VICINITIES, MINTERMS AND PRIME IMPLICANT COVERINGS

Norman M. Martin

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

TR-86-04 January 1986

Abstract

The special properties of the vicinity, i.e. the largest n-cube covering a given minterm but none of its adjacent zeroes, gives rise to an extremely convenient and flexible method for minimizing sum-product and related Boolean forms, since it allows direct calculation of essential prime implicants and the generation of remaining terms of minimal cost expressions in order of increasing complexity, thus obviating much of the calculation involved in standard methods.

1. Theory of Vicinities

The purpose of this paper is to prove a number of theorems of elementary switching theory which underlie a method of simplifying sum-product and related Boolean functions in a way generally more efficient than prevailing methods, especially for manual simplification. In the presentation, we assume the usual terminology and the relationship between n-cubes and product terms.

Standard methods rely, as does this one, on the relationship between cost functions having monotonicity properties with respect to product and sum and prime implicants which are product implicants whose expression cannot be shortened to produce an implicant, and calculate, first, the set of all prime implicants and from this set, determine which are essential (i.e. which must occur in every representation of the given function as a sum of prime implicants) [1]. The algorithm which is derivable from the theorems which follow instead determines essential prime implicants directly and is, furthermore, readily applicable to popular geometric representations such as Karnaugh maps and n-cubes.

In order to conveniently map partial functions we will assume that we are given three Boolean functions f_0, f_1 and f_2 such that $f_0 + f_1 + f_2 = 1$ and $f_0f_1 = f_0f_2 = f_1f_2 = 0$ are Boolean identities. f_0 will be taken as determining the cases in which the partial function is 0, f_1 those in which it is 1 and f_2 those in which it is undefined. The minimal cost sum-product expression will be that sum-product expression which (strictly speaking, whose correlated function) contains f_1 and is contained in $f_1 + f_2$. We will mean by a literal either a variable or its complement and by minterm a product of literals containing no variables twice and at least all variables on which any of f_0, f_1 or f_2 depend (and in referring to several minterms we require that any variable in one is in all).

We shall now add a few definitions of our own. Two minterms M_1 and M_2 will be termed *adjacent* provided there are literals L_1 and L_2 and a product of literals P such that $M_1 = L_1P$ and $M_2 = L_2P$ and $L_1L_2 = 0$. A minterm M_2 will be said to be an *adjacent zero* of a minterm M_1 provided M_2 is adjacent to M_1 and $M_2 \subset f_0$. We will term L an *adjacency literal* of a minterm M_1 provided there is a minterm M_2 which is an adjacent zero of M_1 and a literal L_1 and a product term P such that $M_1 = LP$ and $M_2 = L_1P$. V is the *vicinity* of a minterm M_1 provided it is the product of all the adjacency literals of M_1.

Theorem 1. Every product term implicant of $f_1 + f_2$, of which a minterm M is an implicant, is an implicant of the vicinity of M.

Proof: Let $A \subset f_1 + f_2$, $M \subset A$ and V the vicinity of M. If A were not implicant of V, since it is a product of literals there would be a literal L in V which is not in A. Let B be a product term such that M = LB. Then there is a literal L' such that LL' = 0. Then L'B is an adjacent zero of M. Hence, since every literal in A is in M and L is not in A, B $\subset$ A. Since B is an implicant of $f_1 + f_2$, L'B $\subset f_1 + f_2$. But since it is an adjacent zero of M, L'B $\subset f_0$. But since L'B is a minterm, this gives L'B $\subset f_0$ and L'B $\not\subset f_0$.

Theorem 2. Let M be a minterm and V its vicinity, then V is an implicant if and only if V is a prime implicant.

Proof: If V is a prime implicant it is an implicant. For the converse assume V is an implicant. Then there is a prime implicant A such that V $\subset$ A. But by Theorem 1, A $\subset$ V and hence V is prime.

Theorem 3. Let M be a minterm such that M $\subset f_1$ and let V be its vicinity. Then V is an implicant if and only if V is an essential prime implicant.

Proof: If V is an essential prime implicant, it is an implicant. For the converse assume V is an implicant. By Theorem 2, V is prime. If V were not essential, there would be a prime implicant P such that V $\neq$ P and M $\subset$ P. But then by Theorem 1, P $\subset$ V and, since P is prime, V = P. Hence V is an essential prime implicant.

Theorem 4. A product term A is an essential prime implicant if and only if there is a minterm M such that M $\subset f_1$ and A is the vicinity of M.

