

新しいブール関数簡単化アルゴリズム

苦 米 地 宣 裕

A New Algorithm for Boolean-Function Minimization

Nobuhiro TOMABECHI

Abstract

This paper proposes a novel algorithm for Boolean-function minimization. The algorithm is used for obtaining correct answers in the logic circuit design training software. In such CAI applications, the number of logic variables is limited to the one equal to or less than 5, on the other hand, the processing time is required to be less than a few seconds. In the proposed method, the ternary Karnaugh map is introduced. The minimizing process is as follows; 1. all of the terms obtainable from the given Boolean-function are made in the ternary Karnaugh map, 2. redundant terms are deleted. The proposed method is compared to the traditional Quine-McCluskey method. It is found that the processing time can be reduced to 1/3 of the one by the Quine-McCluskey method.

1. ま え が き

本研究は、論理回路設計演習用 CAI ソフト開発の一環として行われたもので、この CAI ソフトに用いる新しいブール関数簡単化アルゴリズムを提案するものである。

ブール関数の簡単化については、ブール代数の法則を用いる方法やカルノー図を用いる方法がよく知られている^{1)~3)}。これらは、人間の記号処理能力、図形処理能力を利用する方法で、いわば人手による方法といえることができる。一方、コンピュータを用いる簡単化の方法も、クワイン・マクラスキの方法をはじめ種々研究されているが^{4),5)}、これらは、主に、論理変数の多い大規模な論理回路の自動設計を目的として行われているようである。

ところで、論理回路設計演習に用いる CAI ソフトを作成しようとするとき、コンピュータを用いるブール関数簡単化の必要性が生ずる。ブー

ル関数簡単化の演習問題において、正解か否かを自動的に判定するためである。この CAI 応用においては、論理変数の個数は、2~5 個に限定されるが、簡単化に要する時間が、人間の解答待ち時間の心理的限界とされる数秒以内が望まれる。また、使用するコンピュータは安価なパソコンが想定され、演算速度が限定される。従来のクワイン・マクラスキ法に基づいてプログラム作成を行った結果では、上記条件が満足されないことが明らかとなり⁶⁾、より処理時間の短いアルゴリズムの開発が期待されている。

本論文で提案する方法では、まず、ブール関数の表現に、論理変数の値が 3 値をとるカルノー図を導入する⁷⁾。本図により、記号数の一定しない項より成る任意のブール関数を系統的に表現することが可能となる。提案するアルゴリズムの特徴は次の 2 点に要約される。① 与式より生じ得るすべての項を、3 値カルノー図に一挙に作成し、次いで、冗長な項を種類ごとに順次削除していく、② 非必須な項の最適な組み合わせ（最小被覆という）を項の連鎖をたど

平成 5 年 12 月 15 日受理
八戸工業大学 電気工学科 教授

ることにより求める。なお、本論文では第1段階として、禁止なしの場合について論じている。本論文で提案したアルゴリズムに従ってプログラム作成を行い、クワインーマクラスキ法と比較を行った。その結果、提案したアルゴリズムによれば、処理時間がクワインーマクラスキ法による場合の1/3に短縮できることが明らかとなった。

2. 3値カルノー図とその性質

2.1 3値カルノー図の導入

コンピュータのプログラム上では、 P 変数のカルノー図は P 次元の配列で表わすことができる。この P 次元配列を、 $K(A, B, \dots, N)$ 、あるいは、単に K と表わすことにする。ここで、 A, B, \dots, N は、論理変数を表わしており、いずれも、0か1の値をとる。カルノー図のセル、すなわち、最小区画は、配列要素 $K(A, B, \dots, N)$ で表わされる。このセルはブール関数の最小項に対応するので、カルノー図はブール関数を最小項の和で表わすのに適しているといえる。しかし、最小項だけでなく、変数

記号（リテラル）の個数の異なる任意の項よりなる関数をコンピュータで扱おうとすると、カルノー図の表現では困難となる。

この問題に対して、本論文では、3値カルノー図を提案する。これは、通常の2値のカルノー図を拡張して、論理変数が0, 1, 2の3値をとるようにしたものである。3値の論理値は変数を X とすると、論理値0は \bar{X} 、1は X 、2はヌール（記号がないこと）に対応づける。 X のヌールを X_n で表わす。このように3値カルノー図のセルは、最小項に限らず変数記号の個数が任意の項を表わすことができる。従って、 P 変数3値カルノー図は、 P 変数で作り得る任意のブール関数を表現することができる。図1に3値カルノー図の3変数の場合のセルと項の対応を示している。本図において、 T はすべての記号のヌールを表わしている。これは恒等的1に等しい。また、図2に、3値カルノー図を用いたブール関数の表現例を示している。 P 変数3値カルノー図は P 次元配列で表わすことができる。この P 次元配列を、 $U(A, B, \dots, N)$ 、あるいは、単に U と表わすことにする。

