論理式簡単化アルゴリズム MINI-LN の二値入力変数での計算結果と手数について

宮腰　隆，松田　秀雄，畠山　豊正，
中嶋　芳雄

1．はじめに

近年ディジタルシステムの大規模化に伴い論理回路の自動設計が重要性を増している。VLSI 内で多用される PLA（Programmable Logic Array）は論理構造が AND-OR 2 段という単純な構造をしているため，設計自動化が最も進んでいる。PLA における積項線数を減らしたいとか，接続を減らしたいという設計問題は論理式簡単化問題に帰着される。

これまでにいくつも発表されている論理式簡単化アルゴリズムの中でも代表的な方法が MINI[1] であり，ヒューリスティックな手続きで繰り返し解を改善し，良い近似解が得られる。しかし，任意の項の拡大に（与えられた関数 F の）否定 $\bar{F}$ を用いるため，$\bar{F}$ を求める必要がある。このとき関数によっては，$\bar{F}$ の生成項が多くなり大きな記憶容量が必要となる場合がある。そこで我々は $\bar{F}$ を求めず，任意の項の拡大にさいしては，その項が最大限拡大可能な領域を周辺の F 項や D（don't care）項より見当つけ，この限定領域内の否定（これを局所否定，Local negation と呼ぶ）だけを生成し，項の拡大に用いることで記憶容量の軽減を図った MINI-LN 法[2] を考案した。MINI-LN では，項の拡大ごとに局所否定を求める必要も生ずるが，真理値表濃度の小さい関数では，項の拡大時に操作対象の否定項が少なくなり，MINI よりも速く解が得られる。これら四値入力二値出力関数の場合の詳細なデータは既に発表している。

本論文では，応用上より重要とみられる二値入力二値出力関数に適用した場合の解の項数とか計算時間といった計算結果を示す。併せて MINI-LN の手数が，いま入力変数の数 n の併合後の項数を q とすれば，最小の手数は q^2，最大の手数は $q^3/2$ となることを理論的に導く。その妥当性は計算結果を使って確かめている。手数の計算は，手法の計算時間をあらかじめ見当つけて，その関数の計算が実行可能か否かを判定する上で重要で，MINI などではなされていない。

2．アルゴリズム

MINI および MINI-LN については文献 1)，2) にそれぞれ詳述してあるので，ここではその概略のみ記す。特に用語の定義，記号については，文献 2) のものをそのまま用いている。

2.1　MINI の方法

関数 F（$=\vee c_j$，$j=1$，2，…，q）を与える。

1）F の項を（$x \vee \bar{x}$）$=1$ の関係式を使って併合する。併合後の項数を q_1 とする。

2）F の否定 $\bar{F}$ を求める。

3）F を用いて，$\bar{F}$ の各項を拡大する。

4）$\bar{F}$ を用いて，F の各項を拡大する。拡大後の項数を q_2 とする。

5）$q_2 < q_1$ なら $q_1 = q_2$ として次の6）へ行く。それ以外は8）へ行く。

6）Fの各項を縮小する。いま１つの項をCとし，C以外の項をGとし，$F = C \vee G$ で $C \leqq G$ ならば，Cは除去できる。

7）Fの各項を整形する。
手順4）へ戻る。

8）得られた解がはじめに与えた関数と一致するか検証する。

2.2　MINI-LN の方法

ここでは，上記 MINI の手順と対比させて，相違点を中心に記述する。F_C には解が，F_E には（擬似）必須主項として検出した項がそれぞれ求まるものとし，Dを don't care とすれば，はじめは $F_C = F_E = D = \phi$（空集合）とする。また，拡大した主項が（擬似）必須主項かどうかを判定する変数 EPI＝1 としておく。

関数F（$= \vee C_j$，$j = 1$，２，…，q）を与える。

1）Fの項を併合する。
MINI の方法の 2)，3) は実行しない。

＃：Fの項数を q_1 とした後，Fの各項 C_j，（$j = 1$，２，…，q_1）について，次の４a)，４b）を行う。但し，EPI＝０のときは４b）をとばす。

