# 隣接グループ生成法と分割法による論理関数の簡単化の一手法

後　藤　公　雄

# A Simplification Method of Logic Functions by Using both Adjacent groups Generation Method and Division One

Kimio GOTO

**Abstract**

In this paper, one method for simplification of logical functions is proposed. This method uses the algorithm of logical-variables-division method which contains also that of adjacent-minterms-gronp-generation one. In the latter algorithm, lower groups of some adjacent minterms construct up upper ones of larger adjacent minterms, these operations are repeated in turn, and after unnecessary lower gronps are deleted, only remained groups become prime impliconts. In the former algorithm, the original logical function with $n$ variables, is divided into $2^{n-nd}$ ones with $n_d$ variables. Nextly, the adjacent-minterms-group-generation method is applied to each of these divided function, and they are simplified respectively within themselves. At last, $2^{n-nd}$ divided functions after simplification are unified to the original function. FORTRAN programs were drawn up for these two algorithms and run. Running Times of two programs were compared with each other about several variable numbers and Truth-table densities. As a result, this method was proven to be better than the adjacent-minterms-group-generation one.

## 1. 緒　　言

論理関数の簡単化の手順は,

(1)　主項の作成,

(2)　得られた主項から最小数の主項を求める,

という2つのステップに分かれる。(1)の方法としては, Quine McClusky 法, 節展開法[1], 部分マップ法[2]などがあり, 当研究室でも隣接最小項グループ作成法[2),4)]を考案している。この方法は与えられた関数の隣接する最小項のグループを, 最小項の個数が2個, 4個, 8個, …, となるように順次成長させて行く方法である。本論文では, さらに多変数の関数が与えられた場合, これをいくつかのより少ない変数の関数に分割して, それぞれについて隣接最小項グループ作成法を用いて主項を求め, これらを組み合せて元の関数の主項を求める分割法について報告する[5]。

昭和63年9月28日受理

## 2. 基本アルゴリズムおよび計算例

### 2.1 隣接最小項グループ作成法

各項目に分けてつぎに説明する。

(1)　2個隣接グループの作成

$n$ 変数の論理関数について, 最小項 $m_i$ に隣接する最小項 $m_x$ を作成する。そのため $i$ の2進数表示の $j$ 番目のビット($j=0, 1, \cdots, n-1$)を調べ, 1となるビットは変えず, 0のビット1個所だけを変えることにより新たに作られる2進数の10進表示を $x$ とし, $m_i$ に隣接する最小項 $m_x$ を作成する。このようにして作成した3変数の2個の最小項による隣接グループの要素の表（$m_i$ 対 $m_x$）を表1に示す。

(2)　$2^l$ 個隣接グループの作成

いま, $2^{l-1}$ 個の最小項より成る隣接グループ($2^{l-1}$ 個隣接グループ) 2個を $P_{1,\,l-1}$ と $P_{2,\,l-1}$ で表し, これらが $2^{l-2}$ 個隣接グループ $P_{1,\,l-2}$ を共通に含んでおり, さらにそれぞれ残る $2^{l-2}$ 個隣接グループ $P_{2,\,l-2}$ と $P_{3,\,l-2}$ を含んでいるものとする。このとき,

表1．3変数にたいする隣接最小項
Table 1. Adjacent minterms about their variables

| | $m_x$ | | |
|---|---|---|---|
| $m_i=(ABC)$　$2^j\longrightarrow$ | $2^2$ | $2^1$ | $2^0$ |
| 0=(000) | 4 | 2 | 1 |
| 1=(001) | 5 | 3 | —— |
| 2=(010) | 6 | —— | 3 |
| 3=(011) | 7 | —— | —— |
| 4=(100) | —— | 6 | 5 |
| 5=(101) | —— | 7 | —— |
| 6=(110) | —— | —— | 7 |
| 7=(111) | —— | —— | —— |

$$P_{y,\,l-2}=P_{3,\,l-2}+(P_{2,\,l-2}-P_{1,\,l-2}) \qquad (1)$$

となるような $2^{l-2}$ 個隣接グループ $P_{y,\,l-2}$ が得られたものとする。ただし，式(1)の＋，－演算はこの式の右辺の集合の各要素間の ＋，－ 演算とする。このような $P_{y,\,l-2}$ が存在すれば明らかに

$$P_{1,\,l}=P_{1,\,l-2}\cup P_{2,\,l-2}\cup P_{3,\,l-2}\cup P_{y,\,l-2} \qquad (2)$$

