A MECHANIZATION OF QUINE'S CANONICAL FORM

by

ALBERT LEE LAXDAL

A THESIS

submitted to

OREGON STATE COLLEGE

in partial fulfillment of
the requirements for the
degree of

MASTER OF SCIENCE

June 1959

APPROVED:

Redacted for Privacy

Associate Professor of Mathematics

In Charge of Major

Redacted for Privacy

Chairman of Department of Mathematics

Redacted for Privacy

Chairman of School Graduate Committee

Redacted for Privacy

Dean of Graduate School

Date thesis is presented May 13, 1959

Typed by Rose Amos

TABLE OF CONTENTS

A MECHANIZATION OF QUINE'S CANONICAL FORM

CHAPTER I

INTRODUCTION

In recent years there has been much attention brought to bear on the problem of representation of Boolean polynomials. Many methods have been devised for reducing a representation of a Boolean polynomial to a minimal representation. The problem of determining a minimal representation of a polynomial is complicated by the fact that a minimal representation is not unique.

The ambiguity of representation, coupled with the ever present specter of human error gives rise to the following problem. "Given two representations of a Boolean polynomial, how can one determine whether or not they are equivalent."

The purpose of this paper is to answer that problem and show a method whereby an electronic computer can be made to do all of the computational work. The first section of the paper will deal with notions fundamental to the understanding of the problem. The second section will present two solutions. The first solution has been a tool of Boolean Algebra since its conception. The second solution is comparatively new, having been presented by Quine (2) in 1955. Also, in this section, there will be an algebraic-induction proof of the fact that the second solution is, indeed, a solution.

The third section concerns itself with the mechanization of the second solution, while the fourth section contains examples of the mechanization process.

CHAPTER II

SOME FUNDAMENTAL PROPERTIES OF BOOLEAN VARIABLES

Before proceeding to the problem at hand, it will be necessary to definè some basic terms.

Definition 1.

A variable a_i is a Boolean variable iff it may assume only the Boolean values 0 or 1.

Definition 2.

Let S be a set of Boolean variables and let a_j be an element of S. We say that a_j is an independent Boolean variable, relative to S, iff a_j may assume 0 and 1 values independently of the values assumed by any other elements of S.

Definition 3.

Let a_j be an independent Boolean variable. We shall introduce the Boolean variable $\overline{a}_j$, such that if a_j assumes the value 1 then $\overline{a}_j$ assumes the value 0 and vice versa. The variable $\overline{a}_j$ is called the dual of a_j.

Throughout the remainder of this paper we shall use the term "independent variable" to denote the term "independent Boolean variable".

Definition 4.

A Boolean monomial is an independent variable or its dual, or a product of independent variables or of a set of independent variables and a set of duals of other independent variables.

Definition 5.

A representation of a Boolean polynomial is the Boolean constant 0, or a sum of Boolean monomials.

Definition 6.

Two polynomials in the same independent Boolean variables are equal if they have the same value for every set of values of the independent variables.

It is here that an interesting logical point arises. It is not difficult to establish that for every representation P of a Boolean polynomial in n independent variables there exists an equivalence class (P) of representations equal to P. To avoid ambiguity in any discussion concerning Boolean polynomials we shall adopt the following convention. When we speak of a polynomial, we mean an equivalence class of equal Boolean variables. When we speak of a representation of a polynomial, we mean an arbitrary element of the equivalence class.

Notation:

The product of the two Boolean variables A and B is represented by the symbol AB.

The sum of two Boolean variables A and B is represented by the symbol A + B.

Definition 7.

If A, B, and C are three Boolean variables such that

$$A = BC$$

then A is less than or equal to B ($A \leq B$, $B \geq A$) and A is less than or equal to C ($A \leq C$, $C \geq A$).

$BC \to B$
$BC \to C$

Definition 8.

If A, B, C, and D are four Boolean variables and a is an independent Boolean variable such that

$$A = Ca$$

and

$$B = D\bar{a}$$

then the variable CD is called the consensus of A and B.

Definition 9.

Multiplication and addition of Boolean variables are defined by the following tables.

+	1	0
1	1	0
0	0	0

+

·	1	0
1	1	1
0	1	0.

Let A, B, and C be any Boolean variables. On the basis of the foregoing definitions the validity of the following identities is easily shown.

$$A + A = A \qquad AA = A$$

$$A + B = B + A \qquad AB = BA$$

$$A + (B+C) = (A+B) + C \qquad A(BC) = (AB)C$$

$$1 + A = 1 \qquad 1 \cdot A = A$$

$$0 + A = A \qquad 0 \cdot A = 0$$

$$(A+BC) = (A+B)(A+C)$$

Further, if a is an independent Boolean variable, we have,

$$a + \bar{a} = 1 \qquad a\bar{a} = 0$$

Due to the simplicity of the proofs we will omit them.

We now proceed to prove some lemmas that will be of use in our investigation.

Lemma 1.1.

The relation $(\geq)$ given in Definition 7 partially orders any set of Boolean variables.

Proof.

We have

$$A = AA,$$

hence,

$$A \geq A.$$

Assume that

$$A \geq B \qquad (AC = B)$$

and

$$B \geq A \qquad (BD = A).$$

Then,

$$AB = BDB = BBD = BD = A$$

and

$$AB = AAC = AC = B$$

but

$$A = AB = B.$$

Hence

$$A = B.$$

Let A, B, and C be three Boolean variables such that

$$A \geq B \qquad (AE = B)$$

and

$$B \geq C \qquad (BD = C)$$

This gives us

$$AED = C$$

and

$$A \geq C.$$

Lemma 1.2.

Let P be a representation of a Boolean polynomial, and A and B be any Boolean mononomials. Then

$$P + A + AB = P + A.$$

Proof.

$$P + A + AB = P + A(1+B)$$

and

$$P + A(1+B) = P + A.$$

Thus

$$P + A + AB = P + A.$$

We see from the above lemma, that if A and C are two monomials of a representation such that $A \geq C$ then the monomial C may be deleted from the representation without altering the polynomial.