Proof: If there is a minterm M such that A is a vicinity of M and M $\subset f_1$, A is an essential prime implicant by Theorem 3. For the converse, assume A is an essential prime implicant. Obviously, $f_1A \neq 0$, since otherwise A would be redundant. Let $M_1,\ldots,M_n$ be all those minterms such that $M_i \subset f_1A$ for i = 1,...,n. Suppose for each i $(1 \leq i \leq k)$, there is a j $(1 \leq j \leq n)$ such that $M_j = L_iB_i$ and a literal L_i' such that $L_i' B_i \not\subset f_0$ and L_iL_i' = 0. Then $L_i' B_i \subset f_1 + f_2$. But then $B_i = L_iB_i + L_i' B_i \subset f_1 + f_2$. Hence, since B_i is an implicant, there is a prime implicant P_i such that $B_i \subset P_i$. Hence, if $f_1 \subset A + B \subset f_1 + f_2$, it follows that $f_1 \subset \sum_{i=1}^{n} M_i + B \subset f_1 + f_2$ and $M_i \subset P_i$, $f_1 \subset \sum_{i=1}^{n} P_i + B \subset f_1 + f_2$. Since $B_i \not\subset A$, $P_i \not\subset A$ and hence A is not essential.

Theorem 5. Let $E_1,\ldots,E_n$ be the essential prime implicants. If $f_1 \subset \sum_{i=1}^{n} E_i$, $\sum_{i=1}^{n} E_i$ is the minimum cost sum-product expression (provided the cost function satisfies the usual monotonicity properties).

Proof: Since $E_i \subset f_1 + f_2$, $f_1 \subset \sum_{i=1}^{n} E_i \subset f_1 + f_2$ and since the E's are essential $\sum_{i=1}^{n} E_i$ is the minimum cost expression.

Theorem 6. Let $E_i,...,E_n$ be the essential implicants and $M_i,...,M_m$ be the minterms such that $M_i \subset f_1$ and $M_i \sum_{j=1}^{n} E_j = 0$. For each i $(1 \leq i \leq n)$, let V_i be the vicinity of M_i. Let $f_1^* = f_1 \sum_{i=1}^{n} V_i \text{ compl}(\sum_{i=1}^{n} E_i)$, $f_2^* = (f_2 + \sum_{i=1}^{n} E_i) \bullet \sum_{i=1}^{m} V_i$ and $f_0^* = \text{compl}(\sum_{i=1}^{m} V_i) + f_0$. Then if $\sum_{i=1}^{n} E_i + \sum_{i=1}^{k} P_i$ is the minimum cost expression A satisfying $f_1 \subset A \subset f_1 + f_2$ and P_i are prime implicants, $\sum_{k=1}^{k} P_i$ is the minimum cost expression A satisfying $f_1^* \subset A \subset f_1^* + f_2^*$.

Proof: If $P_j \subset f_2 + \sum_{i=1}^{n} E_i$, it could be eliminated. But since $P_j \subset f_1 + f_2$, it follows that $P_j \bullet f_1 \bullet \text{compl}(\sum_{i=1}^{n} E_i) \neq 0$ and hence that there must be at least one minterm in it. Hence, there is an $M_i \subset P_j$. Then by Theorem 1, $P_j \subset \sum_{i=1}^{m} V_i$. Since $f_1^* + f_2^* = \sum_{i=1}^{m} V_i (f_1 + f_2)$, it follows that $\sum_{i=1}^{k} P_i \subset f_1^* + f_2^*$, $(\sum_{i=1}^{m} M_i) \bullet \sum_{i=1}^{n} E_i = 0$ and $f_1 \subset \sum_{i=1}^{n} E_i + \sum_{i=1}^{m} M_i$. Thus $f_1 \sum_{i=1}^{m} V_i \text{ compl}(\sum_{i=1}^{n} E_i) = \sum_{i=1}^{m} M_i \subset \sum_{i=1}^{k} P_i$. Hence $f_1^* \subset \sum_{i=1}^{k} P_i \subset f_1^* + f_2^*$. Suppose there were a B with less cost than $\sum_{i=1}^{k} P_i$ and $f_i^* \subset B \subset f_1^* + f_2^*$. Then since $(\sum_{i=1}^{n} E_i) + B$ is of less cost than $\sum_{i=1}^{n} E_i + \sum_{i=1}^{k} P_i$ and $f_1 \subset \sum_{i=1}^{n} E_i + B$. Since $\sum_{i=1}^{n} E_i \subset f_1 + f_2$ and $f_1^* + f_2^* \subset f_1 + f_2$, $\sum_{i=1}^{n} E_i + B \subset f_1 + f_2$ and $\sum_{i=1}^{n} E_i + \sum_{i=1}^{k} P_i$ would not be the minimum cost sum-product expression, contrary to assumption.

