

論理式簡単化の一手法

一部分マップ法

宮腰 隆 松田秀雄

富山大学 工学部

あらまし 二値論理関数簡単化で有効だった一手法：部分マップ法を多値入力二値出力関数の簡単化へ拡張した。本方法は、(1)項の選出、(2)項の拡大といった基本手続きを繰り返し用いる発見的手法である。特に、部分マップという領域に限定するため、項の拡大が高速に実行できる。本稿では、部分マップ法による簡単化アルゴリズムを詳述したあと、算術関数やランダム関数による実験結果を示す。

A Minimization Algorithm for Logical
Expressions -Submap Method-

Takashi MIYAGOSHI and Hideo MATSUDA
Faculty of Engineering, Toyama University
3190, Gofuku, Toyama-shi, 930 Japan

Abstract We have extended the algorithm called submap method, which efficiently performed for minimization of Boolean functions with two-valued inputs, to Boolean functions with multiple-valued inputs. This method is a heuristic algorithm that iteratively employs fundamental subprocesses, such as (1) selecting of a product term and (2) expanding it. In especially, expanded product terms can be obtained fastly because the expanding operation is implemented only on the submap domain. In this paper, we describe the minimization algorithm of submap method in detail, then show experimental results of some arithmetic functions and some randomly generated functions.

1. はじめに

論理式簡化問題は、近年VLSI内で多用されるPLA (Programmable Logic Array)の設計問題に対応することから研究が活発に行なわれている[1]。論理式簡化問題では最簡形式を得ることが望ましいが、最簡形式を得るには多くの計算時間を必要とすること、又、大規模回路の取扱いを可能にしたいなどの理由から発見的手法を使った高速、高性能なアルゴリズムの開発が重要と考えられる。

本稿で提案する部分マップ法は二値論理関数簡化[2]で良好な結果を得たことを踏まえ、今回、ここに多値論理関数向きに拡張したものである。

発見的手法による多値論理関数簡化アルゴリズムは既に幾つもの提案されており、それらの代表例による性能比較もなされている[3]。それらの中で Besslichの方法[4]や Dueck and Millerの方法[5]は孤立最小項に着目し、その最小項を優先的に選出することから精度の良い解が得られる。ところが最小項に基づいているので変数が増えると最小項の数が爆発的に増大するため、実用的には10変数程度までの関数しか取り扱うことができないといわれている[6]。大変数まで実行できるアルゴリズムとしてはMINI [7]、MINI-II [8]及びESPRESSO-MV [9]などがある。これらの方法では、項の拡大を否定を使って効率よく行なうことができ高速、高性能な解が求まるが、与えられた関数の否定を生成する必要があるため多くの記憶容量を必要とする場合がある。

これらのことを考慮して、部分マップ法では項主体の演算とし、項の拡大もトートロジー判定[10]で効率化を図っている。

2. で幾つかの基本的事項を説明し、3. で部分マップ法による簡化アルゴリズムと例題を詳述する。最後に4. でベンチマーク算術関数とランダム関数により性能を評価する。

2. 基礎的事項

2.1 多値入力二値出力関数 [8]

写像 $f : P_1 \times P_2 \times \dots \times P_n \rightarrow B$, $P_i = \{0, 1, \dots, p_i - 1\}$, $B = \{0, 1\}$ を多値入力二値出力関数とよぶ。 X_i を変数とし、 $S_i \subseteq P_i$ とすると、リテラル $X_i^{S_i}$ を

$$X_i^{S_i} = \begin{cases} 1 & (X_i \in S_i \text{ のとき}) \\ 0 & (X_i \notin S_i \text{ のとき}) \end{cases}$$

と定義する。

リテラルの論理積 $X_1^{S_1} \cdot X_2^{S_2} \cdot \dots \cdot X_n^{S_n}$ が $X_1^{S_1} \cdot X_2^{S_2} \cdot \dots \cdot X_n^{S_n} \leq f$ のとき、項(内項)とよび t_j ($j=1, 2, \dots, m$)とすれば、関数 f は次のように表される。

$$f = t_1 \vee t_2 \vee \dots \vee t_m \quad (1)$$

項の体積を $\prod_{i=1}^n |S_i|$ で表した時、体積が1の項を最小項、最大な項を主項という。項の論理和で表現した論理式(1)は、一般に加法形式とよばれる。関数 f を表す加法形式は幾通りも考えられるので、そのうち項数ができるだけ少なくなるような一組の加法形式を求めることを、本稿では簡化とよぶことにする。