	\bar{A}	A	A_n	\bar{A}	A	A_n	\bar{A}	A	A_n
\bar{B}	$\bar{A}\bar{B}\bar{C}$	$A\bar{B}\bar{C}$	$\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$A\bar{B}C$	$\bar{B}C$	$\bar{A}\bar{B}$	$A\bar{B}$	\bar{B}
B	$\bar{A}B\bar{C}$	$AB\bar{C}$	$\bar{B}C$	$\bar{A}BC$	ABC	BC	$\bar{A}B$	AB	B
B_n	$\bar{A}\bar{C}$	$A\bar{C}$	\bar{C}	$\bar{A}C$	AC	C	\bar{A}	A	T
	\bar{C}	\bar{C}	\bar{C}	C	C	C	C_n	C_n	C_n

図1 3値カルノー図のセルと最小項の対応（3変数の場合）

	\bar{A}	A	A_n	\bar{A}	A	A_n	\bar{A}	A	A_n
\bar{B}	1		1						
B									
B_n								1	
	\bar{C}	\bar{C}	\bar{C}	C	C	C	C_n	C_n	C_n

$$X = \bar{A}\bar{B}\bar{C} + \bar{B}\bar{C} + A$$

図2 3値カルノー図を用いたブール関数の表現例

2.2 用語の定義と3値カルノー図の性質

以下の論述に用いる用語の定義と3値カルノー図の性質について述べる。なお、単にカルノー図というときは、通常の2値のカルノー図を指すことにする。

[定義1] 吸収項, 被吸収項

ある2つの項, $X1, X2$ があって, $X1$ に含まれる最小項がすべて, $X2$ にも含まれているとき, 「 $X1$ は $X2$ の被吸収項である」, あるいは, 「 $X2$ は $X1$ の吸収項である」という。

[性質1] 吸収項の判別1

3値カルノー図上の2つの項, $X1, X2$ において, $X2$ の変数の値が2となる変数以外の変数の値が, $X1$ と $X2$ ですべて等しいとき, $X2$ は $X1$ の吸収項となる。

[例1] 吸収項の判別1の例

3変数の例を示す。 $X1 = A\bar{B}C$, $X2 = A\bar{B}$ とすると, $X1 = A\bar{B}C$ は単一の最小項であり, 一方, $X2 = A\bar{B}\bar{C} + A\bar{B}C$ となるので, $X2$ は $X1$ の吸収項である。この $X1, X2$ を3値カルノー図を用いて表わすと次のようになる。

$$X1 = A\bar{B}C = U(1, 0, 1)$$

$$X2 = A\bar{B} = U(1, 0, 2)$$

$X2$ の3番目の変数の値が2であるので3番目の変数は除くと, 他の2つの変数の値が $X1$ と $X2$ で一致するので, $X2$ は $X1$ の吸収項であると判定できる。

性質1を一般化すると, 次の性質2が得られる。

[性質2] 吸収項の判別2

3値カルノー図上の一つの項を, 一般に, $U(A, B, \dots, N)$ と表わす。この項を被吸収項とする吸収項は, $U(A, B, \dots, N)$ の P 個の変数を, $1 \sim P$ 個の2で置き換えて得られる。ただし, 置き換えて得られた変数の組が元の変数の組と同じになるものは除く。

[例2] 吸収項の判別2の例

3変数の例を示す。被吸収項を $U(A, B, C)$ と表わす。これに対する吸収項は次のようにな

る。

$$U(A, B, 2), U(A, 2, C), U(2, B, C), U(A, 2, 2), U(2, B, 2), U(2, 2, C), U(2, 2, 2)$$

ただし, これらの項の中で, 2と記述された位置に対応する A, B, C の値が同じく2となるときは除外する。

なお, 性質2において生ずる吸収項の数は, 変数の数を P とすると, $2^P - 1$ 個となる。

[性質3] 吸収項の判別3

カルノー図 K 上の一つの項を, 一般に, $K(A, B, \dots, N)$ と表わす。この項を被吸収項とする3値カルノー図上の吸収項は, $U(A, B, \dots, N)$ の P 個の変数を, $1 \sim P$ 個の2で置き換えて得られる。

2値のカルノー図上の項が被吸収項の場合は, 変数の値は0か1だけで2となることがないので, 性質2の場合についた但し書きは必要なくなる。

なお, 性質3において生ずる吸収項の数は, 性質2の場合と同様 $2^P - 1$ 個となる。

[定義2] 必須項

あるブール関数において, その関数に含まれる一つの項 X を除いた関数の真理値表が X を除く前の関数と異なるとき, 項 X を必須項という。

[定義3] 連結項, 連結点

2つの項 $X1, X2$ があって, 同一の最小項を含むとき, 「 $X1$ と $X2$ は連結している」という。また, 「 $X1$ は $X2$ の連結項である」, あるいは, 「 $X2$ は $X1$ の連結項である」という。また, $X1$ と $X2$ の共通の最小項を連結点という。

