ブール多項式環 $\mathbb{Z}_2(x_1,\ldots,x_n)$ におけるイデアル $\langle g\rangle$ のブーリアングレブナー基底の計算について

Computation of a Boolean Gröbner Basis for an Ideal $\langle g\rangle$ in the Boolean Polynomial Ring $\mathbb{Z}_2(x_1,\ldots,x_n)$

芝浦工業大学 大学院理工学研究科　　佐川 嘉信 [*1]

Yoshinobu Sagawa

Graduate School of Engineering and Science, Shibaura Institute of Technology

芝浦工業大学 システム理工学部　　井戸川 知之 [*2]

Tomoyuki Idogawa

College of Systems Engineering and Science, Shibaura Institute of Technology

Abstract

Boolean Gröbner bases are studied mainly in connection with cryptanalysis and formal verification [2]. Every Boolean Gröbner basis can be constructed as a subset of the corresponding (non Boolean) Gröbner basis. Therefore many methods to compute Boolean Gröbner bases are based on the ones for usual Gröbner bases (e.g. Buchberger's algorithm). In this paper we propose an algorithm that compute a Boolean Gröbner basis for an ideal generated by a given Boolean polynomial over $\mathbb{Z}_2(x_1,\ldots,x_n)$. We implemented the algorithm with using binary decision diagram (and zero-suppressed one) for the data structure of Boolean polynomials. Numerical experiments in the article imply that the proposed method and its implementation is more efficient than traditional ones in some cases.

1　はじめに

環 R を係数環, $x_1,\ldots,x_n$ を変数とした多項式環を $R[x_1,\ldots,x_n]$ で表す. また, $f_1,\ldots,f_s\in R[x_1,\ldots,x_n]$ の生成するイデアルを $\langle f_1,\ldots,f_s\rangle$ で, $F\subseteq R[x_1,\ldots,x_n]$ を基底とするイデアルを $\langle F\rangle$ でそれぞれ表す. ある単項式順序が与えられたとき, $f\in R[x_1,\ldots,x_n]$ の先頭項, 先頭単項式, 先頭係数をそれぞれ $\mathrm{LT}(f)$, $\mathrm{LM}(f)$, $\mathrm{LC}(f)$ で表す. ここで項とは, 非零係数 $a\in R$ と単項式 $x_1^{\alpha_1}\cdots x_n^{\alpha_n}$ の積 $a\cdot x_1^{\alpha_1}\cdots x^{\alpha_n}$ のことをいう. また, $F\subseteq R[x_1,\ldots,x_n]$ に対して, $\mathrm{LT}(F)=\{\mathrm{LT}(f)\mid f\in F\backslash\{0\}\}$ で定義する.

1.1　ブール環係数多項式環上のグレブナー基底

以下, $\mathbb{B}$ をブール環とする. $\mathbb{B}[x_1,\ldots,x_n]$ 上でもグレブナー基底を考えることができる.

[*1] mf19034@sic.shibaura-it.ac.jp

[*2] idogawa@sic.shibaura-it.ac.jp

定義 1 (グレブナー基底)
$I \subseteq \mathbb{B}[x_1, \ldots, x_n]$ をイデアルとする. 有限な $G \subseteq I$ が $\langle \mathrm{LT}(G) \rangle = \langle \mathrm{LT}(I) \rangle$ を満たすとき, G は I のグレブナー基底であるという.

$\mathbb{B}[x_1, \ldots, x_n]$ 上のグレブナー基底の構成に際して, 以下を考える.

定義 2 (ブール閉包)
$f \in \mathbb{B}[x_1, \ldots, x_n] \backslash \{0\}$ とする. $\mathrm{LC}(f) \cdot f$ を f のブール閉包といい, $\mathrm{bc}(f)$ で表す. $f = \mathrm{bc}(f)$ であるとき, f はブール閉であるという.

このとき, ブッフバーガーの判定条件にあたる以下の定理が成り立つ.

定理 3
$G \subseteq \mathbb{B}[x_1, \ldots, x_n]$ を, すべての $g \in G$ がブール閉な有限集合とする. このとき, G がグレブナー基底であることの必要十分条件は, G のすべての S ペアが G により 0 に簡約されることである.