[例題2.1] 関数 f の一例を示す。

$$f = X_1^{\{0\}} \cdot X_2^{\{2\}} \vee X_1^{\{1\}} \cdot X_2^{\{1,2\}} \vee X_1^{\{2\}} \cdot X_2^{\{1\}}$$

ここで、 $f : P_1 \times P_2 \rightarrow B$, $P_1 = P_2 = \{0, 1, 2, 3\}$, $B = \{0, 1\}$ である。図1は f をマップ表現したもので、マップ上に記した数字{1, 2, 3}は、それぞれ f の第1項、第2項及び第3項を表し、 f はこれらの最小項で1を、それ以外では0となる関数である。

この例は次のように簡単になる(太線枠)。

$$f = X_1^{\{1,2\}} \cdot X_2^{\{1\}} \vee X_1^{\{0,1\}} \cdot X_2^{\{2\}}$$

2.2 部分マップ

(1) 隣接キューブ、共通キューブ 項 t_i のことをキューブ C_i ともよび、次のようにビット表現する ($n=2$, $P_i = \{0, 1, 2, 3\}$)。

$$X_1^{\{1\}} \cdot X_2^{\{1,2\}} \Rightarrow \begin{matrix} X_1 & X_2 \\ 0100 & - & 0110 \end{matrix}$$

キューブ C_i と周辺のキューブ C_j ($i \neq j$)との積(ピ

	X_2	0	1	2	3	
X_1	0			1		
	1		2	2		⊙: 第1項
	2		3			⊠: 第2項
	3					

図1 関数 f

ットごと) をとり、すべて0の変数部(=k)が一つあるなら、 C_1 と C_2 は変数 X_k で隣接しているという。又、すべて0の変数部がないとき、 C_1 と C_2 は共通部分を持つキューブであり、それぞれ隣接キューブ、共通キューブとよぶ。

[例題2.2] キューブ $C_1=0100-0110$ 、 $C_2=0111-0100$ 及び $C_3=0110-0001$ があると、各キューブ間の関係は、
 $C_1 \cap C_2 = 0100-0100$: 共通キューブ
 $C_1 \cap C_3 = 0100-0000$: 隣接キューブ (変数 X_2)
 $C_2 \cap C_3 = 0110-0000$: 隣接キューブ (変数 X_2)
 となる。(図2参照)。

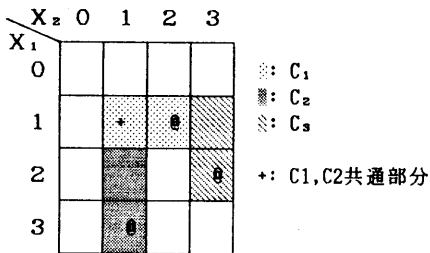


図2 隣接キューブ($C_1:C_3, C_2:C_3$)
 共通キューブ($C_1:C_2$)
 θ : 最大基本キューブ $C_{1,ma}$ ($i=1,2,3$)

(2) キューブの順序付け あるキューブが、もし、すべての変数部が一つだけ1、他は全部0の桁よりなれば、マップ上の一つの最小項を表すが、ここでは基本キューブともよぶ(例、0001-0001)。各基本キューブ(最小項)はそれらを表すビット表現による2進数の大きさで自然に順序付けられる(但し、左側より右側に桁が上がるとみる)。 $n=2$ 、 $P_1=\{0,1,2,3\}$ では、
 $1000-1000 < 0100-1000 < \dots < 0001-0001$

である。キューブ C_1 において、各変数部の最右端の1のみ残り、他の桁のビットをすべて0とおいてできる基本キューブ $C_{1,ma}$ は C_1 に含まれる基本キューブのうち自然の順序で比較して最大のものであり、最大基本キューブとよぶ。例えば、 $C_1=0100-0110$ として、 $C_{1,ma}=0100-0010$ である。従って、各キューブ C_i の最大基本キューブ $C_{i,ma}$ を順序付けることで、各キューブ C_i を昇順(降順)にすることができる。