Lemma 1.3.

Let P be a representation of a Boolean polynomial and let αa and $\beta\bar{a}$ be two Boolean monomials, a being an independent variable. Then

$$P + \alpha a + \beta\bar{a} = P + \alpha a + \beta\bar{a} + \alpha\beta.$$

Proof.

$$P + \alpha a + \beta\bar{a} + \alpha\beta = P + \alpha a + \beta\bar{a} + (a+\bar{a})\alpha\beta$$

and

$$P + \alpha a + \beta\bar{a} + (a+\bar{a})\alpha\beta = P + \alpha a + \beta\bar{a} + a\alpha\beta + \bar{a}\alpha\beta.$$

But

$$\alpha a \geq a\alpha\beta$$

and

$$\beta\bar{a} \geq \bar{a}\alpha\beta.$$

On the basis of lemma 1.2 we may delete the two monomials $a\alpha\beta$ and $\bar{a}\alpha\beta$ from the representation $P + \alpha a + \beta\bar{a} + a\alpha\beta + \bar{a}\alpha\beta$ obtaining an equivalent representation $P + \alpha a + \beta\bar{a}$; hence,

$$P + \alpha a + \beta\bar{a} = P + \alpha a + \beta\bar{a} + \alpha\beta.$$

On the basis of this lemma, if αa and $\beta\bar{a}$ are monomials of a representation of a Boolean polynomial (P), we may add their consensus $\alpha\beta$ without changing the polynomial.

This concludes our discussion of elementary properties of Boolean functions. In the next section we consider two solutions to the problem of representation of Boolean polynomials.

CHAPTER III

BOOLEAN CANONICAL FORMS

We now consider the problem of determining when two representations are elements of the same equivalence class. What we would like to find is a simple algorithm which when applied to all elements of an equivalence class gives the same element of the equivalence class. The representation common to all elements of the equivalence class we call a Boolean canonical form.

A well known Boolean canonical form is derived from the truth table of the representations from a class (P). The form is defined as follows. Let $f(a_1,a_2,\ldots,a_n)$ be a representation of a Boolean polynomial in n independent Boolean variables. Then

$$f(a_1,a_2,\ldots,a_n) = \sum a_1^\eta \, a_2^\eta \ldots a_n^\eta \, f(e_1,e_2,\ldots,e_n).$$

where $a_i^\eta = a_i$ if $e_i = 1$ and $a_i^\eta = \overline{a}_j$ if $e_i = 0$ and the sum is over all of the 2^n possible combinations of 0 and 1 in the ordered set $e_1,e_2,\ldots,e_n$.

That this is a canonical form is proved in standard texts, e.g. Rosenbloom (4). It is instructive to consider an example. Let f(a,b) be a + b, then

$$f(a,b) = abf(1,1) + \overline{a}bf(0,1) + a\overline{b}f(1,0) + \overline{a}\overline{b}f(0,0)$$

and we have,

$$a + b = ab + \bar{a}b + a\bar{b}.$$

While the above method assures that a method of comparison is available, it grows out of hand when the number of variables is large since one must always make 2^n truth evaluations. For very simple representations the canonical representations may be complex.

For example, consider the representation $P = a + b + c + d$. The canonical form of P is the following

$$\begin{aligned} P = {} & abcd + \bar{a}bcd + a\bar{b}cd + ab\bar{c}d + abc\bar{d} \\ & + \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} \\ & + ab\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}b\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c}\bar{d}. \end{aligned}$$

It would not be easy to look at the canonical form of P and see that P = 1 if any one of the variables a, b, c or d assumes the value 1; however, this is very evident when one looks at the original representation.

It is for reasons such as these that the writer feels a second form due to Quine (2) or Samson and Mills (5) is superior to the above form.

Throughout this section the following conventions will be honored:

(i) The first 13 capital English letters will be used to denote Boolean monomials.

(ii) Lower case Greek letters will be used to denote sums of monomials or the Boolean constant 0.

(iii) Lower case English letters will denote independent variables.

There are three conditions Quine requires a representation of a Boolean polynomial to satisfy in order that it may be said to be in canonical form.

Condition 1. If A and B are two monomials of a representation P such that A $\geq$ B, then the monomial B is to be deleted from the representation.

Condition 2. If Aa and B$\bar{a}$ are two monomials of a representation P and AB satisfies the following two conditions:

(i) the monomial A does not contain as a factor the <u>dual</u> of any independent variable contained as a factor in B,

(ii) the product AB is not less than or equal to any monomial C of P, then the monomial AB is to be added to the representation.

Condition 3. If the monomials a and $\bar{a}$ occur in a representation, the polynomial is the Boolean constant 1.

On the basis of lemmas 1.2, 1.3 and our table of Boolean identities we see that the three conditions transform a representation P into an equal representation.

Let us now consider some of the properties of a representation P that satisfies the above conditions. Let P be a non zero representation in n independent variables that satisfies Quine's three conditions. By factoring we may obtain the following Boolean equivalent of P,

$$P = a_n\psi_1 + \bar{a}_n\psi_2 + \psi_3,$$

in which ψ_1, ψ_2 and ψ_3 are representations of polynomials in n-1 variables and not both of ψ_1 and ψ_2 are the zero representation; however, ψ_3 may be the zero representation.

Lemma 2.1.

Considered apart from P, ψ_3 satisfies both conditions.

Proof.

Assume that ψ_3 contains the monomials A and B and $A \geq B$. This implies that P contains A and B contrary to the assumption that P satisfies Condition 1; thus, ψ_3 satisfies Condition 1.