で表される $2^l$ 個隣接グループが存在する。このようにして2個の $2^{l-1}$ 個隣接グループから1個の $2^l$ 個隣接グループが得られる。この隣接グループ作成の説明図を図1に示す。

(3)　下位隣接グループの消去

$2^l$ 個隣接グループを作成したとき，それに含まれる $2^{l-1}$ 個隣接グループは $l\times 2$ 個存在し，これらは，元の $2^l$ 個隣接グループから $2^k$ 個ずつ連続して，$2^k$ 個おきに抽出し，それらを連接させて作った $2^{l-1}$ 個隣接グループそのものである。ここで，$k=0,\ 1,\ 2,\ \cdots,\ l-1$ である。このような $2^{l-1}$ 個隣接グループはすでに作成した $2^l$ 個隣接グループに含まれるからすべて削除されねばならない。たとえば，8個隣接グループ {0, 1, 4, 5, 8, 9, 12, 13} が存在すれば，3×2個の4個隣接グループ，すなわち {0, 1, 4, 5}, {8, 9, 12, 13}, {0, 1, 8, 9}, {4, 5, 12, 13}, {0, 4, 8, 12} および {1, 5, 9, 13} はすべて除去される。

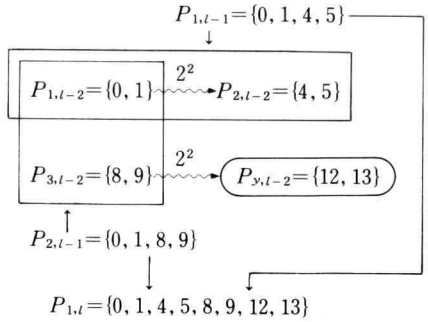


図1．$2^l$ 個隣接最小項グループの生成

Fig. 1. Generation of adjacent group consisting of $2^l$ minterms

### 2.2　分割法のアルゴリズムと計算例

図2のフローチャートに従って分割法のアルゴリズムをつぎに述べる。

［ステップ1］　与えられた $n$ 変数の関数を $n_d$（$<n$）変数の関数で $2^{n-n_d}$ 個に分割する。分割された各関数を分割関数と呼ぶ。

［ステップ2］　$n_d$ 変数の分割関数 $2^{n-n_d}$ 個それぞれに，隣接最小項グループ生成法を適用して各分割関数内でのすべての隣接最小項グループとそれらの包含関係を求める。

［ステップ3］　隣接する分割関数相互間の対応するセルの中で，ともに1を持つセルが少なくとも1個存在すれば，ともに1を持つセルに1を持つ合成関数を作る。さらにこのような合成関数内ですべての隣接グループとその包含関係を求める。

［ステップ4］　合成関数同士で対応するセル全体を調べ，ともに1を持つセルが少なくとも1個存在すれば，ステップ3と同様にともに1を持つセルと同じセルに1を持つ上位合成関数を作る。さらにこのような合成関数内ですべての隣接グループとその包含関係を求める。

［ステップ5］　上位と下位の隣接グループ相互間の包含関係を調べ，上位に含まれる下位の隣接グループはすべて除去する。

［ステップ6］　ステップ4で求まった上位合成関数相互間を調べ，相対応するセル間に隣接するものが一つもないか，またはこの合成関数が最上位のものであれば，ステップ7に移る。そうでなければステップ4に戻る。