(3) 部分マップ キューブ C_i の最大基本キューブ $C_{i,ma}$ と等しいか、より小さい2進数の最小項よりなるマップ上の部分は C_i を含む特定部分となり、この部分を部分マップとよび、 $S(C_i)$ と表す。図3にキューブ $C_1=0100-0110$ の部分マップ $S(C_1)=1100-1110$ を示す(太線枠)。部分マップ $S(C_i)$ はキューブ C_i の最大基本キューブ $C_{i,ma}$ で定義されるので、 $C_{i,ma}$ を

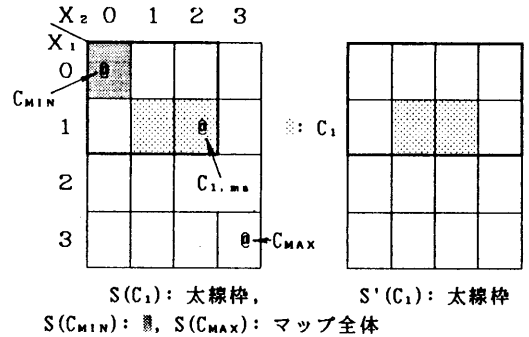


図3 部分マップの例 図4 修正部分マップ

順序付けることにより、キューブ C_i 同様に部分マップ $S(C_i)$ も昇順(降順)にすることができる。図3では $C_{1,ma}=C_{min}=1000-1000$ の部分マップ $S(C_{min})=1000-1000$ が最小で最小項1個だけを含み、 $C_{1,ma}=C_{max}=0001-0001$ の部分マップ $S(C_{max})=1111-1111$ が最大となり元のマップと等しくなる。

(4) キューブの拡大 あるキューブ C_i を部分マップ $S(C_i)$ の領域内で主項にまで大きくすることを拡大とよぶ。しかし、部分マップ $S(C_i)$ はキューブ C_i の最大基本キューブ $C_{i,ma}$ が $C_{j,ma} \leq C_{i,ma}$ なる基本キューブ $C_{j,ma}$ 領域のため、もし $C_{j,ma} > C_{i,ma}$ なるキューブ C_j が周辺に存在する場合、部分マップを修正しておく必要がある。図2のキューブ C_1 の場合、周辺に存在するキューブ C_2 は $C_{2,ma} < C_{1,ma}$ (θ 印)なので無視するが、キューブ C_3 は $C_{3,ma} > C_{1,ma}$ で変数 X_2 で隣接するキューブなので、キューブ C_1 の部分マップ $S(C_1)$ とキューブ C_3 の変数 X_2 での論理和をとり修正した部分マップ $S'(C_1)$ を使用する($S(C_1)=1100-1110 \Rightarrow S'(C_1)=1100-1111$) (図3 \Rightarrow 図4)。

キューブ C_i の部分マップ $S(C_i)$ (修正済み)の領域内で一変数 k ($1 \leq k \leq n$)ごとに、変数部 $k:(S(C_i)-C_i)$ をとった拡大可能を示すキューブ $CA_{i,k}$ (但し、変数部 $k:(S(C_i)-C_i) \neq \emptyset$)を求め、この拡大可能キューブ $CA_{i,k}$ 内に存在するキューブ集合 $F-C_i$ とDC(Don't Care)からトートロジー判定を行う(但し、 $F = \bigvee C_j$ ($j=1 \sim m$), $i \neq j$ とする)。トートロジー判定とは、拡大可能キューブ $CA_{i,k}$ 内に関数値0の最小項が存在する(AUTOLOGY=0)か否(AUTOLOGY=1)かを判定することである。又、キューブ $CA_{i,k}$ の変数部 k に1ビットが2個以上ある場合は、1ビット1個づつに分解して、それぞれトートロジー判定し、AUTOLOGY=1のビットだけの論理和で拡大

可能キューブ $CA_{1,k}$ の変数部 k を確定する。

次に、確認した拡大可能変数部を一変数ずつ順次拡大する。

[例題2.3] 図5のキューブ C_1 の拡大例の場合、 $C_1 = 0100-0110$ 、 $S(C_1) = 1100-1111$ なので、変数 X_1 での拡大可能キューブ $CA_{1,1} = 1000-0110$ 、変数 X_2 での拡大可能キューブ $CA_{1,2} = 0100-1001$ である。ここで、 $CA_{1,2}$ は変数 X_2 で1ビットが2個あるので分解して $CA_{1,2} = (0100-1000 \vee 0100-0001)$ として調べる。分解した一つのキューブ (0100-0001) の領域内にキューブ C_3 が存在するので $TAUTOLOGY = 1$ と判定される。つまりキューブ C_1 を部分マップ $S(C_1)$ 内で拡大すると変数 X_2 方向に1ビットだけ拡大でき、主項 $C_1 = 0100-0111$ となる。