Assume that ψ_3 contains the monomials aA and $\bar{a}$B and the consensus AB is not contained in ψ_3, nor is AB less than or equal to any monomial C of ψ_3. This implies that P is in violation of Condition 2, unless some monomial of $a_n\psi_1$ or $\bar{a}_n\psi_2$ is greater than or equal to AB. No monomial of ψ_3 contains a_n or $\bar{a}_n$ as a factor; hence, every monomial of $a_n\psi_1$ or $\bar{a}_n\psi_2$ contains as a factor at least one independent variable foreign to every monomial of ψ_3. It follows that every monomial of $a_n\psi_1$ or $\bar{a}_n\psi_2$ contains as a factor at least one independent variable foreign to any consensus of monomials of ψ_3, since the process of forming the consensus of two monomials adds no new independent variables. We have, then, that no monomial of $a_n\psi_1$ or $\bar{a}_n\psi_2$ may be greater than or equal to any consensus of monomials of ψ_3; thus, our initial assumption leads to a contradiction and ψ_3 satisfies Condition 2.

If ψ_3 violates Condition 3, then it is immediate that P also violates Condition 3.

It is interesting to note that ψ_3 is the only one of ψ_1, ψ_2, and ψ_3 that must satisfy all three conditions. The following example shows this clearly. For a_n, ψ_1, ψ_2, and ψ_3 choose the following:

$$a_n = a_5,$$
$$\psi_1 = a_2a_3 + \bar{a}_3a_4,$$
$$\psi_2 = a_2\bar{a}_4 + a_4a_1,$$

and

$$\psi_3 = a_1a_2 + a_2a_4,$$

then

$$P = a_5(a_2a_3+\bar{a}_3a_4) + \bar{a}_5(a_2\bar{a}_4+a_4a_1) + a_1a_2 + a_2a_4.$$

Lemma 2.2.

There is no monomial in ψ_3 that is greater than or equal to a monomial in ψ_i, $(i = 1,2)$.

Proof.

Assume that there is a monomial A in Ψ_3 and a monomial B in Ψ_i such that B = AC. Then there is a monomial ACa_n in $\Psi_1 a_n$ or a monomial $aC\bar{a}_n$ in $\Psi_2 \bar{a}_n$. This implies that there is a deletion to be made contradicting the fact that P satisfies Condition 1.

Lemma 2.3.

There is no consensus to be made between the monomials of Ψ_i and Ψ_3, $(i = 1,2)$.

Proof.

If we assume that for all monomials A_j in Ψ_3 there are no monomials in Ψ_i such that a consensus is to be made, the lemma is a trivial statement. Let us assume, for simplicity's sake, that there is a monomial Ab in Ψ_1 and a monomial $C\bar{b}$ in Ψ_3 such that Ab satisfies (i) and (ii) of Condition 2, and AC is not a monomial of Ψ_1. This implies that $a_n AC$ is not a monomial of $a_n \Psi_1$, contradicting the supposition that P satisfied Condition 2.

On the basis of the preceding lemmas we may deduce some properties of $\Psi_i + \Psi_3$, $(i = 1,2)$. We assume that neither of Ψ_i nor Ψ_3 contains as monomials both of the variables a_j and $\bar{a}_j$. Let us consider what operations we should apply to $\Psi_i + \Psi_3$ in order that it should satisfy Conditions 1 and 2. By lemmas 2.2 and 2.3 we see that the only possibility is the deletion of monomials in Ψ_3 that are less than or equal to monomials of Ψ_i. Let

$\psi_1 + R_1$ be the representation of $\psi_1 + \psi_3$ that satisfies Quine's conditions. Since the operation of deleting monomials does not create new monomials, we can say that any monomial of R_1 is a monomial of ψ_3.

Consider the product $(\psi_1+\psi_3)(\psi_2+\psi_3)$.

$$(\psi_1+\psi_3)(\psi_2+\psi_3) = (\psi_1\psi_2+\psi_1\psi_3+\psi_2\psi_3+\psi_3)$$

From lemma 1.2 we obtain

$$\psi_1\psi_2 + \psi_1\psi_3 + \psi_2\psi_3 + \psi_3 = \psi_1\psi_2 + \psi_3.$$

Since P satisfies Condition 2, we must have all of the monomials of the product $\psi_1\psi_2$ that are not zero or less than or equal to monomials of ψ_3 in ψ_3; hence,

$$(\psi_1+\psi_3)(\psi_2+\psi_3) = \psi_3.$$

Theorem 1.

If there are two representations P_1 and P_2 of a Boolean polynomial P such that both P_1 and P_2 satisfy Quine's conditions, then P_1 is identical with P_2 save for the order of the monomials.

The proof is by induction on the number of independent variables in the polynomial.

The case for $n = 1$ is quickly disposed of as any representation in one independent variable that satisfies Quine's conditions must be one of the following:

a,

$\bar{a}$,

1,

0,

and if two representations that satisfy the conditions are equal, they are identical.

Assume that the proposition to be proved is true for all polynomials in $k - 1$ or fewer variables and consider the case $n = k$. Assume that there are two representations P and Q of a polynomial and both representations satisfy Quine's conditions.

By factoring we obtain the fact that P is

$$a_k\alpha_1 + \bar{a}_k\alpha_2 + \alpha_3$$

and Q is

$$a_k\beta_1 + \bar{a}_k\beta_2 + \beta_3.$$

We note that

$$\alpha_1 + \alpha_3 = f(a_1, a_2, \ldots, a_{k-1}, 1) = \beta_1 + \beta_3$$

and

$$\alpha_2 + \alpha_3 = f(a_1, a_2, \ldots, a_{k-1}, 0) = \beta_2 + \beta_3.$$

By the induction hypothesis, since α_3 and β_3 satisfy Quine's conditions and each is a representation of

$$f(a_1, a_2, \ldots, a_{k-1}, 1)f(a_1, a_2, \ldots, a_{k-1}, 0),$$

α_3 and β_3 are identical save for the order of their monomials.

Let us operate on $(\alpha_1+\alpha_3)$, $(\alpha_2+\alpha_3)$, $(\beta_1+\beta_3)$, and $(\beta_2+\beta_3)$ in order that they satisfy Quine's conditions.