［ステップ7］　求まったすべての隣接グループを元の関数の主項に変換する。

つぎに以上に述べたアルゴリズムを用いた計算例を示す。いま，4変数の関数 $f$ が

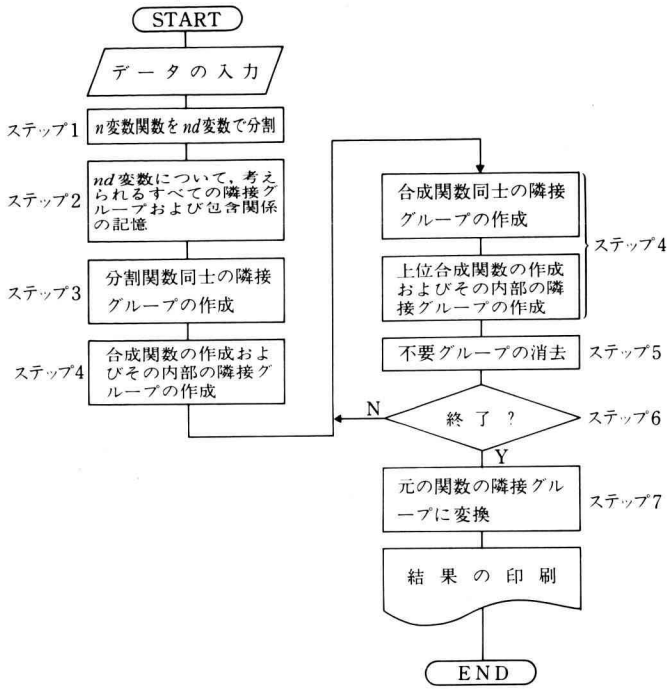


図 2.　分割法のフローチャート

Fig. 2.　Flow chart of division method

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 1 | 1 |
| 01 | 1 | 1 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 1 | 0 | 0 | 1 |
| | ↑ $G_0$ | ↑ $G_1$ | ↑ $G_2$ | ↑ $G_3$ |

図 3.　関数 $f=\sum(0, 1, 2, 4, 5, 8, 10, 12)$ の分割

Fig. 3.　Division of function $f=\sum(0, 1, 2, 4, 5, 8, 10, 12)$

$$f=\sum(0, 1, 2, 4, 5, 8, 10, 12) \qquad (3)$$

で与えられるものとする。ステップ 1 としてこの関数 $f$ を 2 変数の関数 $2^{4-2}=4$ 個に分割し，これらを $G_0$, $G_1$, $G_2$, $G_3$ とすると，図 3 に示すように，

$$G_0=\sum(0, 1, 2) \qquad (4a)$$
$$G_1=\sum(0, 1) \qquad (4b)$$
$$G_2=\sum(0, 2) \qquad (4c)$$
$$G_3=\sum(0) \qquad (4d)$$

が得られる。ステップ 2 として，これらの各分割関数 $G_0$, $G_1$, $G_2$ および $G_3$ について隣接グループを求めると，それぞれ ((0, 1), (0, 2)), (0, 1), (0, 2) および (0) が得られる。また，これらのグループと下位の最小項との包含関係は，これらがそれぞれ ((0, 1), (0, 2)), (4, 5), (8, 10) および (12) に相当することより明白である。ステップ 3 として隣接する分割関数相互間，すなわち $G_0$ と $G_1$, $G_0$ と $G_2$, $G_1$ と $G_3$, $G_2$ と $G_3$ の間で，少なくとも 1 個の対応セルにともに 1 が存在する。そこで対応するセルにともに 1 を持った合成関数 $G_{0,1}$, $G_{0,2}$, $G_{1,3}$ および $G_{2,3}$ を作ると，図 4 のようになり，

$$G_{0,1}=\sum(0, 1) \qquad (5a)$$
$$G_{0,2}=\sum(0, 2) \qquad (5b)$$
$$G_{1,3}=\sum(0) \qquad (5c)$$
$$G_{2,3}=\sum(0) \qquad (5d)$$

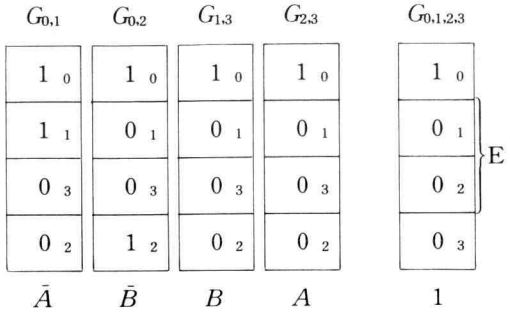


図 4. 合成関数の作成

Fig. 4. Generation of synthesizing function

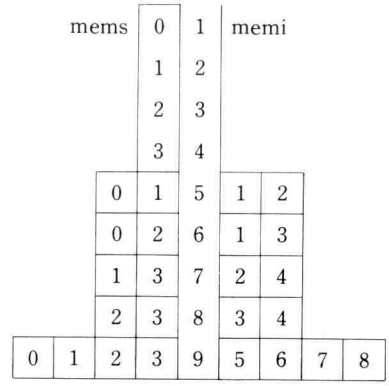


図 5. 2 変数の主項グループと包含関係

Fig. 5. Group of prime implicant about two variables and those inclusive relation