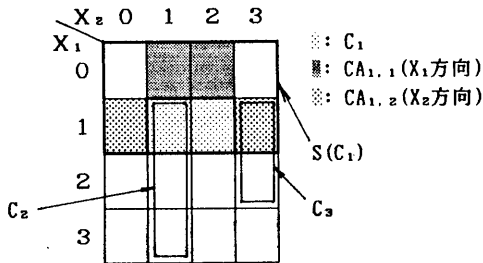


図5 拡大可能キューブ $CA_{1,k}$ ($k=1,2$)

3. 部分マップ法による簡単化

3.1 基本手続き

部分マップという限定領域を示す概念を使って与えられた関数を簡単化することを部分マップ法[2]とよぶことにすれば、部分マップ法で使用する基本手続きは主に次の2つである。

- (1) キューブの選出。
- (2) キューブの拡大。

(1) は与えられたキューブ集合 $F (= \vee C_i)$ の中からどのキューブ C_i を選ぶかという問題である。Besslich[4] は基本キューブ (最小項) に基づいた周辺との関係情報を求め、孤立している基本キューブを選ぶことで精度の良い解を得ている。本方法ではキューブ主体の演算で行なうが、同様な観点から、キューブ C_i と周辺キューブ C_j ($i \neq j$) とで隣接キューブや共通キューブを見つけたし、次の式(2)を満足するキューブ C_i を優先的に選ぶようにしている。

$$C_i \not\subseteq \{G_1 \cup \text{CONS}(C_i, G_2)\} \quad (2)$$

ここで G_1 はキューブ C_1 との共通キューブ集合、 G_2 はキューブ C_1 との隣接キューブ集合で、 $\text{CONS}(C_1, G_2)$ は C_1 と G_2 との式(3)によるコンセンサスである。又、 $\not\subseteq$ は $TAUTOLOGY = 0$ を意味する。キューブ C_1 とキューブ C_2 との $\text{CONS}(C_1, C_2)$ は、

$$C_1 = X_1^{A_1} \cdot X_2^{A_2} \dots X_n^{A_n}$$

$$C_2 = X_1^{B_1} \cdot X_2^{B_2} \dots X_n^{B_n}$$

$$\text{CONS}(C_1, C_2) = \bigcup_{i=1}^n X_1^{A_i \cap B_i} \cdot X_2^{A_2 \cap B_2} \dots$$

$$X_1^{A_i \cup B_i} \dots X_n^{A_n \cap B_n} \quad (3)$$

である。又、本方法では $\text{CONS}(C_1, C_2)$ で求めた一つのキューブ C_j が $C_1 \subseteq C_j$ の場合、 $C_1 \leftarrow C_j$ として、再度 C_1 と周辺とのキューブで関係を調べる。

[例題3.1] 図6のようなキューブ C_1 、 C_2 及び C_3 があるとして、まずキューブ C_1 を考える。 $G_1 = \{C_2\}$ 、 $G_2 = \{\phi\}$ 、 $C_1 \not\subseteq \{C_2\}$ となる。つまり周辺のキューブで被覆されない基本キューブ (*: 1000-0010) が存在する。同様に C_2 は $G_1 = \{C_1\}$ 、 $G_2 = \{C_3\}$ より $C_2 \subseteq \{C_1 \cup \text{CONS}(C_2, C_3)\}$ となる。 C_3 の場合は、 $G_1 = \{\phi\}$ 、 $G_2 = \{C_2\}$ より $C_3 \subseteq \{\text{CONS}(C_3, C_2)\}$ である。ここで $\text{CONS}(C_3, C_2)$ が C_3 を含む場合なので $C_3 (= 0010-0100) \leftarrow \text{CONS}(C_3, C_2) (= 0110-0100)$ として、再度周辺のキューブとの関係を調べ直す。すると $G_1 = \{C_2\}$ 、 $G_2 = \{C_1\}$ となり $C_3 \not\subseteq \{C_2 \cup \text{CONS}(C_3, C_1)\}$ となる。