We obtain

$$\alpha_1 + \alpha_3 = \alpha_1 + R_1,$$
$$\alpha_2 + \alpha_3 = \alpha_2 + R_2,$$
$$\beta_1 + \beta_3 = \beta_1 + E_1$$

and

$$\beta_2 + \beta_3 = \beta_2 + E_2,$$

since

$$\alpha_i + \alpha_3 = \beta_i + \beta_3,$$

we have

$$\alpha_i + R_i = \beta_i + E_i.$$

By the induction hypothesis $\alpha_i + R_i$ is identical with $\beta_i + E_i$ since both are functions of $k - 1$ or fewer variables.

Let us assume that α_i is not identical with β_i for at least one i $(i = 1,2)$. This implies that one of them must contain a monomial not contained in the other. Assume that α_i contains a monomial A that is not contained in β_i. Since we have an identity between $\alpha_i + R_i$ and $\beta_i + E_i$, A must be contained in E_i. This implies that A is contained in β_3 and since β_3 is identical with α_3 , A must be contained in α_3. But, by lemma 2.2, α_1 and α_3 cannot share a monomial; hence a contradiction arises from the assumption that α_1 contains a monomial not contained in β_1. By symmetry, it follows that β_1 cannot contain a monomial that is not contained in α_1; hence, α_1 is identical with β_1 and the theorem is proved. The case of the zero and one representations are trivial in any number of variables.

We have three operations such that when they are applied to any two representations of a Boolean polynomial they give rise to the same representation. One may wonder what methods can be used to insure that the canonical form is obtained in a finite number of steps. We now exhibit an algorithm that will always obtain the canonical form.

Let us establish the convention of calling the process of deleting all monomials that are less than or equal to other monomials of a representation a deletion iteration and the process of forming every possible consensus a consensus iteration.

Theorem 2.

The following algorithm will always insure that a polynomial satisfies Quine's conditions. Begin with a deletion iteration and then follow with a consensus iteration which is followed by a check to see whether or not the monomials a_j and $\bar{a}_j$ both appear. If both appear, write the Boolean constant 1 as a canonical representation.

If a_j and $\bar{a}_j$ are not both elements of the representation which has been obtained by the previous iterations, follow with a deletion iteration.

Proof.

Assume that the above process has been carried out. Consider Condition 3. The check to determine whether or not the representation contained both a_j and $\bar{a}_j$ was made at the time that the representation contained the maximum number of independent variables as monomials; hence, if the representation did not contain them then, it will not contain them at all. Thus, Condition 3 is satisfied.

Assume now, that the monomials a_j and $\bar{a}_j$ do not both occur. Then it is immediate that Condition 1 is satisfied since the last iteration was a deletion iteration. Suppose now that there are monomials Aa and B$\bar{a}$ such that the consensus AB is to be formed. The monomial AB cannot be formed if there is a monomial D of the representation such that $D \geq A$. In the first section, we noted that a deletion iteration does not create new terms; thus, Aa, and B$\bar{a}$ were present during the consensus iteration. Since the term AB was not created during the consensus iteration, it is evident that there was a monomial E of the representation such that $E \geq AB$.

One of two cases may arise.

Case 1. E remains after the deletion iteration.

Case 2. E was deleted during the second deletion iteration.

Consider Case 1. Since E remains, we have $E \geq AB$ and AB cannot be formed.

In Case 2, since E was deleted, we know that there must have been a monomial F of the representation such that $F \geq E$. Now F may have been deleted by a monomial G and G may in turn be deleted, etc. If this process continues, we obtain an ascending chain of monomials

$$M \geq L \geq \ldots F \geq E \ldots \geq AB.$$

Since there can be only a finite number of monomials in a representation, the chain is finite and we obtain a greatest element M, M may be F, that remains in the representation. Since $M \geq AB$, the product AB cannot be formed; thus, the representation satisfies Condition 2.

Let us now consider a variation of the above method. Instead of checking, during the consensus iteration to see if sub condition (ii) of Condition 2 is satisified, form the monomial AB any time the monomials Aa and $B\bar{a}$ occur in the representation and the monomial AB satisfies sub condition (i) of Condition 2. Let us call a consensus that violates sub condition (ii) of Condition 2 a forbidden concensus.

Lemma 2.4.

No forbidden consensus remains after the second deletion iteration.

Let A be a forbidden consensus, then $A \leq B$ for some monomial B of the representation. On the basis of the proof of the preceding theorem at least one monomial F such that $F \geq A$ remains at the end of the second deletion iteration and A is deleted.

Lemma 2.5.

No monomial that is the consensus of two monomials such that at least one of them is a forbidden consensus remains after the second deletion iteration.

Proof.

Let BC be a monomial that is the consensus of two monomials aB and $\bar{a}C$ and aB is a forbidden consensus. We need only show that AB is a forbidden consensus.

Since aB is a forbidden consensus, we know that there exists a monomial D, of the representation, such that $D \geq aB$. One of two cases may occur.

Case 1. D contains a.

Case 2. D does not contain a.

Case 1. If D contains a, then $D = Ea$ and the monomial EC is to be formed.

Now

$$DaB = aB,$$

and

$$EB = B$$

thus,

$$ECBC = ECCB = ECB = CEB = CB,$$

hence

$$EC \geq BC,$$

and BC is a forbidden consensus.

Case 2. Since D does not contain a, we have

$$DB = B$$

and

$$DBC = BC$$

hence

$$D \geq BC$$

and BC is a forbidden consensus.

Lemma 2.6.

The monomial a_j is not a forbidden consensus unless a_j is already a monomial of the representation or the representation is the Boolean constant 1.

Proof.

The only Boolean variables that are greater than or equal to the independent variable a_j are the variable a_j and the Boolean constant 1.

Lemma 2.7.