ブール閉包を考慮に入れたブッフバーガーアルゴリズムの拡張によって, 有限集合 $F \subseteq \mathbb{B}[x_1, \ldots, x_n]$ から, $\langle F \rangle$ のブール閉なグレブナー基底 $G \subseteq \mathbb{B}[x_1, \ldots, x_n]$ を構成できる. すなわち, 非零な正規形 r の代わりに $\mathrm{bc}(r)$ を基底に加え, 非零な $r - \mathrm{bc}(r)$ を基底の候補に加えればよい.

1.2 ブーリアングレブナー基底

定義 4 (ブール多項式環)
剰余環 $\mathbb{B}(x_1, \ldots, x_n) = \mathbb{B}[x_1, \ldots, x_n] / \left\langle x_1^2 - x_1, \ldots, x_n^2 - x_n \right\rangle$ はそれ自身ブール環となり, ブール多項式環という. また, $\mathbb{B}(x_1, \ldots, x_n)$ の元をブール多項式という.

ブール多項式の代表元として, 各変数についての次数が 1 以下であるものをとることができる. 以降, ブール多項式としてこのような元のみを考える. $\mathbb{B}[x_1, \ldots, x_n]$ 上の単項式順序が与えられたとき, $f \in \mathbb{B}(x_1, \ldots, x_n)$ に対して, $\mathrm{LT}(f)$, $\mathrm{LM}(f)$, $\mathrm{LC}(f)$ は $\mathbb{B}[x_1, \ldots, x_n]$ 上と同様に定義される.

定義 5 (ブーリアングレブナー基底)
$I \subseteq \mathbb{B}(x_1, \ldots, x_n)$ をイデアルとする. 有限な $G \subseteq I$ が $\langle \mathrm{LT}(G) \rangle = \langle \mathrm{LT}(I) \rangle$ を満たすとき, G は I のブーリアングレブナー基底であるという.

ブーリアングレブナー基底は $\mathbb{B}[x_1, \ldots, x_n]$ 上のグレブナー基底から構成できることが知られている.

定理 6
単項式順序を固定する. 有限な $F \subseteq \mathbb{B}(x_1, \ldots, x_n)$ に対して, $F \cup \{x_1^2 - x_1, \ldots, x_n^2 - x_n\} \subseteq \mathbb{B}[x_1, \ldots, x_n]$ のグレブナー基底を G としたとき, $G \backslash \{x_1^2 - x_1, \ldots, x_n^2 - x_n\} \subseteq \mathbb{B}(x_1, \ldots, x_n)$ は $\langle F \rangle$ のブーリアングレブナー基底である.

ブーリアングレブナー基底計算の既存手法の多くは, これに従ってグレブナー基底計算を行っている.

通常のグレブナー基底と同様, ブーリアングレブナー基底 G が極小性 $(\forall g \in G[\mathrm{LT}(g) \notin \langle \mathrm{LT}(G \backslash \{g\}) \rangle])$ を満たすとき, G を極小ブーリアングレブナー基底といい, G が既約性 $(\forall g \in G[g \text{ は } G \backslash \{g\} \text{ に対して既約}])$ を満たすとき, G を簡約ブーリアングレブナー基底という. 通常のグレブナー基底と異なり, 極小ブーリアングレブナー基底はその元の数の意味で必ずしも最小ではなく, 簡約ブーリアングレブナー基底の一意性もまた保証されない. ただし, $\mathbb{B} = \mathbb{Z}_2$ の場合, これらは保証される.

2　$\mathbb{B}(x_1,\ldots,x_n)$ 上のブーリアングレブナー基底計算

$\mathbb{B}(x_1,\ldots,x_n)$ 上の計算を用いたブーリアングレブナー基底構成について考える．定理 6 によれば，ブーリアングレブナー基底は $\mathbb{B}[x_1,\ldots,x_n]$ 上のグレブナー基底計算により構成することができるが，ここでは $\mathbb{B}(x_1,\ldots,x_n)$ 上で直接ブーリアングレブナー基底を求めることによる計算の簡略化をねらいとしている．