式(2)は特異基本キューブ(図6の C_1 では*印の基本キューブ)の有無の判定ができるため必須主項の検出[1]に用いられている。その場合は各キューブが主項にまで拡大されていることが必要である。本方法では、拡大前のキューブ (内項) 段階で使用しているため厳密な意味で (擬似) 特異基本キューブを判定しているとはいえないが、拡大解釈して同様な意味で用いている。しかし (擬似) 特異基本キューブが見つからない場合は、最大基本キューブ $C_{1,m}$ で順序付けた最後の

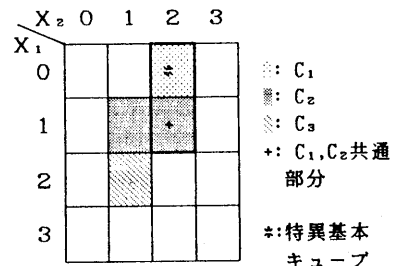


図6 キューブの選出例: C_1

キューブ C_k を選ぶようにしている。

一方、(2) キューブの拡大は2.2(4)のキューブの拡大を使うが、この時変数 k ($1 \leq k \leq n$) の拡大可能性を確認した情報を入れる一つのキューブ CK を用意し、2回目はこのキューブ CK の拡大可能 ($\neq \phi$) な変数だけを拡大する。ここで、どの変数から先に拡大するかという順序が問題となるが、本方法では拡大可能キューブ $CA_{1..k}$ 内のトートロジー判定で $F - C_k$ だけで $TAUTOLOGY = 1$ と判定した変数を DC も加えて $TAUTOLOGY = 1$ と判定した変数より先に拡大している。主項にまで拡大したキューブを解に入れ、 F にキューブがなくなるまで(1)、(2)の基本手続きを繰り返す。

次に、上記のような考えに基づいた簡単化アルゴリズムを記述する。

3.2 簡単化アルゴリズム

- 1° 解がキューブ集合 COV に求まるとする。
 $COV = \phi$ (空) としておく。関数 F をキューブ集合 $\{C_1, C_2, \dots, C_m\}$ で与える。
- 2° 併合後のキューブ数を q とする。
- 3° $NO = 0, i = 1$ 。
- 4° キューブ C_i とキューブ C_j ($i \neq j, j = 1 \sim q$) の関係が共通キューブなら G_1 へ、隣接キューブなら G_2 へ入れる。
- 5° $G_1 \cup G_2 = \phi$ なら、 $NO = i$ として 9° へ、それ以外は 6° へ。
- 6° $G_2 \neq \phi$ なら、 $CONS(C_1, G_2)$ を求める。但し、コンセンサスキューブ C_j が $C_1 \subseteq C_j$ の場合、 $C_1 \leftarrow C_j$ として 4° へ戻る。
- 7° $C_1 \not\subseteq \{G_1 \cup CONS(C_1, G_2)\}$ なら、 $NO = i$ として 8° へ。それ以外は $i = i + 1$ として 4° へ戻る。但し、 $i > q$ なら 8° へ。
- 8° 各キューブ C_i を最大基本キューブ $C_{1..m}$ により順序付けし、昇順 (降順) にする。もし、 $NO = 0$ の場合は、最後 (最初) のキューブを選び $NO = q$ (1) とする。
- 9° キューブ C_{No} の部分マップ $S(C_{No})$ を生成する。ここでキューブ C_{No} の最大基本キューブ $C_{No..m}$ と比較して $C_{1..m} > C_{No..m}$ なる DC も含めた周辺キューブ C_j が存在し且つ共通キューブなら $S(C_{No}) \cup C_j$ とし、又、変数 k での隣接キューブなら、変数 $k : (S(C_{No}) \cup C_j)$ とした修正部分マップ $S'(C_{No})$ を求める。
- 10° キューブ $CK = \phi$ とおく。
 キューブ C_{No} と部分マップ $S(C_{No})$ より、拡大可能キューブ $CA_{No..k}$ ($1 \leq k \leq n$) を求め、確認した変数情報をキューブ CK に入れる。

$CK \neq \phi$ の変数部を F だけのキューブで拡大できる変数部が先になるよう順序付ける。

- 11° 順序付けの順に従って、一変数ずつキューブ C_{No} を拡大する。

$$COV = COV \vee C_{No}, DC = DC \vee C_{No}, F \textcircled{\#} C_{No} \Rightarrow F \text{ とする。}$$