No monomial A is less than or equal to a forbidden consensus B unless it is less than or equal to a monomial that is not a forbidden consensus.

Proof.

Assume that $A \leq B$ and B is a forbidden consensus. Since B is a forbidden consensus, there exists a monomial C of the representation such $B \leq C$; hence $A \leq C$.

We see that allowing forbidden consensus to be formed does not alter the end result of the second deletion iteration.

We have shown the existence of a finite algorithm for obtaining the canonical form of a Boolean polynomial. Such a method is amenable to machine computation. In the next chapter we will consider the mechanization of this algorithm.

CHAPTER IV

THE MECHANIZATION PROCESS

Essentially, there are three problems that one must consider in the programming of the algorithm. One must first find a method of representing a Boolean monomial in a machine. Having done this, one must then determine a method for finding out whether or not a deletion is to be made. Finally, one must determine when a consensus is to be made and how to make it.

The methods discussed in this section apply to a machine with the following characteristics:

1. A word consists of 2n bits.

2. The machine can take the logical sum of two words. The logical sum of two words A and B is a word D that contains a 1 bit in bit position k iff A or B contains a 1 bit in bit position k.

3. The machine can take the extract of two words. The and extract of two words A and B is a word D that contains a 1 bit in position k iff both A and B contain a 1 bit in bit position k.

4. The machine can shift a word until a 1 bit occurs in a preassigned bit position and count the number of bit shifts necessary to bring the 1 bit to the position.

5. The machine can complement a word. The complement of a word A is a word B that contains a 1 bit

in the k^{th} bit position iff A contains a 0 bit in the k^{th} bit position and vice versa.

Throughout the rest of this section we will use the terms "k^{th} position" or "position k" to denote the term "k^{th} bit position".

Let us label our independent variables according to their input order i.e., a_i is the i^{th} independent variable to be entered into the machine. We now introduce two auxiliary words, A and $\overline{A}$. A is a word that contains a 1 bit in the first and $(n + 1)^{st}$ positions and 0's in every other position, while $\overline{A}$ is a word that has a 1 bit in the $(n + 1)^{st}$ position and 0's in every other position.

n: number of indept variables

The independent variable a_i is represented by the word $[a_i]$ that has a 1 bit in the i^{th} and $(n + i)^{th}$ positions and 0's in every other position. The variable $\bar{a}_j$ has a 1 bit in the $(n + j)^{th}$ position and 0's in every other position.

Let us assume that we want to represent the monomial $a_i\bar{a}_j$. We first enter the variable a_i into the machine. The machine determines that it is the i^{th} independent variable to be entered and forms the word $[a_i]$ by calling out A and shifting it i - 1 places to the left. The word $[a_i]$ is then stored. When $\bar{a}_j$ is entered into the machine, the machine determines that it is the j^{th} variable to be entered. The word $[\bar{a}_j]$ is then formed by calling out $\overline{A}$ and shifting it j - 1 places to the left. We now take the logical sum of $[a_i]$ and $[a_j]$ obtaining $[a_i\bar{a}_j]$. The word $[a_i\bar{a}_j]$ has a 1 bit in the $(n + i)^{th}$, $(n + j)^{th}$, and i^{th} positions and 0's in every other position. This method enables us to represent in one word a monomial in n independent variables. what about $a_1\bar{a}_1$

For an example, let us assume we have a machine that has a word length of 6 bits. The bit configuration of the representation of the monomial $a_1\bar{a}_2a_3$ is the following:

$$[a_1\bar{a}_2a_3] = 1\ 1\ 1\ 1\ 0\ 1.$$

A representation of a polynomial is stored in the machine by the process of serial storage of monomial representations, the 0 and 1 representations being separately indicated.

Assume that a representation P is stored in the machine. Let α be a monomial of P. We want to find out whether α is greater than or equal to any other monomial β of P.

Let us denote the result of extracting $[\alpha]$ with $[\beta]$ by $E(\alpha,\beta)$.

Lemma 3.1.

A necessary condition for α to be greater than or equal to β is

$$E(\alpha,\beta) = [\alpha].$$

Proof.

Assume $E(\alpha,\beta) \neq [\alpha]$, then in some position, say position j, $[\alpha]$ contains a 1 bit, while $[\beta]$ does not contain a 1 bit in position k. This implies that α contains as a factor a variable not contained in β; hence $\alpha \ngeq \beta$.

This condition is not a sufficient condition as the following example shows.

Let

$$\alpha = a_1\bar{a}_2$$

$$\beta = a_1a_2a_3$$

then

$$[\alpha] = 0\ 1\ 1\ 0\ 0\ 1$$

$$[\beta] = 1\ 1\ 1\ 1\ 1\ 1$$

and

$$E(\alpha,\beta) = 0\ 1\ 1\ 0\ 0\ 1 = [\alpha]$$

but α is not greater than or equal to β.

Let $\overline{H}_\alpha$ denote the word that consists of the right half word of $[\alpha]$ and has 0's in its left half. H_α is the word that has for its right half word the left half word of $[\alpha]$ and has 0's in its left half.

Theorem 3.

Either one of the following two conditions:

(i) $E(\alpha,\beta) = [\alpha]$,

(ii) $E(H_\alpha,\overline{H}_\beta) = E(\overline{H}_\alpha,H_\beta)$

are necessary for the inequality

$$\alpha \geq \beta$$

to hold and the simultaneous truth of both of them is sufficient for the inequality to hold.

Proof.

Since we have already shown the necessity of (i), consider (ii). Assume that $E(H_\alpha,\bar{H}_\beta) \neq E(\bar{H}_\alpha,H_\beta)$ and let $[\Psi] = E(H_\alpha,\bar{H}_\beta)$ and $[\gamma] = E(\bar{H}_\alpha,H_\beta)$. If $[\Psi] \neq [\gamma]$, then one of them, say $[\Psi]$, contains a 1 bit in the j^{th} position $(j \leq n)$ while $[\gamma]$ contains a 0 bit in position j. This implies that α contains, as a factor, a_j or $\bar{a}_j$ while β must contain a_j as a factor. From the 0 bit in $[\gamma]$, it follows that α contains $\bar{a}_j$ as a factor; hence, α contains factors foreign to β and α is not greater than or equal to β. Since the reasoning is symmetric, the same result may be obtained by choosing $[\gamma]$ to contain a 1 bit in the k^{th} position while $[\Psi]$ contains a 0 bit in the k^{th} position.