Theorem 7. Let E be an essential term and M a minterm such that $M \subset E$. Then either E is the vicinity of M or the minimum cost expression is independent of whether $M \subset f_1$ or $M \subset f_2$.

Proof: Let E be essential, $M \subset E$, V the vicinity of M, and $V \neq E$. Since $M \subset E$, $E \subset V$, by Theorem 1. Then $V \not\subset f_1 + f_2$ since otherwise E could not be prime, by Theorem 2. Hence, more specifically, V is not essential. Hence the set of essential terms does not depend on whether $M \subset f_1$ or $M \subset f_2$. But since $M \subset \sum_{i=1}^{n} E_i$ of Theorem 6 either M $\subset f_0{}^*$ if $M \subset \sum_{i=1}^{m} V_i$ and $M \subset f_2{}^*$ otherwise. In neither case does this depend on whether $M \subset f_1$ or $M \subset f_2$.

2. Determination of Essential and Optional Prime Implicants

Theorems 4 and 7 together provide the basis for an algorithm for determining the essential terms, viz. Let $M_1,\ldots,M_n$ be the minterms in f_1 in any order. For each i, check if M_i is contained in an essential term already determined. If it is, proceed to i+1. If not, determine its vicinity by writing the product of its adjacency literals. Observe if the vicinity is in $f_1 + f_2$ (for Karnaugh maps or the like, this means to see if the n-cube fails to cover any 0-cell). If it does, add the vicinity to the list of essential terms and proceed to i+1. If not, just proceed to i+1. Suitable choice of the order may decrease the number of steps concerned, but cannot affect the final result.

If the minimum cost expression is the sum of essential terms, the algorithm yields it directly. If not, Theorem 6 usually substitutes a somewhat simpler problem to be solved by one or another conventional algorithms.

It should be noted that in the reduced problem generated in accordance with Theorem 6, there are never essential terms, but that, in most cases the number of prime implicants will be substantially decreased.

At this point in the process, one additional advantage which can be utilized systematically is present as compared to e.g. Quine-MacClusky viz. as a consequence of Theorem 1 we know that every product implicant and hence every prime implicant which covers M must contain every literal in the expression for M's vicinity. From the definition of prime implicant we know that prime implicants may contain no literals other than those of M. The process for determining the prime implicants that cover the remaining minterms then is:

1. Adjoin to the expression for V_m one literal in M but not in V_m (in all possible ways). For each resulting product, it is an implicant, it is prime. Let us call those resulting products which are not implicants "unsuccessful candidates of the first level" and the literals added to create them "unsuccessful literals."

2. Adjoin to every unsuccessful candidate of the k^{th} level an unsuccessful literal (in all possible ways) such that the resulting product does not have all the literals in a prime implicant already generated. If the resulting product is an implicant it is prime. If not, call it an unsuccessful candidate of the $k + 1^{st}$ level.

The process should obviously terminate if either we have finished the level in which the number of literals is one less than the number in M or if at some level we have no unsuccessful candidates (since then it *cannot* continue). It can however also be discontinued if on the n^{th} level, there are no more than n unsuccessful candidates, since then the next level is bound to yield no products which are not the result of adding a literal to a prime implicant already detected and hence no candidates on the $n+1^{st}$ level.

We illustrate the method described using Karnaugh maps.

A/B		0				1			
	CD/EF	00	01	11	10	00	01	11	10
0	00	0	0	0	0	0	1^e	1^f	1^g
	01	0	1^a	1^b	0	1^h	0	1^i	0
	11	?	?	?	?	?	?	?	?
	10	0	0	1^c	1^d	1^j	0	1^k	0
		00	01	11	10	00	01	11	10
1	00	0	0	0	1^l	1^m	1^n	1^o	0
	01	0	0	0	0	1^p	1^q	0	1^r
	11	?	?	?	?	?	?	?	?
	10	?	?	?	?	?	?	?	?