４a）ひとつの項 C_j を主項にまで拡大する。

４b）主項 C_j が（擬似）必須主項かどうかを判定する。もし，（擬似）必須主項なら $F_E = F_E \vee C_j$ とする。

Fの項をひととおり拡大した後，

４c）$F_E \neq \phi$ なら $F = F - F_E$，$F_C = F_C \vee F_E$，$D = D \vee F_E$ とする。改めて $F_E = \phi$ として 4d）へ行く。それ以外は EPI＝０として 4d）へ行く。

４d）$F \neq \phi$ ならFの項数を q_2 として5）へ行く。それ以外は8）へ行く。

5）$q_2 < q_1$ なら $q_1 = q_2$ として6）へ行く。それ以外は8）へ行く。

6）Fの各項を縮小する。

7）Fの各項を整形する。
4a）の前の＃：へ戻る。

8）得られた解を検証する。

MINI-LN では，MINI のように否定$\overline{F}$を求めないので，拡大したい項 C_j の拡大可能領域（CM_j）を周辺のFやDの項より見当つける。次に CM_j という限定した領域内の（局所）否定 $\overline{F_L}$ を求め，項 C_j の拡大に $\overline{F_L}$ だけを使用する。また，上記の手順4b）において，１回目に検出されるのが必須主項であり，繰り返し２回目以降に検出されるのが擬似必須主項である。

3．手数の評価

3.1　MINI-LN 法の手数について

MINI-LN 法の正確な手数の評価は困難であるが，実験結果を説明づけるだけの大まかな評価は可能である。そこで，いくつかの仮定をおく。

一般に，p 値入力 n 変数関数では，一つの項 C_j は p ビットの part（変数部）が n 個あるので，＃演算のような演算[1] を行うと，pn に比例した手数が要る。いま表１のように $n \leqq 12$（二値）程度だと pn をほぼ一定とみて，手数の比例定数の部分に含めてしまってよいと考えられる。

そこで基本的に，一つの項 C_j が他の q 個の項と＃演算なり，共通部分があるかないかを調べたり，

隣接しているかどうかを調べたりするときの手数をそれらの演算の種類を問わず q とする。つまり，項と項の間で一つの操作が行われたら，手数1とみなす。

こういう仮定で MINI-LN 法の各部の手数をみてみると，次のようになる。但し，主要でないものは省略する。

（項の拡大）　ここでの手数は，（いま拡大する項 C_j の拡大可能領域）CM_j を求める手数で決まる。CM_j を決めるため，一つの項 C_j が $F \vee D$ の項と手数1の操作を行うので $q_F + q_D = q$，但し，q_F ははじめFに与えた項数，これを q とする。q_D はDの項数で，はじめは0である。C_j が q_F 個あるので，CM_j を求める手数は $q_F q$ に比例する。

（(擬似)必須主項の検出）　判定のために $C_j \trianglelefteq \{H_1 \vee D_1 \vee H_2\}$[2] を使うが，$\{H_1 \vee D_1 \vee H_2\}$ を作るために，一つの項 C_j が $F \vee D$ のすべての項と手数1の操作を行っている。従って，主項1個につき $q_F + q_D = q$，Fのすべての項を主項にまで拡大して，これを調べるので，ここでの手数は $q_F q$ に比例する。

（縮小）　項 C_j が $C_j \# ((F - C_j) \vee D)$ を求めるために，$F \vee D$ のすべての項と手数1の操作を行うので $q_F + q_D$，C_j はFのすべての項につき行うので，ここでの手数は $q_F q$ に比例する。

（整形）　F内の二項 C_1 と C_2 が変形できるかどうかをすべての組み合わせでみるので，$q_F q_F$ の手数がかかる。$q_F q_F \leqq q_F q$ なので，結局すべての手順で $q_F q$ の手数がかかるといえる。

以上の各部の手数は，図1のようになる。

つまり拡大から整形までひととおり手順が行われると，Fの項数が q_F なら $q_F q$ に比例した手数がかかるということになる。ある場合とある場合の手数の比率を求めるときは，比例定数は消えるので，依然として拡大から整形までの手数を $q_F q$ としよう。

