

6H-8

分散記憶型並列計算機に対する
並列二分決定グラフ構成アルゴリズム*

木村晋二

永見康一

奈良先端科学技術大学院大学 情報科学研究科†

1 はじめに

論理回路システム設計の基礎技術である論理関数処理に計算機を援用する手段として、論理関数を計算機上で効率よく表現することができる二分決定グラフ(Binary Decision Diagram; BDD)[1]が考案され、理論と実用の両面から広く研究されている。しかし設計規模の増大にともない、問題によっては多くの計算時間を要する処理が生じ、それらはしばしば逐次計算機的能力を超えるものとなる。本稿では分散記憶型並列計算機を用いてBDDを構成する効率的な手法を示す。分散記憶型の場合、各プロセッサへの負荷分散を処理と記憶領域の両面から考慮することが必要であるが、その手法としてShannon展開によるグラフの分割手法と多出力回路の各出力をプロセッサに振り分ける手法が提案されている[2]。これらを組み合わせた手法について富士通のAP-1000を用いて実現と評価を行なった。

2 準備

BDDは論理関数のグラフ表現であり、変数の順序を固定した二分決定木に対して同型な部分グラフの共有と冗長な節点の削除を施して得られる。共有二分決定グラフ(Shared BDD; SBDD)は部分グラフの共有を異なる関数を表すBDDの間でも行なったものであり、複数の論理関数を少ない記憶領域で表現することができる。さらに記憶領域を低減するために、相補な関数を表すグラフの共有を可能にする表現として否定枝付きBDDが考案されている。

一方、BDDの記憶領域効率は、変数の順序づけによって大きく変化することが知られており、コンパクトなBDDを与える順序づけを求めることはBDDの処理効率を上げるために不可欠である。しかし最小のBDDを与える変数順序(最適順序)は現実的な時間で求めることが困難な場合も多

く、経験則に基づいて近似解を求めることが多い。

3 並列化手法

分散記憶型計算機上でBDDを構成する場合、各プロセッサが独立してSBDDの節点を管理し、プロセッサ間の通信は極力避ける手法が有利である。そこで、求めようとするSBDD全体を何らかの方法で分割し、それぞれを各プロセッサで構成することを考える。分割手法としてShannon展開に基づくものと出力による分割がある。

3.1 Shannon展開による分割

n 変数論理関数 $f(x_1, x_2, \dots, x_n)$ に対して変数 x_i によるShannon展開 $f = x_i \cdot f(x_1, \dots, 1, \dots, x_n) + \bar{x}_i \cdot f(x_1, \dots, 0, \dots, x_n)$ を施すことにより、2つの論理関数 $f_1 = f(x_1, \dots, 1, \dots, x_n)$, $f_0 = f(x_1, \dots, 0, \dots, x_n)$ が得られる。さらには、 m 個の相異なる変数によってこの展開を再帰的に行なうと 2^m 個の関数 $f_{00\dots 0}, f_{00\dots 1}, \dots, f_{11\dots 1}$ による f の表現が得られる。これらの部分関数を表現するBDDを、各プロセッサで並列に構成するのがShannon展開による分割手法である。この手法は、グラフの共有が少ないSBDD(乗算回路がその典型例である)に対しては有効な負荷分散をもたらすが、逆に共有が多い場合には共有しているグラフを異なるプロセッサで同時に構成することになり、並列計算の効果を発揮できないという問題がある。

3.2 出力による分割

多出力の組合せ論理回路を表現するSBDDを構成する場合に、各出力をそれぞれ一つのプロセッサで担当するBDD構成法が考えられる。出力間の独立性が高い場合にはこの手法が有効となるが、逆の場合(典型的にはある出力が他の出力へのファンインになっているような場合)にはこれらの出力を並列に構成することは無意味となる。また、各出力についてそれを構成するために必要な論理演算の数に偏りがある場合には、平均化を考慮する必要がある。

*Parallel BDD Construction Algorithm on a Distributed Memory Multi-Processor System.