[性質4] 連結項の判別

3値カルノー図上の2つの項, $X1, X2$ において, 論理値2をとる変数以外の変数の値が, $X1, X2$ ですべて一致するとき, 2つの項は連結していると判定できる。

[例4] 連結項の判別例

4変数の例を示す。 $X1 = AB\bar{D}$, $X2 = AC\bar{D}$ とすると, $X1 = ABC\bar{D} + ABD\bar{D}$, $X2 = A\bar{B}C\bar{D} + ABC\bar{D}$ となるので, $X1$ と $X2$ は同一の最小項

$ABC\bar{D}$ を含む。従って、 $X1$ と $X2$ は連結している。この $X1, X2$ を 3 値カルノー図を用いて表わすと次のようになる。

$$X1 = AB\bar{D} = U(1, 1, 2, 0)$$

$$X2 = AC\bar{D} = U(1, 2, 1, 0)$$

論理値 2 をとる変数以外の変数、すなわち、1 番目と 4 番目の変数の値が、 $X1, X2$ で一致しているので、2 つの項は連結していると判定できる。

[定義 4] 連鎖

連結し合っている一連の項を連鎖という。ただし、連鎖には被吸収項や必須項は含まれていないとする。

図 3 に、連鎖の例を示している。ただし、2 値のカルノー図で表現している。

[定義 5] 端末項

連鎖に含まれている項の中で、必須項と連結している項を端末項という。

[定義 6] 分岐項, 分岐点

連鎖に含まれている項の中で、3 個以上の連結項を持つ項を分岐項という。また、分岐項とそれに連結する項の連結点を分岐点という。

[定義 7] 単連鎖

端末項か分岐項を両端だけに持つ連鎖を単連鎖という。

[定義 8] 閉連鎖

端末項も分岐項も持たない連鎖、すなわち、単一のループをなす連鎖を閉連鎖という。

図 4 に、閉連鎖の例を示している。ただし、2 値のカルノー図で表現している。

	A	A	\bar{A}	\bar{A}	
B	1				\bar{D}
B	1	1			D
\bar{B}		1	1		D
\bar{B}					\bar{D}
	\bar{C}	C	C	\bar{C}	

$$X = AB\bar{C} + ABD + ACD + \bar{B}CD$$

図 3 連鎖の例 (2 値カルノー図で表現)

	A	A	\bar{A}	\bar{A}	
B	1	1	1		\bar{D}
B	1		1		D
\bar{B}	1	1	1		D
\bar{B}					\bar{D}
	\bar{C}	C	C	\bar{C}	

$$X = AB\bar{C} + A\bar{C}D + A\bar{B}D + \bar{B}CD + \bar{A}CD + \bar{A}BC + BC\bar{D} + AB\bar{D}$$

図 4 閉連鎖の例 (2 値カルノー図で表現)

単連鎖と閉連鎖に関して以下の性質が成り立つ。

[性質 5]

単連鎖を構成する項の記号数はすべて等しい。ただし、端末項と分岐項は除く。

[性質 6]

単連鎖から得られるカルノー図を $K3$ とすると、単連鎖を構成する項は次々に連結し合って、 $K3$ を 2 重に被覆する。ただし、端末項の必須項と重なる部分、および、分岐点は 1 重となる。

[性質 7]

単連鎖を構成する項は、互いに連結することのない項で構成されるグループに 2 つに分けることができる。

[性質 8]

閉連鎖を構成する項の記号数はすべて等しい。

[性質 9]

閉連鎖を構成する項の個数は偶数となる。

3. 簡単化アルゴリズム

本論文では、禁止 (don't care) は考慮しないこととする。また、論理変数の数は 5 以下とする。また、簡単の意味は、ブール関数を積和標準形に表わしたとき、項数が少ないこと、項数が同じときは項の記号数が少ないことをいうこととする。

本論文で提案するブール関数簡単化の方法

は、まず、与えられたブール関数から生じ得るすべての項を3値カルノー図に作成し、次いで、冗長な項を項の種類ごとに順次削除していくという手順をとる。この手順は次のように定式化できる。ただし、簡化の対象となるブール関数はカルノー図 $K1$ の形で与えられるとする。また、解は3値カルノー図 U の形で得るとする。

