

多分枝展開法を用いた主項生成について

5B-6

伊藤 貴* 塩原 悟 凌 暁萍 後藤 公雄
神奈川工科大学情報工学科

1. 緒言 主項生成([1],[2])において, 多分枝展開法を開発した. この手法では, 展開により関数を分割して隣接性を調べる場合, 1個のみの変数ではなく, 複数個の変数を同時に用いるようにする. この手法のアルゴリズムをC言語で実行したので報告する.

2. アルゴリズム 本手法のアルゴリズムを述べる.
[ステップ1] 与えられた論理関数 f を最小項の番号(10進数)の集合で表す. この番号を昇順にソートしたものが集合 S_j^i (または S_j^i) である.

[ステップ2] 分割ビット長 n_b (複数個の変数を用いて展開を行う. 今後この変数の個数をビット長 n_b と呼ぶ) を集合 S_j^i (または S_j^i) に対して決定する.

[ステップ3] MSBを含む連続した上位 n_b ビットが, 抜き取られる. このとき2進 n_b の組合せ(今後これを分割ビットと呼ぶ)は, 集合 S_j^i (または S_j^i) の最小項番号を 2^{n_b} 個の異なるグループ G_j ($j = 0, 1, 2, \dots, 2^{n_b} - 1$) に分割する. これらの各グループは分割集合 T^i (ここでは i 回目の展開) を生成する.

[ステップ4] 分割集合 T^i の各グループ G_j の全ての j の中から隣接する 2^k 個の j よりなる集合の集合として, 隣接集合 R_k が生成される. グループ G_j が4個のときの R_k は次のように表される.

$$R_k = ((0,1,2,3) (0,1) (2,3) (0,2) (1,3))$$

[ステップ5] ステップ3で生成された分割集合 T^i を元に隣接集合 R_k を参照することによって, 共有集合 K^i を生成する.

[ステップ6] 共有集合 K^i に含まれている全ての要素の間で第一吸収判定を行う. 第一吸収判定とは共

有集合 T^i の各グループが他のグループに吸収されるかの判定である. その後に, 第二吸収判定を分割集合 T^i と共有集合 K^i の間で行う. 第二吸収判定とは分割集合 T^i の各グループが共有集合 K^i の各グループのどれかに吸収されるかの判定である.

[ステップ7] 共有集合 K^i または分割集合 T^i の残っている全ての要素についてステップ6の吸収判定後, もし, それらが主項の候補ならば, 第三吸収判定を行う. これは前の展開レベルで既に主項が得られているとき, その主項の候補が吸収されるかどうかの判定である.

[ステップ8] ステップ7の後, 集合 K^i または集合 T^i の残っている全ての要素について隣接判定によって主項かどうか判定される. 主項として判定された要素は記憶後除外される. そして, 残っている要素は結合し, それらは $(i+1)$ 次展開のための新しい集合 S_j^{i+1} として用いられる.

[ステップ9] ステップ8の後, 幾つかの要素が主項の候補でもなく, 集合 K^i 又は集合 T^i に残っている他の要素に吸収されないとき, 次の展開はステップ2に戻って行われる. 又, (1)集合 K^i と T^i の両方に次の展開のための要素がない場合 (2)全ての n ビットに対して展開が終わり, 分割ビットが無くなった場合には, このアルゴリズムは終了する. 本アルゴリズムによる展開の事例を図1に示す.

3. 結果と検討

この多分枝展開法をC言語で記述し, SUN sparc station 5 で実行し, 生成された主項の数と演算時間を分割ビットの長さ n_b を変化させて測定した. その結果を図2と図3に示す.

これらの測定では, 10変数から23変数の積和形論理関数を対象とした. 10進数の最小項番号を生成し, これを入力として使用した. 図2と図3の横軸は存在し得る全最小項に対する最小項の数の割

合を表す。これらの測定では同変数同濃度で10回
の入力を行い、その平均値を結果とした。

図2より、3つの場合 ($n_b = 1, 2$ 及び3) で生成さ
れた主項の数に関して差異がないことが証明された。
しかも、この結果はこの手法が24又はそれ以上の
変数で、7000又はそれ以上の主項生成に耐えること
を保している。

図3より、演算時間に対して次の結果が得られた。

- (1) 分割ビット数 n_b が2ビット又は3ビットよりも
1ビットの方が演算時間が大きくなった。
- (2) 7変数以下では、分割ビット数 n_b が2ビットと3
ビットでの演算時間の違いはほとんど存在しない。
- (3) 分割ビット数 n_b が2ビット又は3ビットに対す
る演算時間は、1ビットに対する演算時間の約72%
である。

4. 結言

多分枝展開法によって論理関数の全主項が、計算
機上で生成された。結果として、分割ビット数が2
ビット又は3ビットの演算時間は、分割ビット数が
1ビットの演算時間より好結果となった。この手法
は高変数 (例えば15変数以上) そして多くの主項
(例えば2000個以上の主項) に適している。

参考文献

[1] Goto, K., Tatumi, H. and Kobayashi, S.: "Generation of Prime Implicants of Logic Function by Multi-Level-Division-Synthesis Method", Proc. of JTC-CSCC'93, Vol.1, p.409-414, July, 1993.
[2] Goto, K., Menjo, M., Tatumi, H., Ling, X.P. and Takahashi, S.: "Improvement of Prime Implicants Generation of Logic Function by Multi-Level-Synthesis Method", Proc. of the IEEE TENCON'94, Vol.2, p.938, Aug, 1994.