	V	ess?	Covered by		V	ess?	Covered by
a.	$\overline{AB}DF$	x	a	j.	$A\overline{CD}E$	x	j
b.	$\overline{A}DF$		a	k.	CD		
c.	CE		d	l.	$\overline{A}BC\overline{DF}$	x	l
d.	$\overline{A}CE$	x	d	m.	$AB\overline{C}$	x	m(q)
e.	$AD\overline{EF}$	x	e	n.	A		e,m(q),r
f.	A		e,g	o.	$AD\overline{F}$		e
g.	$A\overline{B}C\overline{EF}$	x	g	p.	A		h,m(q),r
h.	$A\overline{CD}F$	x	g	q.	$AB\overline{C}$	x	m(q)
i.	$\overline{B}CD$			r.	$AB\overline{D}F$	x	r

Table 1

Table 1 illustrates the application of the method to a function chosen more or less at random (f_2 was chosen arbitrarily as BE + EF and f_1 then literally generated randomly). There are 18 minterms in f_1, 24 in f_2 and the remaining 22 in f_0. There are 9 essential prime implicants as indicated. The author chose these 9 in 10 tries (with 1 negative test on minterm i- and a final test one on k). ΣV_i (of Theorem 6) thus becomes

$\overline{B}CD + CD$ which equals CD, so that the reduced partial function becomes:

0	0	0	0	0	0	?	0
0	0	?	0	0	0	1	0
0	0	?	0	0	0	?	0
0	0	?	0	0	0	1	0
0	0	0	0	0	0	?	0
0	0	0	0	0	0	0	0
0	0	?	0	0	0	?	0
0	0	?	0	0	0	?	0

Table 2

The remaining prime implicant for minimizing diode cost ($A\overline{B}CD$) is extremely obvious and application of a standard method such as iterated consensus is quite easy and short rather than laborious as it would have been with the initial function.

Applying instead our indicated algorithm we note that the vicinity for i is $\overline{B}CD$ and i is $A\overline{B}CD\overline{E}F$ while the vicinity for k is CD and k is $A\overline{B}CD\overline{E}F$. For i our first level candidates are then: $A\overline{B}CD$, $\overline{B}CD\overline{E}$, $\overline{B}CDF$ of which the first and third are implicants and hence prime. Since $\overline{B}CD\overline{E}$ is the only unsuccessful candidate, they are all of the prime implicants that cover i. For k, our first level candidates are: ACD, $\overline{B}CD$, CDE, $CD\overline{F}$ of which only CDE is an implicant and hence prime. Our successful literals are A, $\overline{B}$ and $\overline{F}$ and hence our second level candidates are $A\overline{B}CD$, ACDF and $\overline{B}CD\overline{F}$, $A\overline{B}CD$ is a prime implicant as we have seen and since $ACD\overline{F}$ is an implicant, it is also prime. $A\overline{B}CD$ is not an implicant and hence an unsuccessful candidate. Since there are not more than 2 unsuccessful candidates on level 2, we are left with $A\overline{B}CD$, $\overline{B}CDF$, CDE and $ACD\overline{F}$ of which the first two cover i and the first, third and fourth cover k.

3. Multiple Minterm Covering and the Full Algorithm

This process can be simplified additionally if we know the cost function. For instance, returning to the point at which we have calculated the vicinities and assuming all literals equal in cost we know that the best we could hope to do is to choose a 4 literal prime implicant which covers i and k since no prime implicant which covers i can have less than 4 literals (since its non-implicant vicinity has 3). Such a prime implicant would however have to agree with both i and k in all its literals and hence could only be $A\overline{B}CD$. Since this is a 4 literal implicant, it is prime and is an appropriate choice. Note that we do not need to determine the other prime implicants.

Suppose, on the other hand, our cost function assigned a cost of 1 to each use of a literal except A, a cost of 5 for each use of A and no additional charge for the "and" gate. In that case we can, in advance of determining the prime implicants which cover i and k, determine the following order of preference for coverings:

1. A 4 literal prime implicant which covers i and k and contains no A.

2. A 5 literal one which satisfies the same condition.

3. A 4 literal one which covers i and a 3 literal one which covers k, neither of which contains A.

4. Either a 5 and 3, or two 4's, neither containing A or a 4 literal one containing A.

Since we know that any prime implicant which covers i must have at least the literal in its vicinity plus one and i and k have only 4 literals in common, one of which is A, it follows that 1. and 2. above cannot occur. Hence a solution satisfying 3. is minimal cost. We then find $\overline{B}CDF$, either by observation of the map or by adding successively literals in i which are neither A nor in the expression for the vicinity. In the same way we find CDE.

