Boolean Buchberger Algorithm とその並列化 (2)- 動作解析

岩山 登, 佐藤 健, 毛受 哲, 佐藤洋祐
(財) 新世代コンピュータ技術開発機構

1 はじめに

本稿では, Boolean Grōbner Base を求める Boolean Buchberger Algorithm の逐次マシンでの実現と, その動作解析について報告する.

[1] のアルゴリズムは計算の効率を考慮していなかった. ここで示すアルゴリズムでは, ヒューリスティクスを導入して高速化を図っている.

また, この逐次アルゴリズムを並列化するために, 各処理の大きさ, すなわち, 処理時間を調べた. 並列性がどの程度あるか, どの処理部分を並列化すればよいかの, 検討材料にするためである.

2 Boolean Buchberger Algorithm

ここでは, 逐次マシン上に実現した Boolean Buchberger Algorithm を示す. [1] で述べられているアルゴリズムは, 一般のブール代数に対してのものである. ここでは, 係数が $\{1,0\}$ となるブール多項式の場合 (0 の項は現れない) について述べる.

input EQlist: ブール多項式のリスト
$GB := \emptyset$
while $EQlist \neq nil$

1. $choose\_rule(EQlist, p)$
2. $reduce\_all(p, GB, poly)$
3. $reducible\_rule(poly, GB, NewGB, Can1)$
4. $reducible\_can(EQlist, NewGB, RestEQlist, Can2)$
5. $make\_cp(poly, NewGB, Can3)$
6. $reduce\_can(NewGB, Can1, Can2, Can3, NewEQlist)$
7. $EQlist := NewEQlist + RestEQlist$
   $GB := NewGB$

endwhile
output GB

1 から 7 の各処理の詳細

1. $EQlist$ 中の, 最大単項が最小の多項式を p とする.
2. p の最大単項以外の部分を GB で書き換え, $poly$ とする.
3. GB の各多項式を $poly$ で書き換える. 最大単項が書き換えられない多項式は最大単項以外の項を書き換えて $NewGB$ に, 最大単項が書き換え可能な多項式は $Can1$ に入れる. さらに, $poly$ を $NewGB$ に入れる.
4. $EQlist$ の各多項式の最大単項が $NewGB$ で書き換えられるかどうか調べる. 書き換えられない多項式は $RestEQlist$ に, 書き換え可能な多項式は $Can2$ に入れる.
5. $poly$ の自己要対と, $NewGB$ の各多項式と $poly$ の要対を $Can3$ に入れる.
6. $Can1, Can2, Can3$ の各多項式を $NewGB$ で書き換え, 書き換え結果が 0 以外ならば $NewEQlist$ に入れる (ここでの書き換えは, 最大単項に対し書き換えられるだけ行う).
7. $NewEQlist$ と $RestEQlist$ を append して $EQlist$ とする (差分リストとして実現されているので, コストはほとんどかからない).

上に示したアルゴリズムと, [1] のアルゴリズムの本質的な違いは, $choose\_rule$ での多項式の選択である. $choose\_rule$ では, p 以外を選んでもアルゴリズムは正しく動くが, ヒューリスティクスとして p を選ぶ.

3 アルゴリズムの動作解析

ここでは, 動作解析の方法と, 解析に用いた問題について述べる.

前節で述べたアルゴリズムの, 1 から 6 までの各処理の前後に, 計時のためのコードを埋め込み, while ループの各周ごとの, それぞれの処理に要した時間を測った. また, それぞれの処理ごとに計算時間を足し合わせて, それぞれの処理にかかる時間の全計算時間に対する割合を調べた.

なお, プログラムは ESP で記述し, PSI-II 上で計測を行った.

解析に用いた問題は, 論理回路問題とクィーン問題である. 論理回路問題は, 入力された 1 の個数を 2 進数で答える問題で, 31 変数 26 方程式からなる (詳しくは, [2] を参照されたい). また, クィーン問題は, 以下に示すようにブール式で表現した.