[アルゴリズム 1] 簡化アルゴリズム

ステップ 1-1 カルノー図 $K1$ より生ずるすべての項を求め3値カルノー図 U に記入する。

ステップ 1-2 U より被吸収項を削除する。

ステップ 1-3 U より非必須項を求め、非必須項登録表に登録する。

ステップ 1-4 非必須項登録表より一つの連鎖を取り出す。

ステップ 1-5 連鎖から得られるカルノー図 $K2$ の最小被覆を求め、最小被覆に寄与しない項を U から削除する。

ステップ 1-6 ステップ 1-4, ステップ 1-5 の操作を非必須項登録表に連鎖がなくなるまでくり返す。

U に残った項が解となる。

アルゴリズム 1 の各ステップをより詳細に記述すると以下ようになる。

ステップ 1-1 のカルノー図より生ずるすべての項を求める処理は、次のように定式化できる。

[アルゴリズム 2] すべての項の算出

ステップ 2-1 3値カルノー図 U の各項をすべて1にする。

ステップ 2-2 カルノー図 $K1$ から出力が0となる項を1つとる。

ステップ 2-3 この項を被吸収項とする3値カルノー図 U 上の吸収項をすべて削除する。吸収項の識別には性質3を適用する。

ステップ 2-4 ステップ 2-2, ステップ 2-3 の操作を、カルノー図 $K1$ の出力が0となるすべての項についてくり返す。

3値カルノー図 U 上の1で残った項がカルノー図 $K1$ より生ずるすべての項となる。

上記のステップ 2-3 において、3値カルノー図 U 上に吸収項が存在するか否かの判別に、性質3を適用するとしている。性質3を適用すれば、通常考えられる性質1をすべての項について適用する方法に比較して、判別処理時間が大幅に短縮できる。ただし、性質3の適用にあたっては、判別文(以下、IF文という)を 2^p-1 個記述することが必要となる。本研究では、 $P \leq 5$ と想定しているので、IF文の数は最大でも31個にとどまる。

ステップ 1-2 の被吸収項の削除は、次のように定式化できる。

[アルゴリズム 3] 被吸収項の削除

ステップ 3-1 3値カルノー図より、出力が1となる項を1つとる。

ステップ 3-2 この項を被吸収項とする3値カルノー図上のすべての吸収項について1か0か調べる。1つでも吸収項が1ならば、この項を0にする。吸収項の識別には性質2を適用する。

ステップ 3-3 ステップ 3-1, ステップ 3-2 の操作を、3値カルノー図の出力が1となるすべての項についてくり返す。

上記のステップ 3-2 において、3値カルノー図 U 上に吸収項が存在するか否かの判別に、性質2を適用するとしている。これにより判別処理時間がやはり大幅に短縮できる。

ステップ 1-3 の必須項、非必須項の登録は、次のように定式化できる。

[アルゴリズム 4] 必須項、非必須項の登録

ステップ 4-1 3値カルノー図 U より、出力が1となる項を1つとる。

ステップ 4-2 この項を除いた3値カルノー図よりカルノー図 $K2$ を求める。この処理の詳細はアルゴリズム 5 に記す。

ステップ 4-3 カルノー図 $K2$ と元のカルノー図 $K1$ を比較する。 $K1$, $K2$ の内容が不一致ならば必須項、一致すれば非必須項と判定できる。必須項を表 W に、非必須項を表 Q に登録する。

上記のアルゴリズム 4 において、3値カルノー図 U からカルノー図 $K2$ を求める手続き

は次のようになる。

[アルゴリズム 5] $U \rightarrow K2$

ステップ 5-1 カルノー図 $K2$ 上のすべての項を 0 にする。

ステップ 5-2 カルノー図 $K2$ 上の項を 1 つとする。

ステップ 5-3 この項を被吸収項とする 3 値カルノー図上のすべての吸収項について 0 か 1 か調べる。吸収項が 1 つでも 1 ならばこの項を 1 にする。吸収項の識別には性質 3 を適用する。
ステップ 5-4 カルノー図 $K2$ 上のすべての項について、ステップ 5-2、ステップ 5-3 をくり返す。

なお、必須項、非必須項の登録表としては、3 値カルノー図を用いることも可能であるが、通常の一連番号表を用いる方が空白が少なくなり処理が効率的となる。表は、2 次元配列、 $W(I, J)$ 、 $Q(I, J)$ を使い、変数 I で一連番号を表わし、変数 J で A, B, C, \dots を表わすとよい。

ステップ 1-4 の連鎖の取り出しは以下のように行う。連鎖が単連鎖のときはそのまま、単連鎖でないときは単連鎖に分割して取り出す。

単連鎖の両端の項として、次のようなケースが有りうる。

- ① 端末項
- ② 分岐項
- ③ 端末項・分岐項ともになし

② の分岐項については、最初の単連鎖では、3 個以上の連結項を持つという条件で判定する。この場合、分岐点を求めこれを分岐点登録表に登録する。2 番目以降の単連鎖では、この分岐点に連結する項を分岐項と判定する。

③ のケースは閉連鎖の場合に生ずる。この場合は、単連鎖の中のどれか 1 つの項を端の項とする。

単連鎖の取り出しは、まず一方の端の項を取り出し、以下この項に連結する 2 番目の項、2 番目の項に連結する 3 番目の項、…、と順次連結する項を取り出し、もう一方の端の項に行き着くまで続ける。以上の操作は次のように定式化