Let us call a set of minterms compatible if there exists a product implicant which covers all of them. The following theorem relates vicinity and compatibility.

Theorem 8. Let $m_1,...,m_n$ be minterms. We will call P the p-cover of $m_1,...,m_n$ provided P is the longest product such that $\sum_{i=1}^{m} M_i \subset P$ (i.e. P is the product of all literals shared by all of the M_i). Let $V_1,...,V_m$ be the vicinities of $m_1,...,m_n$ respectively. Then if P is an implicant a necessary and sufficient condition for there to be a prime implicant using $m_1,...,m_n$ is $P \subset \prod_{i=1}^{m} V_i$.

Proof: Let P' be a prime implicant covering each m_i. This by Theorem 1; P' $\subset \prod_{i=1}^{m} V_i$. Since P' covers each m_i, every literal in P' is in each m_i and hence in P. Hence P $\subset$ P' and P $\subset \prod_{i=1}^{m} V_i$. Suppose P $\subset \prod_{i=1}^{m} V_i$. Since P is an implicant there exists a prime implicant P' such that P $\subset$ P'. But since $\sum_{i=1}^{m} m_i \subset P$, $\sum_{i=1}^{m} m_i \subset P'$.

Theorem 8 results in a structure similar to that between vicinity and minterm expressions, namely every literal in the product of vicinities is in every prime implicant that covers $m_1,...,m_n$ and only literals in the expression for the p-cover, that is, literals in each of $m_1,...,m_n$, are. Since, of course, the product of vicinities may itself be a prime implicant, the generating algorithm starts with the product rather than one step later.

The systematic determination then proceeds as follows:

1. Determine the essential terms.

2. For each remaining minterm, determine all the remaining minterms in its vicinity.

3. Prune the lists determined in step 2 by eliminating the non-symmetric cases.

4. For each minterm still remaining, determine the product of the vicinities and the P-cover of the minterms on its list. If the P-cover is an implicant and included in the product, determine the least cost prime implicant(s) included in the product of vicinities and check that it is a minimal cost cover of the minterm. If it is, add it to the list of accepted terms, eliminate the minterms it covers from the problem, correct the lists and proceed. Otherwise proceed to the next minterm [2].

5. If any minterms remain, compare costs of minimal covering products of each remaining compatible covering and choose the best.

To illustrate, consider our test function. Minterm i is in the vicinity of k and vice versa. The product of their vicinities is $\overline{B}CD$ and their P-cover is $A\overline{B}CD$. Since $\overline{B}CD$ is not a prime implicant, $A\overline{B}CD$ is the only prime implicant which covers both and hence should be added if it is a minimal cost prime implicant for either i or k. For example, if the cost of increasing any one literal is the same, step 4 dictates it. Otherwise we compare its cost with the minimal cost coverings of i and k (which in this case would have to be $\overline{B}CDF$ and CDE) and choose the cheapest between adding $A\overline{B}CD$ or both of $\overline{B}CDF$ and CDE.

Notice that as minterms get covered as a result of terms adopted by application of steps 2-4, they are eliminated from the problem, so that normally as the method proceeds, not only does the number of lines indicating minterms decrease, but also the number of entries indicating minterms which might be covered with the given minterm decrease as well; specifically, only minterms still to be covered may appear there. Accordingly, especially in manual application, it is frequently desirable to apply steps 2-4 (and, occasionally, step 5 as well) to promising minterms one at a time, thus taking advantage of the simplifications which result. Indeed, when a minterm cost covering of a minterm is one literal more than the vicinity, this fact can frequently be directly read off of the Karnaugh map. To illustrate, consider the following partial function (with the cost being one unit for each occurrence of a literal):

B/A		0				1			
	EF/CD	00	01	11	10	00	01	11	10
	00	0	1^c	0	0	1^o	0	0	0
0	01	1^a	?	1^n	1^e	0	1^2	?	0
	11	?	1^d	2	1^b	1^p	0	0	1^o
	10	1^b	0	2	1^g	1^g	0	1^u	1^t
	00	0	1^k	?	?	0	0	0	0
1	01	0	0	1^m	1^l	0	1^w	1^{aa}	?
	11	1^i	0	0	2	?	1^x	0	0
	10	1^j	0	1^n	0	1^v	1^y	1^{bb}	1^2