ここで $\textcircled{\#}$ は Disjoint sharp 演算 [7] である。

- 12° 11° で $F = \phi$ なら 13° へ。それ以外は F の項数を改めて q として 3° へ戻る。
- 13° COV 中の冗長キューブを除去する。
- 14° COV と与えた F で検証する。

[例題3.2] 表1に示す6個のキューブで与えられる二値3変数1出力の対称関数 (SYM3) に上記 3.2 簡単化アルゴリズムを適用してみる。

$$COV = DC = \phi, \text{ 関数 } F$$

としてキューブ C_i ($i = 1 \sim$

6) を与える。併合すると4個になる (図7-(1))。但し、マップ上の数字はキューブの番号 i ($i = 1 \sim 4$) を表す。
 $NO = 0, i = 1$ として、先ずキューブ C_1 と他のキューブ C_2, C_3 及び C_4 と調べる (4°)。

$$G_1 = \phi, G_2 = \{C_2, C_3\}.$$

$CONS(C_1, C_2)$ で生成した

$$C_7 = 10-11-01 \text{ は } C_1 = 10-$$

10-01 を含むので $C_1 = 10-11-01$ ($\leftarrow C_7 : 2$ 重線枠) として、もう一度関係をみる (6°)。今度は $G_1 = \{C_2\}$ 、 $G_2 = \{C_3\}$ となり、7° の関係を調べるが $TAUTOLOGY = 1$ となる。 i を1増やし $i = 2$ として、同様にキューブ C_2 とキューブ C_1, C_3 及び C_4 と調べるが $TAUTOLOGY = 1$ である。 $i = 3, i = 4$ の場合も $TAUTOLOGY = 1$ である。但し、キューブ C_4 の場合はキューブ C_1 の時同様にコンセンサスでキューブが修正される (図7-(2))。一通りキューブ C_i を調べたが $TAUTOLOGY = 0$ のキューブがない ($NO = 0$) ので、各キューブの最大基本キューブ $C_{1..m}$ で昇順 (図7-(3)) とし最後 ($q = 4$) のキューブ C_4 を拡大するキューブとして選ぶ (8°)。この場合 $C_{1..m}$ 以上の最大基本キューブは周辺にないので部分マップを修正する必要がない (図7-(4))。キューブ $C_4 = 10-01-11$ 、部分マップ $S(C_4) = 10-11-11$ なので変数 X_2 方向だけの拡大可能キューブ $CA_{4..2}$ が考えられ、拡大を試みてもキューブ C_4 のままである。 $COV = COV \vee C_4, DC = DC \vee C_4, F \textcircled{\#} C_4 \Rightarrow F$ とする (11°)。 $\textcircled{\#}$ によりキューブ C_4 と重複しているキューブ

表1 SYM3

	X1	X2	X3
C1	01	10	10
C2	10	01	10
C3	01	01	10
C4	10	10	01
C5	01	10	01
C6	10	01	01

C_1 と C_3 が縮小される(図7-(5))。

残りのキューブ3個で上記手順を繰り返す。先ず、キューブ C_1 が $CONS(C_1, C_2)$ で生成したキューブで修正される($C_1 \leftarrow 01-11-10$)。再度キューブ C_1 とキューブ C_2, C_3 で関係を調べると $G_1 = \{C_2\}$ 、 $G_2 = \{\phi\}$ より $C_1 \# \{C_2\}$ となるので $NO = 1$ として拡大するため選出する(図7-(6))。キューブ $C_1 = 01-11-10$ の拡大では共通キューブ C_3 が存在するため部分マップ $S(C_1)$ が修正され $S'(C_1) = 11-11-11$ (修正前: $11-11-10$)となる(図7-(7))。キューブ C_2 と C_3 の位置が入れ替わっているのはキューブを昇順にしているためである。どの変数方向にも拡大できず元のままのキューブ C_1 が COV に入る(図7-(8))。