Assume now

$$E(\alpha,\beta) = [\alpha]$$

and

$$E(H_\alpha,\bar{H}_\beta) = E(\bar{H}_\alpha,H_\beta).$$

Since $E(\alpha,\beta) = [\alpha]$, we know that $[\beta]$ contains a 1 bit in every position that $[\alpha]$ contains a i bit. This reduces the proof to showing that α does not contain, as a factor, the dual of any independent variable appearing as a factor of β. Assume that α contains, as a factor, $\bar{a}_k$ and β

contains a_k as a factor. Then, H_α, H_β and $\bar{H}_\beta$ contain a 1 bit in position k while $\bar{H}_\alpha$ has a 0 in the k^{th} position; hence,

$$E(H_\alpha,\bar{H}_\beta) \neq E(\bar{H}_\alpha,H_\beta).$$

Since this contradicts our hypotheses, α may not contain, as a factor, the dual of any independent variable occurring as a factor of β and we have shown

$$\alpha \geq \beta.$$

On the basis of the preceding theorem the method for the deletion iteration is constructed. Let $[\alpha]$ and $[\beta]$ be two machine representations of Boolean monomials. Form $E(\alpha,\beta)$ and check to see whether or not $E(\alpha,\beta) = [\alpha]$. If $E(\alpha,\beta) = [\alpha]$, we check to see whether or not $E(H_\alpha,\bar{H}_\beta) = E(\bar{H}_\alpha,H_\beta)$. If $E(\bar{H}_\alpha,H_\beta) = E(H_\alpha,\bar{H}_\beta)$, then we delete β. If either one of the two equalities fails to hold, we proceed to check $[\alpha]$ with other monomial representations.

The process used in the formation of the consensus of two monomials uses the words H_α, $\bar{H}_\alpha$, H_β and $\bar{H}_\beta$, also.

Let $CE(\bar{H}_\alpha,\bar{H}_\beta)$ denote the complement of $E(\bar{H}_\alpha,\bar{H}_\beta)$.

Lemma 3.2.

$E[CE(\bar{H}_\alpha,\bar{H}_\beta) , E(H_\alpha,H_\beta)]$ will contain a 1 bit in position k iff both α and β contain $\bar{a}_k$ as a factor or one of them contains a_k, as a factor, while the other contains $\bar{a}_k$ as a factor.

Consider $E(\bar{H}_\alpha,\bar{H}_\beta)$, it will have a 0 in position k if both α and β contain $\bar{a}_k$, as a factor, or if one of them contains $\bar{a}_k$ as a factor while the other contains a_k as a factor. Thus, under the above conditions $CE(\bar{H}_\alpha,\bar{H}_\beta)$ will contain a 1 bit in position k.

If either of the above conditions are met, $E(H_\alpha,H_\beta)$ will contain a 1 bit in position k. Since both words of the extract contain a 1 bit in position k, we must have that their extract contains a 1 bit in the k^{th} position and we have shown the sufficiency of the conditions.

If the above conditions are not met, three cases may arise.

Case 1. Neither of α or β contains a_k or $\bar{a}_k$ as a factor.

Case 2. One of them contains either a_k or $\bar{a}_k$ as a factor but the other does not contain a_k or $\bar{a}_k$ as a factor.

Case 3. Both α and β contain a_k as a factor.

In Cases 1 and 2, $E(H_\alpha,H_\beta)$ must contain a 0 in position k; hence $E[CE(\bar{H}_\alpha,\bar{H}_\beta)\ ,\ E(H_\alpha,H_\beta)]$ has a 0 bit in position k.

In Case 3, $E(\bar{H}_\alpha,\bar{H}_\beta)$ must contain a 1 bit in position k, hence $CE(\bar{H}_\alpha,\bar{H}_\beta)$ must contain a 0 in the k^{th} position and $E[CE(\bar{H}_\alpha,\bar{H}_\beta)\ ,\ E(H_\alpha,H_\beta)]$ has a 0 in position k. And we have shown the necessity of the conditions.

Let $S(\overline{H}_\alpha,\overline{H}_\beta)$ denote the logical sum of $\overline{H}_\alpha$ and $\overline{H}_\beta$.

Lemma 3.3.

$E\left\{E[CE(\overline{H}_\alpha,\overline{H}_\beta)\ ,\ E(H_\alpha,H_\beta)]\ ,\ S(\overline{H}_\alpha,\overline{H}_\beta)\right\}$ will contain a 1 bit in position k iff α contains a_k as a factor while β contains $\bar{a}_k$ as a factor or α contains $\bar{a}_k$ as a factor and β contains a_k as a factor.

Proof.

If either one of the above conditions are satisfied then $S(\overline{H}_\alpha,\overline{H}_\beta)$ will contain a 1 bit in position k and by lemma 3.2, so will $E[CE(\overline{H}_\alpha,\overline{H}_\beta)\ ,\ E(H_\alpha,\overline{H}_\beta)]$. And we have shown the sufficiency of either one of the conditions.

Assume that neither one of the above conditions are satisfied. Then, four cases are possible, the first three being the three cases of lemma 3.2. The fourth case consists of both α and β containing $\bar{a}_k$ as a factor.

In the three cases of lemma 3.2
$E[CE(\overline{H}_\alpha,\overline{H}_\beta)\ ,\ E(H_\alpha,H_\beta)]$ does not contain a 1 bit in position k; thus, $E\left\{E[CE(\overline{H}_\alpha,\overline{H}_\beta)\ ,\ E(H_\alpha,H_\beta)]\ ,\ S(\alpha,\beta)\right\}$ cannot contain a 1 bit in the k^{th} position.