できる。

[アルゴリズム 6] 単連鎖取り出し

ステップ 6-1 単連鎖の先端の項を取り出し、連鎖登録表 R に登録する。すなわち、次の操作を行う。

① 必須項登録表 W 、または、分岐点登録表 BR より項を 1 つとり、これに連結する項を非必須項登録表 Q より 1 個求め、連鎖登録表 R に登録すると同時に表 Q から削除する。

② 表 W のすべての項および表 BR のすべての点について、表 Q 内に連結する項がなく、かつ、表 Q に項があるときは閉連鎖が形成されている。この場合は、表 Q から一つの項をとり、表 R に登録すると同時に表 Q から削除する。

ステップ 6-2 表 R にある項に連結する項を表 Q より取り出し、表 R に追加する。

ステップ 6-3 ステップ 6-2 を、連鎖の後端が検出されるまでくり返す。後端の判定は次のように行う。

① 連結する項が表 W 、または表 BR にあるとき。

② 連結する項が表 Q に 2 個以上あるとき。この場合、分岐点を求めて表 BR に登録する。

③ 連結する項が表 Q がないとき。

ステップ 1-5 の連鎖の最小被覆を求める処理は以下のように行う。ただし、連鎖は単連鎖に分割されて取り出されるとする。

まず、性質 5～性質 7 より、単連鎖の最小被覆を求めるには、アルゴリズム 6 の単連鎖の取り出しの際、単連鎖を構成する項を互いに連結することのない項で構成されるグループに 2 つに分けておき、単連鎖の取り出しを終了したとき、項数の少ないグループ、項数が等しい場合は記号数の少ないグループを解とするとよい。また、閉連鎖の場合は、性質 8、性質 9 より、どちらのグループを解としてもよい。

以上の手続きは、次のように定式化できる。

[アルゴリズム 7] 最小被覆探索

ステップ 7-1 非必須項登録表 Q より先端の項を取り出し、グループ 1 の登録表 A に登録す

る。

ステップ 7-2 表 A にある項と連結する項を表 Q より取り出し、グループ 2 の登録表 B に登録する。

ステップ 7-3 表 B にある項と連結する項を表 Q より取り出し、表 A に追加登録する。

ステップ 7-4 ステップ 7-2、ステップ 7-3 の操作を、連鎖の後端が検出されるまで繰り返す。

ステップ 7-5 単連鎖の項数が奇数のときは、表 A の項数 > 表 B の項数、であるから、表 B を単連鎖の最小被覆とする。

単連鎖の項数が偶数のときは、表 A の項数 = 表 B の項数、となる。この場合は、端末項と分岐項の記号数の少ない方の表を最小被覆とする。記号数も等しいときは、分岐点を含む方の表を最小被覆とする。

閉連鎖の場合は表 A、表 B のいずれを最小被覆としてもよい。

最小被覆とならない表の項は冗長であるから、3

値カルノー図から削除する。

なお、ステップ 1-5 における特別な処理として、登録された分岐点の中で、いずれの単連鎖によっても被覆されない点がある場合は、その点と連結する項の一つを 3 値カルノー図に付加する。

図 5(a)～図 5(f) に、提案したアルゴリズムに基づく簡化の例を示している。(a) は、簡化の対象となるブール関数を 2 値のカルノー図に示している。(d) は非必須項を示しているが、この 2 つの項は連鎖をなしている。(e) 連鎖を構成する項の数が偶数なので、どちらの項を最小被覆としてもよい。本例では、 $A\bar{B}$ を選

	A	A	\bar{A}	\bar{A}
B	1		1	
\bar{B}	1	1	1	
	\bar{C}	C	C	\bar{C}

図 5(a) 簡化対象

	\bar{A}	A	A_n	\bar{A}	A	A_n	\bar{A}	A	A_n
\bar{B}		1		1	1	1		1	
B		1		1					
B_n		1		1					
	\bar{C}	\bar{C}	\bar{C}	C	C	C	C_n	C_n	C_n

図 5(b) すべての頂発生結果

	\bar{A}	A	A_n	\bar{A}	A	A_n	\bar{A}	A	A_n
\bar{B}						1		1	
B									
B_n		1		1					
	\bar{C}	\bar{C}	\bar{C}	C	C	C	C_n	C_n	C_n

図 5(c) 被吸収項削除結果

	\bar{A}	A	A_n	\bar{A}	A	A_n	\bar{A}	A	A_n
\bar{B}						1		1	
B									
B_n									
	\bar{C}	\bar{C}	\bar{C}	C	C	C	C_n	C_n	C_n

図 5(d) 非必須項登録結果

	\bar{A}	A	An	\bar{A}	A	An	\bar{A}	A	An
\bar{B}								1	
B									
Bn									
	\bar{C}	\bar{C}	\bar{C}	C	C	C	Cn	Cn	Cn

図 5(e) 最小被覆を求めた結果

	\bar{A}	A	An	\bar{A}	A	An	\bar{A}	A	An
\bar{B}								1	
B									
Bn		1		1					
	\bar{C}	\bar{C}	\bar{C}	C	C	C	Cn	Cn	Cn