3回目の繰り返しでキューブ C_1 がコンセンサスで修正された後(図7-(9))、周辺に F のキューブが存在しない(6°)のでただちに拡大するため 9° へいく。キューブ $C_1 = 11-10-01$ を部分マップ $S(C_1) = 11-10-11$ 内で拡大する(図7-(10))。拡大項を COV に入れると $F = \phi$ なので終了する。 COV に求めた3個のキューブで $SYM3$ は次のように表現できる。

$$SYM3 = 10-01-11 \vee 01-11-10 \vee 11-10-01$$

4. 計算結果

3. で提案した簡単化アルゴリズムとMINI-II [8]をIBM 3081-KX4(富山大学情報処理センター)上にFORTRANでプログラム化し、計算比較を行なった。

表2は、簡単化プログラム評価用のベンチマーク算術回路[1]の中から選んだ5種類の関数で比較した結果である。ほぼ同程度の項数が得られるが、必須主項が少ない関数—例えば、MLP4は12個の必須主項を持つ—で変数が大きくなると本方法はMINI-IIより極

表2 算術関数による比較

入力データ名	MINI-II		本方法	
	項数	時間	項数	時間
ADR3	31	1.1	32	1.6
MLP3	34	1.1	32	1.1
MLP4	126	41.3	145	83.4
NRM4	121	15.2	121	25.1
ROT8	58	3.5	58	8.9
WGT6	63	1.5	63	0.5
WGT8	255	38.5	255	18.7

時間(CPU TIME):秒

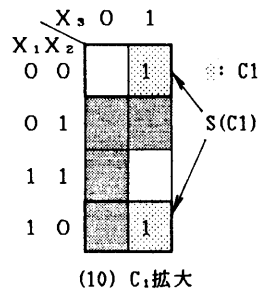
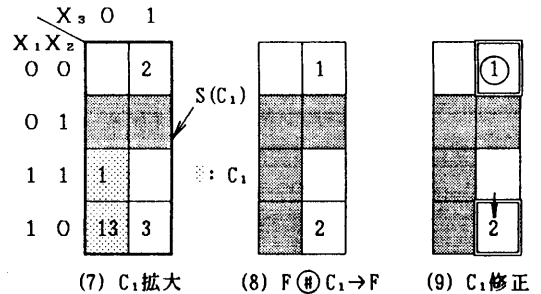
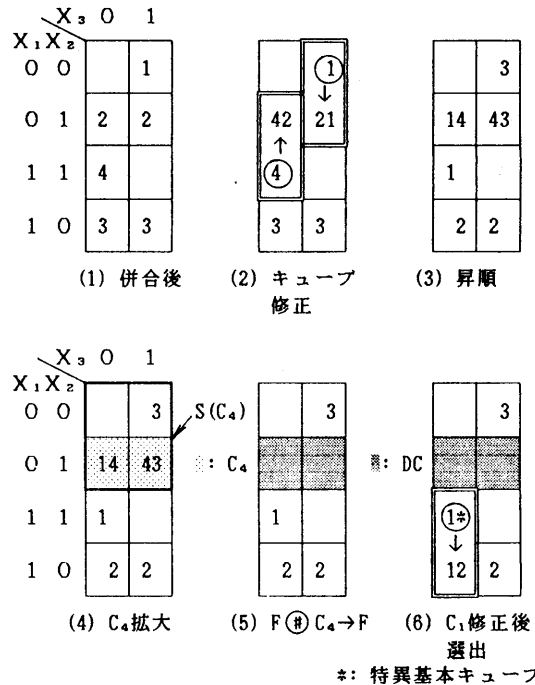


図7 SYM3の簡単化手順
(数字:キューブ C_i の i を表す)

端に精度が悪く且つ時間もかかる。逆に、必須主項が多い関数—例えばWG T 8 は129個の必須主項を持つ—は同じ項数の解が2~3倍高速に求まる。この理由は、本方法では(擬似)特異基本キューブが得られない場合、単に最大基本キューブで昇順に順序付けた最後のキューブを選んでいることが原因と考えられる。