Boolean Buchberger Algorithm and its Parallelization(2) Analysis of Sequential Algorithm
Noboru Iwayama, Ken Satoh, Satoshi Menju, Yosuke Sato
Institute for New Generation Computer Technology

問題	1	2	3	4	5	6	計算時間
4queens	25.8	3.2	8.4	17.3	25.4	19.0	1.8
5queens	6.4	3.4	22.3	3.7	14.5	50.4	53.5
6queens	1.0	1.5	15.0	2.0	2.6	77.7	2240.0
回路	2.1	4.2	8.2	3.3	8.0	74.0	70.7

表 1: 各処理の割合 (%) と計算時間 (秒)

4 クィーンの場合

$x11 \wedge x12 = 0, x11 \wedge x13 = 0, x11 \wedge x14 = 0,$
$x12 \wedge x13 = 0, x12 \wedge x14 = 0, x13 \wedge x14 = 0,$
$x21 \wedge x22 = 0, x21 \wedge x23 = 0, x21 \wedge x24 = 0,$

(略)

$x23 \wedge x32 = 0, x23 \wedge x34 = 0, x24 \wedge x42 = 0,$
$x24 \wedge x33 = 0, x31 \wedge x42 = 0, x32 \wedge x41 = 0,$
$x32 \wedge x43 = 0, x33 \wedge x42 = 0, x33 \wedge x44 = 0,$
$x34 \wedge x43 = 0,$
$x11 \vee x12 \vee x13 \vee x14 = 1, x21 \vee x22 \vee x23 \vee x24 = 1,$
$x31 \vee x32 \vee x33 \vee x34 = 1, x41 \vee x42 \vee x43 \vee x44 = 1$

4 解析結果

ここでは、解析の結果について述べる.

表 1では、問題ごとに、1 から 6 までの各処理でかかっている時間を全計算時間に対する割合で示した.

表を見ると、4 クィーン問題を除き、6 の、多項式を GB で書き換える処理が 50% から 70% を占めている. このことから、6 の *reduce_can* の並列化は有望であると考えられる.

5 解析結果その 2

処理 3、4、5 は互いに独立に行うことができる. また、*Can*1、*Can*2、*Can*3 の書き換えも互いに独立に行うことができる. そこで、2 節のアルゴリズムを書き直した以下のアルゴリズムについて考える.

input EQlist: ブール多項式のリスト
$GB := \emptyset$
while $EQlist \neq nil$

1. $choose\_rule(EQlist, p)$
2. $reduce\_all(p, GB, poly)$
3. *reducible_rule* 及び *Can*1 の書き換え
4. *reducible_can* 及び *Can*2 の書き換え
5. *make_cp* 及び *Can*3 の書き換え
6. $EQlist := NewEQlist + RestEQlist$
   $GB := NewGB$

endwhile
output GB

このアルゴリズムでは、3, 4, 5 を並列に実行することができる. もし、それらの 3 つの処理が同程度の時間で実行できるならば、並列化の効果が期待できる. 表 2に、変更したアルゴリズムの、3, 4, 5 の各処理でかかった時間を全計算時間に対する割合で示した. これをみると、3, 4, 5 の処理にかかる時間はバランスしているとは言えない. つまり、このアルゴリズムの 3, 4, 5 を並列化するのは、適当と言えない.

問題	3	4	5
4queens	12.1	23.5	36.4
5queens	24.7	11.2	54.0
6queens	15.5	39.9	41.9
回路	8.2	33.5	51.7

表 2: 変更したアルゴリズムの各処理の割合 (%)

6 まとめ

Boolean Gröbner Base を求める Boolean Buchberger Algorithm の逐次アルゴリズムの説明と、その動作解析を行った. Boolean Buchberger Algorithm は、主に等式の書き換え、要対の生成、ルールの変更の 3 つの部分からなる. PSI-II 上での動作解析の結果、等式の書き換えが処理全体の 70% 程度と大きな部分を占めることがわかった.

参考文献

[1] 佐藤洋祐 他, Boolean Buchberger Algorithm とその並列化 (1)– 基礎理論, 情報処理学会第 43 回全国大会論文集, 7N-3.

[2] 相場 亮, これからの高レベル言語は?, 数理科学, No.313, pp. 50 – 57, July, 1989.