図 5(f) 被吸収項削除結果から最小被覆に含まれない項を削除（最終解）

んでいる。

4. 従来のアルゴリズムとの比較

4.1 プログラム作成

提案したアルゴリズムに基づきプログラム作成を行った。言語は BASIC 言語を用いた。変数の数 P は最大で 5 個とした。

プログラム作成において重要となるいくつかの点を以下に示す。

まず、アルゴリズム 2 (すべての項の算出) のステップ 2-3 (吸収項の削除), すなわち, 性質 3 の適用は, 次のようなプログラムに記述できる。

[ルーチン 1] 吸収項の削除

```

10 FOR E=0 TO 1
20 FOR D=0 TO 1
30 FOR C=0 TO 1
40 FOR B=0 TO 1
50 FOR A=0 TO 1
60 IF K (A, B, C, D, E) < > 0 THEN
    GOTO 380
70 U (A, B, C, D, 2)=0
80 U (A, B, C, 2, E)=0
90 U (A, B, 2, D, E)=0

```

.....

```

370 U (2, 2, 2, 2, 2)=0
380 1F A+B+C+D+E=P THEN
    GOTO 400
390 NEXT A, B, C, D, E
400 END

```

アルゴリズム 3 (被吸収項の削除) のステップ 3-2 (吸収項の識別), すなわち, 性質 2 の適用は次のようなプログラムに記述できる。

[ルーチン 2] 吸収項の識別

```

10 FOR E=0 TO 2
20 FOR D=0 TO 2
30 FOR C=0 TO 2
40 FOR B=0 TO 2
50 FOR A=0 TO 2
60 IF U (A, B, C, D, E)=0 THEN GOTO
    380
70 IF U (A, B, C, D, 2)=1 THEN U (A,
    B, C, D, E)=0
80 IF U (A, B, C, 2, E)=1 THEN U (A, B,
    C, D, E)=0
90 IF U (A, B, 2, D, E)=1 THEN U (A, B,
    C, D, E)=0

```

.....


```

370  IF U (2, 2, 2, 2)= 1 THEN U (A, B,
      C, D, E)=0
380  IF  A+B+C+D+E=2 * P  THEN
      GOTO 400
390  NEXT A, B, C, D, E
400  END

```

完成したプログラムのステップ数は441ステップとなった。

4.2 クワイン-マクラスキ法による簡化

本論文で提案したアルゴリズムを従来のアルゴリズムと比較する。従来のアルゴリズムとしては、よく知られているクワイン-マクラスキ法をとる。

クワイン-マクラスキ法の簡化手順は次のようになる³⁾⁻⁵⁾。

[アルゴリズム 8] 簡化アルゴリズム

ステップ 8-1 主項を求める。

ステップ 8-2 主項の最小被覆を求める。

ステップ 8-3 サイクリックテーブルがあるときはその最小被覆を求める。

上の記述において、主項とは、真理値表より生じ得るすべての項の中から被吸収項を除いた項を指している。また、サイクリックテーブルとは、本研究の用語でいう閉連鎖に対応している。アルゴリズムの詳細は付録に示している⁵⁾。

本研究のアルゴリズムとクワイン-マクラスキ法の処理の対応を示すと次のようになる。

本研究の方法 クワイン-マクラスキ法

ステップ 1-1 } → ステップ 8-1
ステップ 1-2 }

ステップ 1-3 }
ステップ 1-4 } → { ステップ 8-2
ステップ 1-5 } { ステップ 8-3
ステップ 1-6 }

以下、ステップ 1-1～ステップ 1-2を主項作成処理、ステップ 1-3 以下を最小被覆処理とよぶことにする。

付録に示したアルゴリズムに基づきブール関数簡化プログラムを作成した。言語はBASICを用い。変数の数Pは最大で5とした。完成したプログラムのステップ数は310ステップとなった。

4.3 メモリ使用量の比較

表1に、メモリ使用量の比較を示している。メモリ使用量は、変数・配列領域とテキスト領域に分けて示している。

本表より次が知られる。

① 提案した方法はテキスト領域、クワイン-マクラスキ法は変数・配列領域を多く必要とする。

② 全メモリ使用量は、提案した方法がクワイン-マクラスキ法より30%少ない。

以上の結果は、クワイン-マクラスキ法では、キューブの併合と主項表の作成に大きな配列を必要とするためと考えられる。

4.4 処理時間の比較

表2に、処理時間の比較を示している。処理時間は、主項作成処理に要する時間と最小被覆処理の時間に分けて表わしている。テストデータは、変数の数Pとパターンを変えている。データの1の数とは、カルノー図に表わしたときの1の数を表わしている。また、処理時間は、クロック周波数20MHzの32ビットパソコンを用い、BASICインタプリタ動作で行ったとき得られた値を示している。この値は、BASICコンパイラを用いれば、約1/4に低下することが確認されている。