†Shinji KIMURA, Kouichi NAGAMI, Nara Institute of Science and Technology

さらに、出力数がプロセッサ数よりも少ない場合には何らかの方法で出力をさらに分割し、プロセッサを有効に活用しなければならない。

3.3 Shannon 展開された出力による分割

前述の二手法を同時に用いるために出力の分割に Shannon 展開を用いる。すなわち、各出力を単一のプロセッサではなくいくつかのプロセッサから成るプロセッサ・グループによって担当させ、グループ内のプロセッサは、Shannon 展開によって担当出力を分割した部分関数の BDD を構成する。例えば図 1 では、2 つの出力関数 f, g を破線枠で囲んだ 2 つのプロセッサ・グループが構成している。各々のグループは $2^3 = 8$ 個のプロセッサからなり、各プロセッサ上には部分関数を表現する BDD が構築されている。

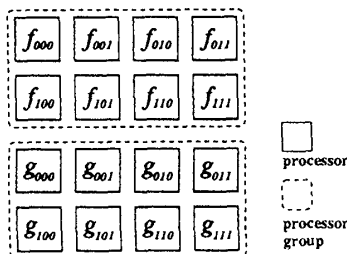


図 1: Shannon 展開された出力による分割

逐次計算機、あるいは共有主記憶型の計算機における SBDD の構成では、良い変数順序 (BDD のサイズが小さくなるような順序) が一般には異なるような関数を共存させることになるため、変数順序の変更による BDD の改良は制約を受けることになる。これに対して我々の手法では、各グループが単一の出力関数に専念できるので、その出力を表す BDD に適した変数順序を他のグループとは独立に選択できるという利点がある。これを活かし、各プロセッサで変数順序の変更処理を施す。今回用いた変数順序づけアルゴリズムは回路の結線情報に基づくもので、出力から入力に向けて深さ優先順に回路をたどり、先にたどり着いた外部入力から順に上位の (すなわちグラフの根に近い方の) 順序をつけるというものである。

4 実現と評価

以上に基づき、回路記述からの並列 BDD 構成プログラムを AP-1000 を用いて実現した。AP-1000 は、セルと呼ばれるプロセッサを最大で 1024 台結合したシステムとして利用することができる分散記憶型高並列計算機である。

まず、並列計算の台数効果を表 1 に示す。ここでは Shannon 展開のみによる分割手法を用いて 16 ビット乗算器の二つの回路記述についてそれぞれ BDD を構成する時間を計測した。“展開数”は Shannon 展開によって分割された部分関数の数である。並列計算によって台数に比例した高速化が得られていることがわかる。

次に提案手法についての実験結果を示す。64 プロセッサのシステムを用いて、ISCAS'89 ベンチマーク回路の BDD を構成する時間を表 2 に示す。“1 台”(セル 1 台での実行)、“S 展開”(Shannon 展開のみによる分割)、“出力”(出力のみによる分割)の欄のデータは文献 [2] より引用している。Shannon 展開と出力分割の、結果の良い方に匹敵する効率が達成されている。

表 1: 乗算器に対する台数効果

回路	展開数	512 台	1024 台
16bit Mult.	8192 (2^{13})	374.4 秒	193.0 秒
C6288	16384 (2^{14})	421.1 秒	210.8 秒

表 2: BDD の構成時間

回路 (出力数)	1 台	S 展開	出力	提案手法
C432 (7)	2.05	1.28	1.73	0.23
C499 (32)	3.01	2.57	0.81	0.86
C880 (26)	0.84	0.41	0.34	0.23
C1355 (32)	8.80	6.33	1.46	0.80
C1908 (25)	5.29	2.26	3.14	1.69

(単位は全て秒)

5 おわりに

以上、分散記憶型並列計算機を用いた BDD 構成について、出力による分割と Shannon 展開とを併用する手法について述べた。そしていくつかの実験結果を示したが、もっと大きな回路に対する並列計算の台数効果を測定し、提案手法の改良余地を探していくつもりである。また、今後はファクタを用いた展開による分割についても考える必要がある。

謝辞 日頃から御討論いただく本学情報科学研究科渡邊研究室の皆様へ感謝致します。また、AP1000 の利用環境を与えて下さる富士通研究所に深謝致します。

参考文献

- [1] R.E.Bryant: Graph-based Algorithms for Boolean Function Manipulation, *IEEE Trans. Comput.*, C-35, No.8, pp.677-691 (Aug. 1986).
- [2] 松本高明, 木村晋二, 羽根田博正: 非共有記憶型並列計算機上での二分決定グラフの並列化とその応用, 情報処理学会第 46 回全国大会, 8L-06 (1993).