In case four, $S(\overline{H}_\alpha,\overline{H}_\beta)$ does not contain a 1 bit in position k; hence $E\left\{E[CE(\overline{H}_\alpha,\overline{H}_\beta)\ ,\ E(H_\alpha,H_\beta)]\ ,\ S(\alpha,\beta)\right\}$ does not contain a 1 bit in position k. And we have shown the sufficiency of the conditions.

Let $NE_{\alpha\beta}$ denote the number of 1 bits in
$E\left\{E[CE(\overline{H}_\alpha,\overline{H}_\beta)\ ,\ E(H_\alpha,H_\beta)]\ ,\ S(\alpha,\beta)\right\}$.

Theorem 4.

A consensus is to be made between α and β iff $NE = 1$.

Proof.

Assume $NE = 1$. From lemma 3.3, we know that α contains one and only one dual of an independent variable of β; thus, by Quine's second condition a consensus is to be made.

Assume $NE \neq 1$. Two cases arise.

Case 1. $NE = 0$.

Case 2. $NE \geq 2$.

In Case 1, α does not contain the dual of any variable of β; hence, no consensus is to be made.

In Case 2, α contains the dual of at least two independent variables of β; hence, by sub condition (i) of Quine's second condition no consensus is to be made.

The process used in the consensus iteration is the following. Let α and β be two monomials of the representation. Determine whether or not $NE_{\alpha\beta}$ is equal to one. If $NE_{\alpha\beta} \neq 1$, compare α with some other monomial of the representation. If $NE_{\alpha\beta} = 1$, shift $E\left\{E[CE(\overline{H}_\alpha,\overline{H}_\beta)\ ,\ E(H_\alpha,H_\beta)]\ ,\ S(\alpha,\beta)\right\}$ to the left until its 1 bit is in the $2n^{th}$ position and count the number of bit shifts necessary to do this. Let C be the number of shifts. Form the difference $2n - C$. Call out the auxiliary word A and shift it $2n - C$ places to the left. We then have a representation of the independent variable a_k that occurs in α and has its dual in β. Form $C[a_k]$ and store it. Take the word $S(\alpha,\beta)$ and form $E[S(\alpha,\beta)\ ,\ C(a_k)]$. This word is the consensus of α and β.

Since the two words that form a consensus γ are not unique, we must store each word that is a consensus so that we may know whether or not we have formed a consensus before, lest we exceed the storage capacity of a machine by storing it many times.

Lemma 3.4.

$E[S(\alpha,\beta)\ ,\ C(a_k)]$ will contain only 0's iff the consensus is between the two monomials a_k and $\bar{a}_k$.

Assume that the monomials $\bar{a}_k$ and a_k are being checked by the machine to determine whether or not a consensus is to be made. $NE_{a_k\bar{a}_k}$ will be one, and the 1 bit will be in the k^{th} position; thus, $C[a_k]$ will have a 0 in only the k^{th} position while $S(a_k,\bar{a}_k)$ will have a 1 bit in only the k^{th} position. It follows that $E[S(\alpha,\beta)\ ,\ C(a_k)]$ has 0's in every position.

Assume that the machine makes the consensus of two monomials α and β and at least one of α or β contains more than one independent variable as factors. The word $E[S(\alpha,\beta)\ ,\ C(a_k\}]$ must contain at least one 1 bit, since the process of forming $E[S(\alpha,\beta),\ C(a_k)]$ eliminates only one independent variable from $S(\alpha,\beta)$ and β or α contains at least two independent variables as factors.

Once the consensus iteration is finished, apply one more deletion iteration to complete the process.

CHAPTER V

EXAMPLES

In this section we will consider some examples that have been worked out by an electronic computer. For each polynomial whose canonical form has been determined there are three entries. The first entry is the representation whose canonical form is to be determined. The second entry is the result of one deletion iteration and one consensus iteration. In the second entry it will be noted that some of the monomials are underlined. The underlined monomials are the result of the deletion iteration. It is seen that to this representation one may add any combination of the monomials of the second entry that are not underlined and not alter the polynomial. The third entry is the canonical form of the polynomial.

Example 1.

$$P = abc + \bar{a}bc + a\bar{b}c + ab\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c}$$

Intermediate results:

$$P = a\bar{c} + a + \bar{b}c + a\bar{b} + c + \bar{a}b + \bar{a}c + b + b\bar{c} + ab + ac$$
$$+ bc + \underline{\bar{a}bc} + \underline{a\bar{b}c} + \underline{ab\bar{c}} + \underline{\bar{a}\bar{b}c} + \underline{abc} + \underline{\bar{a}b\bar{c}} + \underline{a\bar{b}\bar{c}}.$$

Canonical form:

$$a + b + c.$$

Example 2.

$$P = a\bar{b}c + abc + cd + \bar{c}f$$

Intermediate results:

$$P = df + abf + acf + a\bar{b}f + af + ac + \underline{a\bar{b}c} + \underline{abc} + \underline{cd} + \underline{\bar{c}f}$$

Canonical form:

$$P = df + af + ac + cd + \bar{c}f. \quad = \quad ac + cd + \bar{c}f$$

← i.e. not "shortest"

Example 3.