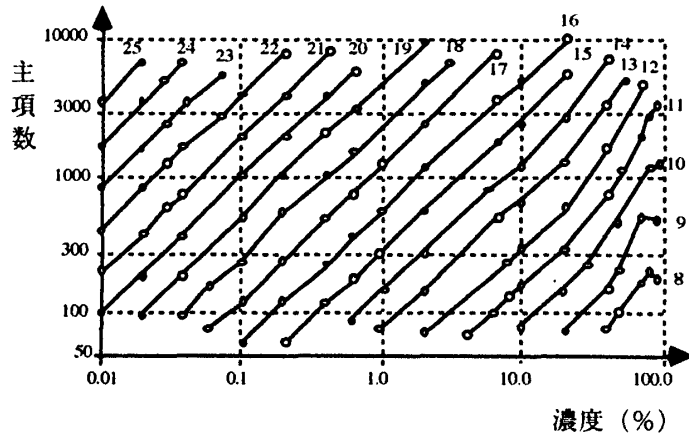


図2. 主項数

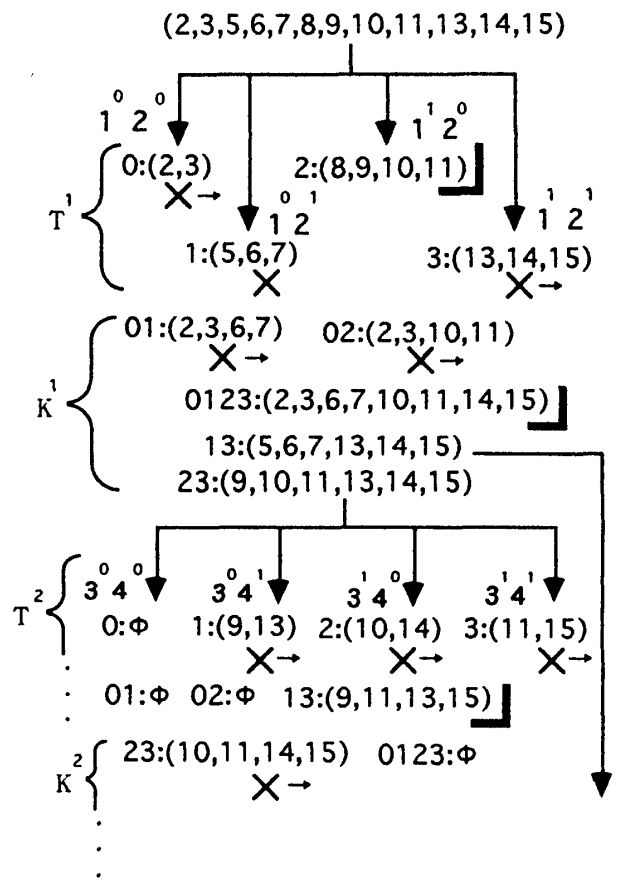


図1. 本アルゴリズムによる展開例

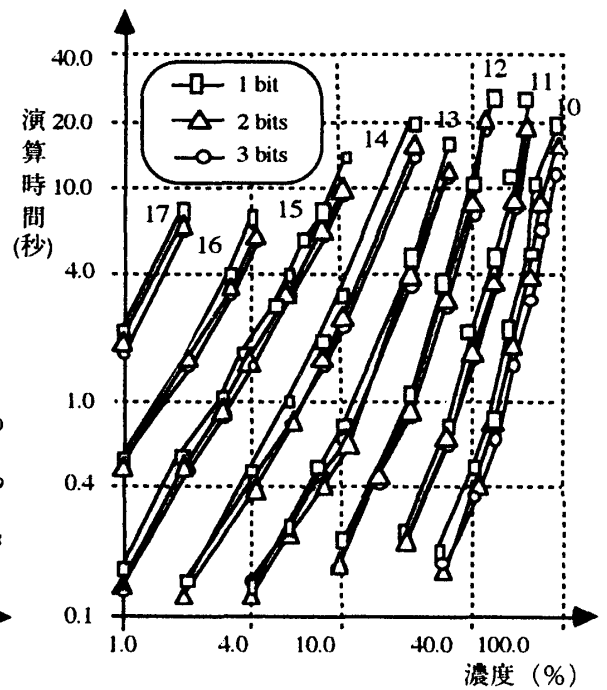


図3. 演算時間