さて，ここでq 個の項数をもつ関数Fが与えられたとする。すなわち，はじめ $q_F = q$，$q_D = 0$，$q_F + q_D = q$ である。そして拡大から整形までの手順ループを次のようなプロセスをへて，項数 αq，$\alpha \leqq 1$ の解が得られたとする。但し，ここで項が減るのは必須主項あるいは擬似必須主項が求まって減るのか，あるいは，縮小のところで，他の項に含まれて減るのかは問わないことにする。

a）Fの項が拡大→整形の手順を一回通るごとに1個ずつ項が減っていき，$(1-\alpha)q$ 個減って，αq 個の解が求まる場合。

このプロセスでは，図1を一回まわるごとに q_F の数は q，$q-1$，

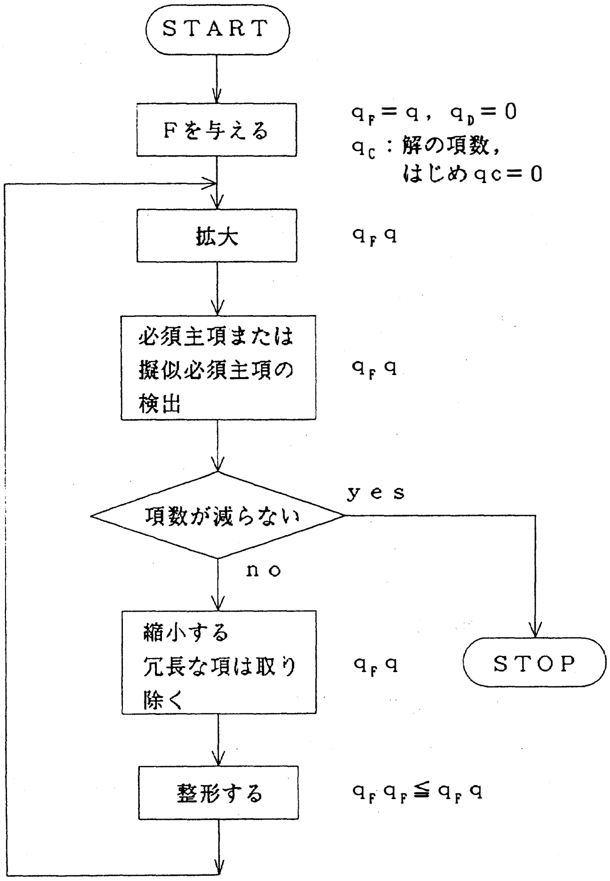

図1　MINI-LN 法の各部の手数

$q-2$，…と減っていき，$(1-\alpha)q$ 個まで減るので，手数の総数は
$\{q+(q-1)+(q-2)+\cdots+(1-\alpha)q\}q$ である。
｛　｝中は $\dfrac{q+(1-\alpha)q}{2}\cdot(\alpha q+1)$ となるので，

$$\text{手数の総数}=\frac{(2-\alpha)\alpha q}{2}\cdot q^2, \qquad ①$$

となる。但し，$q\gg 1$ とする。
これは真理値表濃度が大きい場合に起こる。

b）F の項が拡大→整形の手順を一回通るごとに r 個ずつ項が減る場合は，a）同様に考えると，手数の総数は
$\{q+(q-r)+(q-2r)+\cdots+(1-\alpha)q\}q$ である。
｛　｝の中は $(\alpha/r)q+1$ の項からなり，
｛　｝中 $=\dfrac{q+(1-\alpha)q}{2}\times\left(\dfrac{\alpha}{r}q+1\right)$ であり，

$$\text{手数の総数は}=\left\{\frac{q+(1-\alpha)q}{2}\times\left(\frac{\alpha q}{r}+1\right)\right\}q, \qquad ②\cdot 1$$

$$\fallingdotseq\frac{(2-\alpha)\alpha q}{2r}\cdot q^2, \qquad (\text{但し，}q\gg 1) \qquad ②\cdot 2$$

