n-j}$.

The m^{th} term of the sequence corresponding to X_j' of $\tilde{B}_n$ is given by:

$X_j'(m) = 2m - 2^{n-j}$ if m is a multiple of 2^{n-j}.

$X_j'(m) = m_1 \cdot 2^{n-j+1} + m_2$ if $0 < m_2 < 2^{n-j}$.