表3は乱数により生成したランダム関数でn変数四値入力二値1出力関数のnを変化させた場合である。但し、各変数nはいずれも部分濃度 d_1, d_2, d_3, d_4 を0.8, 0.1, 0.1, 0.0と指定したキューブ数70個の関数を生成し、いずれの場合とも解の項数が70個得られるので計算時間だけを示してある。ここで部分濃度というのは、四値の場合、各変数部で4ビット中1のビットが1個現れる、2個現れる、3個現れる、4個現れるという確率が各々 d_1, d_2, d_3, d_4 であるキューブを生成することである。例えば、 d_1, d_2, d_3, d_4 を1, 0, 0, 0と指定してやれば最小項で関数を生成し、0, 0, 0, 1とすれば UNIVERSAL CUBEが生成され、部分濃度を変えて任意のキューブが生成できる。変数が大きくなってくると最小項では無理なので部分濃度を調整してやるが必要となる。ちなみに $n=50$, 70個のキューブを単純に最小項に分解して総和を求めてみると8317920個にもなる。このような部分濃度では、変数が大きくなるにつれ両方法の時間差がひらいてきて本方法が非常に高速であることがわかる。これは、変数が大きくなってくると MINI-IIの否定の項数(表3の()で示した数値)が増大してくることが一因と考えられる。そして我々の作成MINI-IIでは、否定の項数>5000となる部分

濃度、キューブ数の関数は実行不能となる(実行領域8MB)。一方、本方法において、マップの全最小項のうち関数値1をとる最小項が多くなる(いわゆる真理値表濃度が高い場合に相当する)ような部分濃度、キューブ数を指定してやると、MINI-IIと比較できる小さな変数でも逆に精度が悪く、時間もかかる。例えば、 $n=7$ で d_1, d_2, d_3, d_4 を0.1, 0.3, 0.4, 0.2とし、100個のキューブを生成して簡単化するとMINI-IIでは項数31の解が21.7秒で求まるのに対して、本方法では項数34の解が求まるのに61.2秒もかかる。

5. おわりに

本稿で多値入力二値出力関数の一簡単化アルゴリズムを提案した。本方法は、真理値表濃度(我々は部分濃度で指定している)が低い関数は非常に高速に簡単化できる。反面、真理値表濃度の高い関数は苦手として変数が大きくなると非実用的となる。それ故、真理値表濃度の高い関数は否定を求めてから、その否定に本方法を適用するという考えられるが、周辺キューブとの関係情報をうまく利用したキューブ選出法及び拡大の変数順の決定等見直すべき課題も幾つかある。

参考文献

- [1] 笹尾勤: "PLAの作り方・使い方"、日刊工業新聞社(1986)。
- [2] 宮腰、松田: "2値論理関数簡単化の一手法—部分マップ法について—"、電子情報通信学会論文誌、Vol. J71-D, No. 11, pp. 2259-2265, 1988。
- [3] P. Tirumalai and J. T. Butler: "Analysis of minimization algorithms for multiple-valued programmable logic arrays", 18th ISMVL, pp. 226-236 (1988)。
- [4] P. W. Besslich: "Heuristic minimization of MVL functions: a direct cover approach", IEEE Trans. on Comput., pp. 134-144 (Feb. 1986)。
- [5] G. W. Dueck and D. M. Miller: "A direct cover MVL minimization using the truncated sum", Proc. of 17th ISMVL, pp. 221-226 (1987)。
- [6] G. W. Dueck and D. M. Miller: "Directed search minimization of multiple-valued functions", Proc. of 18th ISMVL, pp. 218-225 (1988)。
- [7] S. J. Hong, R. G. Cain, and D. L. Ostapko: "MINI: A heuristic approach for logic minimization", IBM J. Res. & Dev., 18, 5, pp. 443-458 (Sept. 1974)。

表3 ランダム関数による比較例(四値)

変数の数 n	MINI-II 時間	本方法 時間
10	19.2 (680)	0.9
20	194.7 (2033)	1.7
30	501.2 (2447)	2.5
40	1324.0 (3773)	3.4
50	2308.6 (4275)	4.3

$(d_1, d_2, d_3, d_4) = (0.8, 0.1, 0.1, 0.0)$,
 $F = \bigvee C_i (i=1 \sim 70) \Rightarrow 70$ 個の解
 (): 否定の項数(但し、拡大前), 時間: 秒

- [8] T.Sasao:"Input variable assignment and output phase optimization of PLA's",IEEE Trans. Comput.,Vol.C-33,No.10,pp.879-894(Oct.1984).
- [9] R.L.Rudell and A.L.M.Sangiovanni-Vincentelli : "Espresso-MV:Algorithms for multiple-valued logic minimization",CICC-85,pp.230-234(May. 1985).
- [10] T.Sasao:"Tautology checking algorithms for multiple-valued input binary functions and their application",Proc. of 14th ISMVL,pp. 242-250(May.1984).