となる。特に，一回の拡大→整形で解が求まってしまう場合，すなわち，F の項が拡大の結果，全部，必須主項である場合がこれにあたるが，このとき $r=q$，$\alpha=1$ なので②・1より

$$F\text{の項が全部必須主項のときの手数}=q^2, \qquad ③$$

となる。これは真理値表濃度が小さい場合に起こる。

さて，ここで関数を F に与えてから，解が求まるまでは，いろんなプロセスがあろうが，a)，b）のいずれかの場合に近似的にあてはめてしまうことができると思われる。すると，手数が最もかかるのはa）の場合で，手数が最も少ないのはb）式③のときである。とりわけ，a）の場合で $\alpha=1$ なら，その手数 $=q^3/2$ で最大となる。これははじめに q 個の項数の関数を F に与えると，拡大→整形のループをまわるごとに F の項が1つずつ減り，その分 D と F_C（解が求まる集合）にたくわえられていって F が ϕ（空集合）になり，F_C に q 個の項数の解が求まったときである。

すなわち，MINI-LN 法の手数は最小で q^2，最大で $q^3/2$ であり，その他の解が求まるプロセスの手数はこの間にあるといえる。但し，これは手数の比率のことで，ある場合の手数とある場合の手数を比較すると，その比率がこの間に入るということである。

4．計 算 結 果

4.1　二値入力変数での比較

表1は，入力変数の数 $n=8\sim12$ について，二値ランダム関数を真理値表濃度 $d=0.2$，0.5，0.8 で各5個ずつ発生し，その平均値で表したMINI，MINI-II，MINI-LN およびA5の解の項数と計算時間（秒）である。従って，平均値の項は計15個の関数の総平均値を表す。但し，$n=12$ だけは各 d で2個ずつ発生した関数の平均値である。MINI-II[3] は，MINI に必須主項の検出を追加する等改良した方法であり，比較のために併せて実行した。A5[4] は，関数の否定 $\overline{F}$ を使わない方法なので参考データとして付記した。

解の項数は，$n=8\sim12$における各平均値でみて MINI-LN が一番少ない。これは MINI-LN では必須主項ならびに擬似必須主項を確実に検出して解に採用しているためと考えられる。また，MINI-II でも，はじめに必須主項の検出を行うため特異最小項が多い $d=0.2$ で MINI より項数の少ない良い解が得られている。

表1　ランダム関数による比較結果（二値）

入力変数の数 n	真理値表濃度 d	発生最小項数	併合後の項数	MINI 解の項数	MINI TIME（秒）	MINI-II 解の項数	MINI-II TIME（秒）	MINI-LN 解の項数	MINI-LN TIME（秒）	A5 解の項数	A5 TIME（秒）
8	0.2	52.2	35.4	32.4	0.6	32.4	0.6	32.4	0.2	32.4	1.1
	0.5	135.6	60.8	46.8	1.6	47.4	1.7	47.4	1.2	48.6	7.8
	0.8	207.6	61.4	38.2	1.2	38.2	1.5	37.2	2.2	41.2	16.1
	平均値	131.8	52.5	39.1	1.1	39.3	1.3	39.0	1.2	40.7	8.4
9	0.2	107.4	69.8	63.0	3.0	62.8	3.0	62.8	0.7	63.0	6.6
	0.5	256.2	121.2	90.2	8.1	89.2	8.7	90.2	4.3	94.6	41.9
	0.8	409.4	118.8	72.8	5.8	73.2	7.2	72.2	8.5	77.0	83.9
	平均値	257.7	103.3	75.3	5.6	75.1	6.3	75.1	4.5	78.2	44.1
10	0.2	212.8	134.6	122.4	13.7	121.6	13.2	121.4	2.9	122.0	38.6
	0.5	506.8	233.4	169.0	34.9	168.0	37.3	165.6	19.8	178.4	235.2
	0.8	818.0	238.2	129.0	27.7	132.0	31.9	128.2	31.7	148.6	465.5
	平均値	512.5	202.1	140.1	25.4	140.5	27.5	138.4	18.2	149.7	246.4
11	0.2	416.8	257.0	230.0	65.8	229.6	57.6	229.2	10.6	231.2	216.3
	0.5	1011.6	443.0	315.8	154.2	314.0	192.1	312.0	81.8	338.6	1495.5
	0.8	1644.8	470.8	238.8	148.4	238.8	175.0	235.0	125.9	273.0	2980.4
	平均値	1024.4	390.3	261.5	122.8	260.8	141.6	258.7	72.8	280.9	1564.1
12	0.2	832.0	504.5	447.0	324.8	446.5	273.0	446.5	46.3	453.0	1395.5
	0.5	2084.0	904.0	609.0	933.2	612.0	1096.9	607.5	435.0	667.5	10999.1
	0.8	3268.0	906.0	450.5	875.7	462.5	806.4	446.0	547.0	473.5	22991.8
	平均値	2061.3	771.5	502.2	711.2	507.0	725.4	500.0	342.8	531.3	11795.5