$\mathbb{B}(x_1,\ldots,x_n)$ 上で S ペア，簡約を $\mathbb{B}[x_1,\ldots,x_n]$ 上と同様に定義することはできる．しかし，($\mathbb{B}[x_1,\ldots,x_n]$ 上に拡張した) ブッフバーガーアルゴリズムをそのまま適用しても，ブーリアングレブナー基底は求まらない．

例 1
$x>y>z$ の辞書式順序を用いる．$f_1=xy+y+1$, $f_2=z+1\in\mathbb{Z}_2(x,y,z)$ とする．このとき，

$$S(f_1,f_2)=z\cdot f_1-xy\cdot f_2=xy+yz+z\to_{f_1} yz+y+z+1\to_{f_2} z+1\to_{f_2} 0.$$

ここで，$\to_f$ は f による簡約を表す．一方で，$\{f_1,f_2\}\subseteq\mathbb{Z}_2(x,y,z)$ はブーリアングレブナー基底ではない．実際，$x\cdot f_1=x$ であり，$\mathrm{LT}(f_1)$, $\mathrm{LT}(f_2)$ のいずれもこの先頭項 x を割り切らない．

通常の多項式環では，任意の多項式 f に対して，$\{f\}$ は $\langle f\rangle$ のグレブナー基底であり，ブッフバーガーアルゴリズムはこれを前提としている．$\mathbb{B}[x_1,\ldots,x_n]$ 上や $\mathbb{B}(x_1,\ldots,x_n)$ 上ではこの前提が一般に成り立たないため，そのままブッフバーガーアルゴリズムを適用してもグレブナー基底が求まらない．一方で，$\mathbb{B}[x_1,\ldots,x_n]$ 上のブール閉な元は先の前提を満たす．このため，ブール閉を考慮に入れることで，ブッフバーガーアルゴリズムを $\mathbb{B}[x_1,\ldots,x_n]$ 上に適応させることができた．実は，$f\in\mathbb{B}[x_1,\ldots,x_n]$ がブール閉であることと $\{f\}$ が $\langle f\rangle$ のグレブナー基底であることは同値であり，定理 3 は以下のように書き換えられる．

系 7
$G\subseteq\mathbb{B}[x_1,\ldots,x_n]$ を有限集合とし，任意の $g\in G$ に対し，$\{g\}$ が $\langle g\rangle$ のグレブナー基底であるとする．このとき，G がグレブナー基底であることの必要十分条件は，G のすべての S ペアが G により 0 に簡約されることである．

このように書き換えると，$\mathbb{B}(x_1,\ldots,x_n)$ 上でも同様のことがいえる．

系 8
$G\subseteq\mathbb{B}(x_1,\ldots,x_n)$ を有限集合とし，任意の $g\in G$ に対し，$\{g\}$ が $\langle g\rangle$ のブーリアングレブナー基底であるとする．このとき，G がブーリアングレブナー基底であることの必要十分条件は，G のすべての S ペアが G により 0 に簡約されることである．

したがって，$\mathbb{B}(x_1,\ldots,x_n)$ 上で，$\mathbb{B}[x_1,\ldots,x_n]$ 上のブール閉化にあたる操作，すなわち，ある多項式 $f\in\mathbb{B}(x_1,\ldots,x_n)$ から，$\mathrm{LT}(g)=\mathrm{LT}(f)$ かつ $\{g\}$ が $\langle g\rangle$ のブーリアングレブナー基底となるような $g\in\langle f\rangle$ を求める操作によって，$\mathbb{B}[x]$ 上と同様の，$\mathbb{B}(x_1,\ldots,x_n)$ 上に対応したブッフバーガーアルゴリズムの拡張が可能になる．

例えば，係数環 $\mathbb{Z}_2$，辞書式順序の場合に限定して，以下の定理が得られた．ここで，単項式順序とは異なる $\mathbb{B}(x_1,\ldots,x_n)$ 上の半順序関係 $f\preceq g$ を $fg=f$ で定義する．