表1 メモリ使用量の比較

	本研究の方法 (バイト)	クワイン- マクラスキ法 (バイト)
変数・配列領域	9221	25464
テキスト領域	16220	8405
合 計	25441	33869

表2 処理時間の比較

テストデータ			本研究の方法 (秒)			クワインーマクラス スキ法 (秒)		
P	パターン	1の数	主項作成	最小被覆	合計	主項作成	最小被覆	合計
3	(0, 0, 0) のみ 1	1	1	1	2	1	—	1
3	(0, 0, 0) のみ 0	7	1	1	2	—	1	1
3	単連鎖	5	1	1	2	—	1	1
3	閉連鎖	6	1	2	3	—	2	2
4	(0, 0, 0, 0) のみ 1	1	1	1	2	1	—	1
4	(0, 0, 0, 0) のみ 0	15	4	2	6	5	1	6
4	単連鎖	7	2	4	6	1	1	2
4	閉連鎖	8	2	5	7	2	5	7
5	(0, 0, 0, 0, 0) のみ 1	1	4	3	7	1	1	2
5	(0, 0, 0, 0, 0) のみ 0	31	13	4	17	44	4	48
5	単連鎖	13	4	11	15	2	9	11
5	閉連鎖	14	4	13	17	2	31	33

—は0秒以下

表2より、次が知られる。

① 主項作成時間については、両方法ともに、Pの値、または、1の数が多くなるとともに急速に増大する。増大の程度は、クワインーマクラススキ法の方が大きい。

② 最小被覆処理時間については、両方法ともに、連鎖の長さが長くなるとともに急速に増大する。クワインーマクラススキ法ではとくに閉連鎖の有ると、Pの値とともに急速に増大する。

③ 合計処理時間について、最も長い場合をとると、提案した方法の処理時間は、クワインーマクラススキ法の約1/3となる。

④ コンパイラ動作をとると、提案した方法では、処理時間を最大でも4秒程度にすることが可能であり、CAI用としての所期の目標を達成している。

4.5 処理時間に関する考察

両アルゴリズムについて、計算の複雑さの観点から考察を行う。

本研究で提案したアルゴリズムの処理時間TAは次式で表わされる。

$$TA = \text{全項発生時間} + \text{被吸収項除去時間} + \text{非必須項登録時間} + \text{連鎖処理時間}$$

各処理時間のオーダーは、1項当たりの処理時間で規格化すると、次のようになる。ただし、Pは変数の個数を表わしている。

全項発生時間 $\sim \alpha 2^P$

1項だけ1のとき: $\alpha = 2^P$

1項だけ0のとき: $\alpha = 1$

被吸収項除去時間 $\sim \beta 2^P$

1項だけ1のとき: $\beta = 1$

1項だけ0のとき: $\beta = 3^P$

非必須項登録時間 $\sim \gamma 2^P$

1項だけ1のとき: $\gamma = 1$

半分1で相互に隣接しないとき: $\gamma = 2^{P-1}$

1項だけ0のとき: $\gamma = 1$

連鎖処理時間 $\sim \delta^2$

1 項だけ 1 のとき: $\delta=1$

半分 1 で相互に隣接していると

き: $\delta=2^{p-1}$

1 項だけ 0 のとき: $\delta=1$

上記の 4 つの処理の中で、処理時間の P に対する依存性が最も大きいものを求めると、被吸収項除去となる。その処理時間は、カルノー図がオール 1 のとき、 6^p のオーダーとなる。これは、変数が 1 個増えるごとに処理時間が 6 倍になることを意味している。

被吸収項除去に対応するクワイン—マクラスキ法上の処理は、主項作成中のキューブの併合処理、すなわち、付録に示したアルゴリズム 9 のステップ 9-4～ステップ 9-5 となる。この処理においては、キューブの併合ができるか否かの判別を、1 の個数が 1 個異なる項同士の組み合わせすべてについて、くり返し行うことになる。項の組み合わせは変数の個数 P の増加とともに急速に増加し、処理時間も増大する。表 3 に、プログラムを実行して求めたキューブ併合処理における判別の回数を示している。表より次式が得られる。

キューブ併合処理時間 $\sim 10^p$

上記のように、本研究の方法で最も処理時間の P に対する依存性の大きい処理は被吸収項除去であり、その時間は 6^p のオーダーとなる。これに対応するクワイン—マクラスキ法のキューブの併合処理は 10^p のオーダーとなる。この観点からも本研究で提案した方法が優れていることが裏付けられる。

表 3 クワイン—マクラスキ法のキューブ併合判別回数

変数の数	デ ー タ	併合判別回数
2	(0, 0) のみ 0	2
3	(0, 0, 0) のみ 0	30
4	(0, 0, 0, 0) のみ 0	316
5	(0, 0, 0, 0, 0) のみ 0	3029