n = 8 ～11は d =0.2, 0.5, 0.8で各５個ずつ発生した関数の平均値，n =12は２個ずつの関数の平均値．
TIME：計算時間．

表2　算術関数による比較結果（二値多出力）

関数名	入力変数の数	出力数	発生項数	MINI 解の項数	MINI TIME（秒）	MINI-II 解の項数	MINI-II TIME（秒）	MINI-LN 解の項数	MINI-LN TIME（秒）
MLP4	8	8	225	129	18.2	130 (12)	22.0	129 (12)	13.5
ADR4	8	5	255	75	6.8	75 (35)	8.9	75 (35)	5.8
NRM4	8	5	255	124	8.8	121 (23)	13.3	120 (23)	10.6
SQR6	6	12	63	50	1.3	50 (3)	2.3	50 (3)	1.9
ROT8	8	5	255	57	2.1	57 (9)	3.3	57 (9)	2.3
WGT8	8	4	255	255	30.0	255 (129)	35.8	255 (129)	20.1
INC8	8	9	256	37	4.0	37 (10)	6.1	37 (10)	6.6
SYM9	9	1	420	86	8.4	88 (0)	10.5	87 (0)	7.2

（　）は必須主項の数．TIMEは計算時間．

計算時間では，$d \leqq 0.5$ で MINI-LN が MINI や MINI-II よりも高速である。真理値表濃度が小さいところでは変数が四値であれ，二値であれ，（拡大する項の）最大拡大可能領域（CM_{max}）がマップの高々半分以下の大きさであり，局所否定の最大生成項数（$\overline{F_{Lmax}}$）も少なくなり，拡大に要する操作回数が，関数の否定$\bar{F}$を操作対象とする MINI や MINI-II に比べて減少するからである。（四値の場合，CM_{max} と $\overline{F_{Lmax}}$ の値は，$d=0.65$でそれぞれマップの0.48，$\bar{F}$ の0.55である[2)]。）

表２は，アルゴリズムの性能評価によく使われる算術関数[5)]の中からいくつかを選んで実行してみた結果である。SYM 9 以外は多出力関数であるが，入力変数の数 n が６～９とそれ程大きくないので，三方法にほとんど差異はみられない。

いずれの方法も FORTRAN でプログラム化し，富山大学情報処理センターの IBM 3081-KX 4（実行領域 8 MB）で実行した。

4.2　二値入力変数での手数

二値入力変数の場合，変数の数 n が変化すると，関数の項の数ははじめに与えた真理値１の最小項の項数でみても，併合後の項数でみても約２倍ずつで増えている。これは２値入力 n 変数のマップのセルが 2^n 個あることから，当然予想されることである。このことから，n 変数の項の数 q は，$n-1$ 変数の項の数 q_0 の２倍で $q=2q_0$ と表せる。従って，MINI-LN 法の手数は１変数増えると，最小 $2^2=4$ 倍から最大 $2^3/2=4$ 倍の間で，すなわち４倍で増加するといえる。