The vicinity of a is $\overline{AB}F$ which, since it covers no zeroes, is an essential prime implicant and also covers d, h, e and f. The vicinity of j similarly determines the essential prime implicant $\overline{CD}E$ which also covers b, i, p, q and v. Similarly, the vicinity of a determines $\overline{ABCDF}$ which also covers q, and that of s $\overline{BDE}$ which also covers b, f, g, p, q and t. Checking the remaining l-cells we obtain the following vicinities (none of which are implicants):

c	$\overline{ACDE}$	n	$D\overline{E}F$	z	$AE\overline{F}$
k	$\overline{ABEF}$	u	$CE\overline{F}$	aa	$\overline{E}F$
l	C	w	ADF	bb	$E\overline{F}$
m	$C\overline{E}$	x	$AB\overline{C}$		
n	$CD\overline{F}$	y	ABE		

The vicinity of c contains but one remaining minterm, k and $A\overline{C}D\overline{EF}$ is minimal for c and covers k. The vicinity of n contains u and bb and $CDE\overline{F}$ is minimal for n and covers u and bb. The vicinity of r contains m, w and aa, but since r is not in the vicinity of m, the prime implicant. $AD\overline{E}F$, which covers w and aa is minimal for r, is the least cost way of covering r by rules 2-4. Now the vicinity of x covers w and y, but since w is already covered and $AB\overline{C}E$ is minimal for x and covers y, it is accepted by rules 2-4. The vicinity of n covers w and bb. This leaves us with l and m. The vicinity of m, $C\overline{E}$ covers l and since none of $\overline{A}C\overline{E}$, $BC\overline{E}$, $CD\overline{E}$ and $C\overline{E}F$ are implicants, $\overline{A}C\overline{E}F$ (or $BC\overline{E}F$) is minimal for m and covers l. Hence $\overline{AB}F + \overline{CD}E + A\overline{BCDF} + \overline{BD}E + A\overline{C}D\overline{EF} + CDE\overline{F} + AD\overline{E}F + AB\overline{C}E + \overline{A}C\overline{E}F$ is a minimal cost covering of the given function.

The reader who intends to use this method should note that when the cost function, though monotonic, is rather complex, he may prefer to use the first two algorithms and choose the best coverings among the optional prime implicants rather than the more complicated full algorithm, since the advantage of the full algorithm lies primarily in the elimination of the calculation of some of the optional terms, an advantage which is considerably less when the cost function is complex [3].

Affiliation of author: Professor of Computer Sciences, of Philosophy and of Electrical Engineering, University of Texas at Austin.

4. References

[1] The classic references are:

W. V. Quine, "The problem of simplifying truth function," *American Mathematical Monthly*, Vol. 59, pp. 521-531 (Oct. 1952).

W. V. Quine, "A why to simplify truth functions," *American Mathematical Monthly*, Vol. 62, pp. 627-631 (Nov. 1955).

E. J. McCluskey, Jr., "Minimization of Boolean functions," *Bell System Technical Journal*, Vol. 35, pp. 1417-1444 (Nov. 1956).

An excellent summary of the general theory can be found in E. J. McCluskey, Jr. and T. C. Bartee (ed.) *A Survey of Switching Circuit Theory*, McGraw-Hill, New York, 1962, pp. 67-88.

[2] Note that this implies that where no other remaining minterms are compatible with a given minterm, we choose the minimal cost prime implicant which covers the minterm.

[3] The part of the method depending on Theorem 4 was discovered by the author early in 1957 and expanded to a full algorithm in 1958, in connection with his logical design course at UCLA. The initial formulation was stated in a form more directly in the terminology of Karnaugh maps and took its present form early in the 1960's. A preliminary publication (primarily for class notes) is contained in the author's "The Adjacent-Zero Method with Karnaugh Maps", Report TNN-77, Computation Center, University of Texas, 1968. An independent result roughly equivalent to Theorem 4 was made (apparently in the mid-50's) and published by Antonin Svoboda in 1968 with a somewhat different (and in this author's judgement, less convenient) algorithm. See G. Klir, *Introduction to the Methodology of Switching Circuits*, Van Nostrand, New York, 1972, pp. 88-120, for details of Svoboda's work.

It should be noted that the indicated method is readily amenable to machine representation (although the advantages of choosing good candidates for determining essential terms probably is not practically available). While not decisive, preliminary evidence seems to indicate efficiency advantages relative to the Quine-McCluskey algorithm in machine versions. These advantages are obviously more extreme in manual application.