が得られる。これらは，それぞれ元の関数の隣接グループとしては，(0, 1, 4, 5), (0, 2, 8, 10), (4, 12) および (8, 12) に相当する。さらにステップ 4 として，$G_{0,1}$～$G_{2,3}$ の上位合成関数として $G_{0,1,2,3}$ が考えられ，

$$G_{0,1,2,3}=\sum(0) \tag{6}$$

が求められる。これは元の関数の隣接グループで表すと (0, 4, 8, 12) に相当する。ステップ 5 として上位隣接グループで下位隣接グループを除去することができる。ステップ 6 としてこれ以上合成関数は作り得ないから，ステップ 7 として求まった隣接グループを最初の関数の主項となる隣接最小項グループで書き直すと，(0, 1, 4, 5), (0, 2, 8, 10) および (0, 4, 8, 12) となる。なお，合成関数から最小項の論理和への書直しはつぎのようにして行われる。すなわち，セル番号 $i$ が 1 となる合成関数が，$n_d$ 個の変数より成り，しかも

$$G_{k_1,\ k_2,\ \cdots,\ k_j,\ \cdots,\ k_l} \tag{7}$$

のように $l$ 個の添字 $k_1, k_2, \cdots, k_j, \cdots, k_l$ を持った関数で表現されるとき，

$$\left.\begin{aligned} a_j&=\mathrm{Shift}\,[k_j,\ n_d]+i \\ &\text{ただし}\quad j=1,\ 2,\ \cdots,\ l \end{aligned}\right\} \tag{8}$$

で表わされる元の関数のすべてのセル番号 $a_i$ を採用すればよい。ただし，式 (8) の関数 Shift $[k_j, n_d]$ は 10 進数 $k_j$ を 2 進数に変換し，これを左へ $n_d$ ビットだけシフトさせたときの 2 進数の 10 進数表示とする。

## 3. プログラムと実行結果

隣接最小項グループ生成法では，隣接グループの要素をすべて記憶し，その都度，それに含まれる下位隣接グループを除去している。これにたいし，分割法では，始めに図 5 のように 2 次元配列 mems ($no$) と memi ($no$) を作り，分割する $n_d$ 変数の関数について考えられるすべての隣接グループの要素を，$no$ が 1 から 9 (一般には記憶すべき隣接最小項グループの全数) までの配列 mems ($no$) に記憶し，memi($nu$) には，$no=nu$ となるような mems ($no$) に記憶されている隣接最小項の添字番号がさらに下位の mems ($no$) に格納されているものについて，その下位の mems ($no$) の番号を記憶するようにする。これにより，下位の隣接最小項グループの除去が容易となる。

以上の 2 つの手法について，FORTRAN77 を言語としてプログラムを作成し (ステップ数はそれぞれ 200 および 300 ステップ)，M170F (富士通) を用いて計算を実行させた。最小項を乱数を用いて発生させ，真理表濃度と CPU 時間を求めた。この CPU 時間には，同一真理値表濃度を 20 回乱数発生させ，その都度時間を測定して平均をとったものを用いた。この結果を図 6 と 7 に示す。これらの図より，真理値表濃度が低い場合は，隣接最小項グループ生成法の方が速く，高濃度になると分割法の方が速くなり，また，変数が多くなるにしたがって分割法の方が有利になることがわかる。なお，分割法では 9 変数で濃度 0.9 のとき演算時間は約 1.5 秒となった。

また，図 8 には 11 変数の場合について，節展開法[1]や部分マップ法[2]にたいし，真理値表濃度と CPU 時間の関係を比較したものを示す。使用した計算機の演算