5. む す び

本論文では、論理回路設計演習用 CAI に適用するので論理変数の数は 5 以下に限られるという条件のもとで、新しいブール関数簡化アルゴリズムを提案した。提案したアルゴリズムの特徴は次のとおりである。

① 3 値カルノー図を導入している。

② 3 値カルノー図に、与えられた関数から得られるすべての項を一挙に作成し、ついで、冗長な項を種類ごとに順次削除していく。

③ 非必須項の最小被覆を項の連鎖をたどることにより求める。

提案したアルゴリズムと従来のアルゴリズムについて、プログラム作成を行って処理時間の比較を行った。その結果、処理時間が最も長くなる場合で、提案したアルゴリズムによる処理時間が、従来のアルゴリズムの約 1/3 になることが分かった。

なお、本論文では、変数の数は 5 以下としているが、提案したアルゴリズムは、変数の数 6 以上に拡張できる可能性があり今後検討したい。また、本論文では、禁止無しとしているが、禁止有りの場合についても検討中であり、別途報告する予定である。

参 考 文 献

- 1) フィスタ, 尾崎訳, “デジタル計算機の論理設計”, 朝倉書店, (1960)
- 2) F.J. Hill & G.R. Peterson, “Introduction to Switching Theory & Logical Design”, John Wiley & Sons, Inc., (1968)
- 3) 足立暁生, “論理設計の基礎”, 東海大学出版会, (1973)
- 4) 吉田典司, “論理数学 II”, 共立出版, (1978)
- 5) 向殿, 笹尾, “スイッチング理論演習”, 朝倉書店, (1984)
- 6) 苦米地宣裕, “コンピュータを用いた論理式簡化の一方法”, 1993 年度電子情報通信学会秋期全国大会論文集, 6-8, (1993)
- 7) 苦米地宣裕, “コンピュータを用いたブール関数簡化の一方法”, 電子情報通信学会技術報

告, Vol. COMP93-52, pp.19-18, (1993)

付 録

クワインーマクラスキ法による ブール関数簡単化アルゴリズム

クワインーマクラスキ法は以下のように定式化されている。なお、以下の記述において、キューブとは項の表現法の1つで、3値カルノー図と同様、変数が偽のとき0, 真のとき1, 恒等的に1のとき2と表すものである。

[アルゴリズム9] 簡単化アルゴリズム
(主項を求める)

ステップ9-1 真理値表から、最小項とその10進表現、および、キューブ表現を求める。

ステップ9-2 キューブを1の個数の少ない順に並べ換えたリストを作る。

ステップ9-3 キューブを1の個数に従ってグループに分割し、第1番目のリストを作る。 $1 \leftarrow 1$ とする。

ステップ9-4 第 I 番目のリストの第 J グループのキューブと第 $(J+1)$ グループの間で比較を行い、2つのキューブ間でただ1つの変数のみ異なっているような対があれば、第 $(I+1)$ 番目リストの第 J グループ欄に2つのキューブに包含される最小項の10進表現とキューブを併合して得られるキューブを書く。

ステップ9-5 ステップ9-3, ステップ9-4を、第 I 番目のリストにグループがただ1つになるまでくり返す。

ステップ9-3～ステップ9-5の間にキューブの併合を行わなかった項を主項として登録する。

(主項の最小被覆を求める)

ステップ9-6 主項を垂直軸に沿って記号数の少

ない順に並べ、主項に包含される最小項の10進表現を水平軸に沿って並べた表を作成する。行 i に対応する主項が列 j に対応する最小項を包含するとき、交点 (i, j) に×印を記入する。

ステップ9-7 ×印をただ1個しか持たない列を求める。この列に×印を持つ行を解に含める。この列と行を除去する。

ステップ9-8 行 Q が×印を持つ列に行 P も×印を持つとき、行 Q を除去する。

ステップ9-9 列 j が×印を持つ行に列 i も×印を持つとき、列 i を除去する。

ステップ9-10 ステップ9-7～ステップ9-9を、×印をただ1個しか持たない列がなくなるまでくり返す。

(サイクリックテーブルの最小被覆を求める)

ステップ9-11 上述までの手続きを用いてもすべての列が除去できないときはサイクリックテーブルとよばれる。以下その処理を行う。まず、1つの行をとる。

ステップ9-12 この行が最終解に含まれると仮定し、解に仮登録の上削除する。

ステップ9-13 ステップ9-7～ステップ9-10をくり返す。

ステップ9-12, ステップ9-13の手続きを処置Aとする。

ステップ9-14 ステップ9-11で選んだ行が最終解に含まれないと仮定して単に削除する。

ステップ9-15 ステップ9-7～ステップ9-10をくり返す。

ステップ9-14, ステップ9-15の手続きを処置Bとする。

ステップ9-16 処置Aと処置Bの結果を比較し、項数の少ない方を解とする。

解は、ステップ9-7, および、ステップ9-16で得られた項の和となる。