表１において，MINI-LN 法では $n=8$ から $n=12$の間で１変数増えるごとに，手数（計算時間）は平均値でみて約４倍で増えており，上記の理論とよく合致していることがわかる。また，項数が減るのは，必須主項または擬似必須主項で減るのか，冗長な項があって減るのかを問わないとしているため，上記の MINI-LN の手数の解析は，MINI では項の拡大→整形だけで冗長な項を減らして解を求めているとか，MINI-II はこれに必須主項を求める手順が加わる等といった各方法に違いがあろうとも，いずれの方法にも適用可能だと思われる。実際，MINI，MINI-II についても n 変数と $n-1$ 変数との計算時間（手数）の比率は MINI-LN の傾向とほぼ同じである。

5．ま　と　め

本論文では，MINI-LN と基本的考え方が類似している MINI と MINI-II を二値入力二値出力関数に適用した計算結果を示した。

MINI-LN 法の特徴は，他の二方法と比較して次のとおりである。

(1)　真理値表濃度 $d \leqq 0.5$ の関数（入力変数の数 $n=8$ ～12）では，解の項数および計算時間ともに優れている。

(2)　局所否定の使用により，MINI，MINI-II では否定項数が多すぎて記憶容量の制限上解が求まらない場合でも，求めることができる（四値で確認済み[2)]）。

更に，MINI-LN 法の手数[6)]が，本論文のように二値入力変数の場合，１変数増加するごとに４倍ずつ増加することを理論的に導き，ランダム関数による計算結果とよく合致することを示した。また，この性質は，他の MINI や MINI-II にもあてはまることを示した。

参 考 文 献

1）Hong, S.J., Cain, R.G. and Ostapko, D.L.: MINI: A heuristic approach for logic minimization, *IBM J. Res. Dev.*, **Vol. 18**, No. 5, pp. 443-458(1974).

2）宮腰　隆，松田秀雄，帛山豊正：論理式簡単化の一手法：MINI-LN，情報処理学会論文誌，**Vol. 34**, No. 12, pp. 2624-2635(1993).

3) Sasao, T.: Input variable assignment and output phase optimization of PLA's, *IEEE Trans. on Comput.*, **Vol. C-33,** No. 10, pp. 879-894(1984).
4) 石川啓二，笹尾　勤，寺田浩詔：論理式簡単化アルゴリズム：A 5，電子通信学会論文誌，**Vol. J66-D,** No. 1, pp. 41-48(1983).
5) 笹尾　勤：PLA の作り方・使い方，日刊工業新聞社(1986).
6) 宮腰　隆，松田秀雄，畠山豊正，中嶋芳雄：論理式簡単化アルゴリズム MINI，MINI-LN の計算時間量について，平成 5 年度電気関係学会北陸支部連合大会講演論文集 **E-39** (1993).

平成 5 年度電気関係学会北陸支部連合大会で一部発表（9 月)。

Computing Results and Complexity of Basic Operations of The Logic Minimization MINI-LN Algorithm for Boolean Functions

Takashi Miyagoshi, Hideo Matsuda, Toyomasa Hatakeyama
and Yoshio Nakajima

A heuristic logic minimization technique called MINI-LN, which we recently reported, and other two heuristic methods, MINI and MINI-II, are applied to randomly generated functions with two-valued input variable $n=8\sim12$. Consequently, our method produces fewer number of terms in the final expressions and compute faster than the other two methods on the average of computing results at the functions with truth table density $d\leqq0.5$.

Furthermore, we theoretically derive that the complexity of basic operations performed in MINI-LN increase four times for every one variable rise, and confirm the suitableness with experimental results.

〔英文和訳〕

論理式簡単化アルゴリズムMINI-LNの二値入力変数での計算結果と手数について

宮腰　　隆，松田　秀雄，畠山　豊正，
中嶋　芳雄

我々が先に提案した論理式簡単化の一手法MINI-LNと他の二方法MINI，MINI-IIを二値入力変数$n=8\sim12$のランダム関数に適用し比較評価している。その結果，MINI-LN法は真理値表濃度$d\leqq0.5$の関数の実行結果の平均で，他の二方法より最終的に得られた論理式の項数が少なく，計算時間が速い。

また，MINI-LN法の手数が二値入力変数の場合，1変数増えるごとに4倍ずつ増加することを理論的に導き，その妥当性を計算結果により確認している。