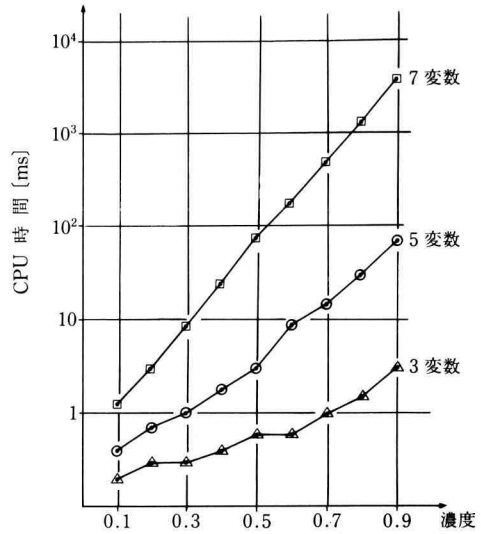


図 6.　濃度と実行時間（隣接最小項グループ法）

Fig. 6.　Truth table density and CPU time for adjacent group method

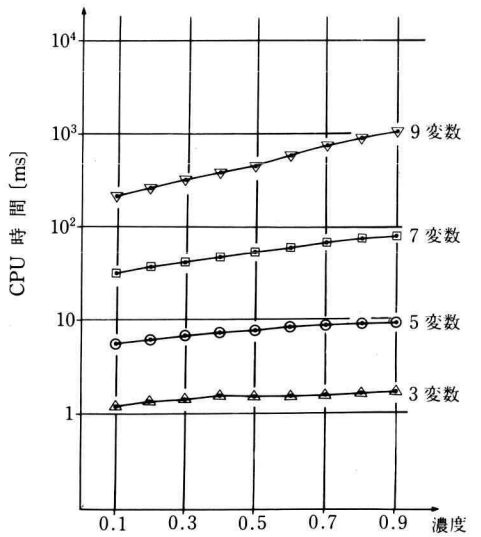


図 7.　濃度と実行時間（分割法）

Fig. 7.　Truth table density and CPU time for division method

速度によって必ずしも即断はできないが，性能上では節展開法に近く，部分マップ法には及ばないことがわかる。なお，ここで示した節展開法と部分マップ法の結果は名古屋大学の M200 を用いて計算された結果を参照した。

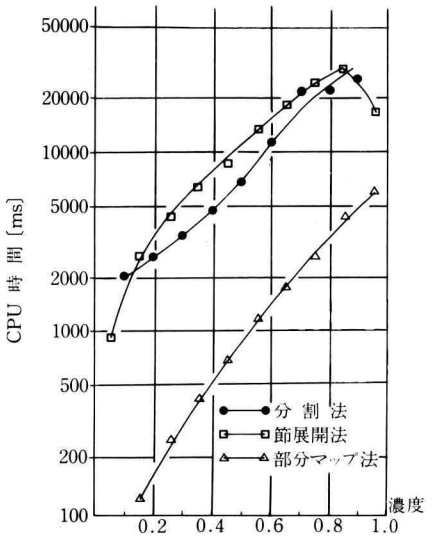


図 8.　真理値表濃度と計算時間の関係（他の論文の手法との比較，$N=11$）

Fig. 8.　Truth table density and CPU time (comparison between this paper and others, $N=11$)

## 4. 結　　言

本論文は，隣接最小項グループ生成法を，より多変数の関数に適応させるため，元の関数を分割する手法（分割法）を取り入れたものについて述べている。なお，この論文では特に述べないが分割変数の数 $N_D$ を真理値表濃度により変える手法を用いた。今後の研究課題として

(1)　このような分割変数の個数の検討

(2)　多重分割（複数回繰り返して分割する方法）の検討

(3)　演算時間の縮小と変数の増大

をとり上げたい。

終りに，本研究の推進に努力された昭和 61 年度卒業研究生小高浩三君に謝意を表する。

## 参 考 文 献

1)　上林，岡田，矢島："節展開法を用いた論理関数の主項の生成"，信学論 (D)，J62-D, pp. 89-96.

2) 松田, 宮腰："論理関数の主項を高速に導出する手法", 信学論(D), J67-D, No. 2, pp. 208-215.
3) 後藤："最小項添字よりなる隣接グループの予測による論理関数素項の計算手法について", 昭61, 電気学会全国大会, 1381.
4) 後藤："最小項の隣接グループ作成による論理関数の簡単化手法について", 昭61, 電気関係学会関西支部連合大会, G6-10.
5) 後藤他："隣接最小項グループ法とその分割法による論理関数の簡単化手法の比較", 昭62, 電気学会全国大会, 1443.