$$P = \bar{a}\bar{b}\bar{c}\bar{d} + agh + b\bar{g}j + \bar{j}kp$$

Intermediate results:

$$\begin{aligned}
&b\bar{g}kp + abhkp + a\bar{c}\bar{d}ghj + ab\bar{c}\bar{d}hj + b\bar{c}\bar{d}\bar{g}hj \\
&+ b\bar{c}\bar{d}\bar{g}hkp + ab\bar{c}\bar{d}hkp + a\bar{c}\bar{d}ghkp + a\bar{c}\bar{d}ghjkp \\
&+ ab\bar{c}\bar{d}hjkp + b\bar{c}\bar{d}\bar{g}hjkp + cbhj + \bar{a}\bar{c}\bar{d}\bar{g}kp \\
&+ \bar{a}\bar{c}\bar{d}\bar{g}hj + \bar{a}\bar{b}\bar{c}\bar{d}hkp + \bar{a}\bar{c}\bar{d}\bar{g}hkp + \bar{a}\bar{c}\bar{d}\bar{g}hjkp \\
&+ \bar{b}\bar{c}\bar{d}ghjkp + \bar{a}\bar{b}\bar{c}\bar{d}hjkp + \bar{b}\bar{c}\bar{d}ghkp + \bar{b}\bar{c}\bar{d}ghj \\
&+ \bar{a}\bar{b}\bar{c}\bar{d}hj + \bar{a}\bar{c}\bar{d}\bar{g}j + \bar{b}\bar{c}\bar{d}gh + \underline{\bar{a}\bar{b}\bar{c}\bar{d}} + \underline{agh} + \underline{b\bar{g}j} + \underline{\bar{j}kp}.
\end{aligned}$$

Canonical form:

$$\begin{aligned}
&b\bar{g}kp + abhkp + abhj + \bar{a}\bar{c}\bar{d}\bar{g}kp + \bar{a}\bar{c}\bar{d}\bar{g}j + \bar{b}\bar{c}\bar{d}gh \\
&+ \bar{a}\bar{b}\bar{c}\bar{d} + agh + b\bar{g}j + \bar{j}kp.
\end{aligned}$$

The intermediate results of the above example define a class of 301 equal representations.

The examples that we have seen so far have been rather well behaved. Some very pathological examples may be created by the following process. From a set of n variables ($n \geq 5$, the process is not worthwhile for $n < 5$) construct a representation by first forming all of the possible sets containing only $n - 1$ independent variables. There will be n of these. From each set form all of the possible monomials that may be formed by replacing two of the variables by their duals. There will be $(n-2) + (n-3) + \dots + (n-k+1)$ of them.

The total number of monomials formed will be

$$(n-1)\frac{(n-2)n}{2} = \frac{n^3 - 3n^2 + 2n}{2} .$$

No deletions can be made because the monomials are formed in such a way that either one contains as a factor the dual of an independent variable occurring as a factor of the other or each contains independent variables foreign to the other.

Any consensus between two monomials gives rise to another monomial of the representation. For, assume that α and β are two monomials of this representation.

Let

$$\alpha = a_1 a_2 \dots \bar{a}_i \dots \bar{a}_j \dots a_n$$

and

$$\beta = a_1 a_2 \dots a_i \dots \bar{a}_j \dots \bar{a}_k \dots a_p \qquad p \leq n.$$

For the consensus to be made we must have that a_k is foreign to α; thus the consensus is the same as the word formed by removing the factor a_i from α and adding the factor $\bar{a}_k$. The consensus contains $n - 1$ factors of which two are primed.

Example 4.

For n = 5 we obtain

$$\begin{aligned} P = {} & \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + \bar{a}bc\bar{d} + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + ab\bar{c}\bar{d} \\ & \bar{a}\bar{b}ce + \bar{a}b\bar{c}e + \bar{a}bc\bar{e} + a\bar{b}c\bar{e} + a\bar{b}\bar{c}e + ab\bar{c}\bar{e} \\ & \bar{a}\bar{b}de + \bar{a}b\bar{d}e + \bar{a}bd\bar{e} + a\bar{b}\bar{d}e + a\bar{b}d\bar{e} + ab\bar{d}\bar{e} \\ & \bar{a}\bar{c}de + \bar{a}c\bar{d}e + \bar{a}cd\bar{e} + a\bar{c}\bar{d}e + ac\bar{d}\bar{e} + a\bar{c}d\bar{e} \\ & \bar{b}\bar{c}de + \bar{b}c\bar{d}e + \bar{b}cd\bar{e} + b\bar{c}\bar{d}e + b\bar{c}d\bar{e} + bc\bar{d}\bar{e}. \end{aligned}$$

The standard canonical form of P is

$$\begin{aligned} P = {} & ab\bar{c}\bar{d}\bar{e} + a\bar{b}c\bar{d}\bar{e} + abc\bar{d}\bar{e} + \bar{a}b\bar{c}d\bar{e} \\ & \bar{a}b\bar{c}d\bar{e} + ab\bar{c}\bar{d}\bar{e} + \bar{a}\bar{b}cd\bar{e} + a\bar{b}cd\bar{e} \\ & \bar{a}bcd\bar{e} + a\bar{b}\bar{c}\bar{d}e + \bar{a}b\bar{c}\bar{d}e + a\bar{b}c\bar{d}e \\ & \bar{a}\bar{b}c\bar{d}e + a\bar{b}c\bar{d}e + \bar{a}bc\bar{d}e + \bar{a}\bar{b}\bar{c}de \\ & a\bar{b}\bar{c}de + \bar{a}b\bar{c}de + \bar{a}\bar{b}cde + \bar{a}bc\bar{d}\bar{e}. \end{aligned}$$

In closing, it is felt that Quine's canonical form is superior to the standard canonical form, due to its characteristic of retaining or reducing the degree of complexity of the polynomial representation and its amendability to machine computation.

BIBLIOGRAPHY

1. Phister, Montgomery. Logical design of digital computers. New York, Wiley, 1958. 508 p.

2. Quine, W. V. A way to simplify truth functions. The American Mathematical Monthly 62:627-631. 1955.

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

4. Rosenbloom, Paul. The elements of mathematical logic. New York, Dover, 1950. 214 p.

5. Samson, Edward W. and Burton E. Mills. Circuit minimization: Algebra and algorithms for new Boolean canonical expressions. 1954. 54 p. (AFCRC Technical Report 54-21)(Multilith).