定理 9
$x_1>\cdots>x_n$ の辞書式順序を用いる．$g\in\mathbb{Z}_2(x_1,\ldots,x_n)\backslash\{0\}$ とし，$p=g(1,x_2,\ldots,x_n)$, $q=g(0,x_2,\ldots,x_n)$ とおく．さらに，$p\neq q$ を仮定する．このとき，$\{g\}$ が $\langle g\rangle$ のブーリアングレブナー基底であることと，$\{p+q\}$ が $\langle p+q\rangle$ のブーリアングレブナー基底，かつ $pq=0$ であることは同値である．

証明 $g = x_1p + (x_1+1)q = x_1(p+q) + q$ と表せる. 特に $\mathrm{LT}(g) = x_1 \cdot \mathrm{LT}(p+q)$. また, $pqg = pq$ より, $pq \preceq g$. これは $pq \in \langle g \rangle$ と同値であることに注意する.

$\Rightarrow$: $\{g\}$ が $\langle g \rangle$ のブーリアングレブナー基底であるとする. $\{p+q\}$ が $\langle p+q \rangle$ のブーリアングレブナー基底でないとき, $\mathrm{LT}(p+q) \nmid \mathrm{LT}(h)$ なる $h \in \langle p+q \rangle$ が存在する. このとき, $\mathrm{LT}(hg) = x_1 \cdot \mathrm{LT}(h)$ であり, $\mathrm{LT}(g) \nmid \mathrm{LT}(hg)$. これは矛盾. $pq \neq 0$ のとき, 明らかに $\mathrm{LT}(g) \nmid \mathrm{LT}(pq)$ であり, 矛盾.

$\Leftarrow$: $\{p+q\}$ が $\langle p+q \rangle$ のブーリアングレブナー基底かつ $pq = 0$ とする. $f \in \langle g \rangle$ を $\mathrm{LT}(g) \nmid \mathrm{LT}(f)$ なる元とし, $p' = f(1, x_2, \ldots, x_n)$, $q' = f(0, x_2, \ldots, x_n)$ とおく. $f \preceq g$ より, $p' \preceq p$ かつ $q' \preceq q$ であり, 特に $0 \neq p' + q' \in \langle p+q \rangle$. このとき, $\mathrm{LT}((p+q)f) = \mathrm{LT}(f) = x_1\mathrm{LT}(p'+q')$ だが, $\mathrm{LT}(g) \nmid \mathrm{LT}(f)$ より $\mathrm{LT}(p+q) \nmid \mathrm{LT}((p+q)f)$ が従う. これは矛盾. ∎

この定理より, 与えられた $f \in \mathbb{Z}_2(x_1, \ldots, x_n)$ から, 求める $g \in \langle f \rangle$ を再帰的に構成する手法が従う. また, この操作を用いたブッフバーガーアルゴリズムの拡張によって, 辞書式順序に関する $\mathbb{Z}_2(x_1, \ldots, x_n)$ 上のブーリアングレブナー基底が構成できる.

3 提案手法

定理 9 に付随して, 単集合 $\{g\} \subseteq \mathbb{Z}_2(x_1, \ldots, x_n)$ からブーリアングレブナー基底を構成する新たな手法が得られた. これは, $\langle g \rangle$ から, 各先頭項に対して定理 9 の性質を満たす多項式を取り出すというものである. $\mathbb{Z}_2(x_1, \ldots, x_n)$ は単項イデアル環であり, $f \vee g = f + g + fg$ として, $\langle f_1, \ldots, f_s \rangle = \langle f_1 \vee \cdots \vee f_s \rangle$ であることが知られている. したがって, この前処理を含めれば, この手法は $\mathbb{Z}_2(x_1, \ldots, x_n)$ 上の任意のイデアルに対して適用可能である.

以降, 定理 9 と同様に, 係数環を $\mathbb{Z}_2$ とし, $x_1 > \cdots > x_n$ の辞書式順序を用いる.

定理 10
$g \in \mathbb{Z}_2(x_1, \ldots, x_n) \backslash \{0\}$ に対し, $p = g(1, x_2, \ldots, x_n)$, $q = g(0, x_2, \ldots, x_n)$ とおく. 加えて, G_1, G_2 をそれぞれ $\langle pq \rangle$, $\langle p \vee q \rangle \subseteq \mathbb{Z}_2(x_2, \ldots, x_n)$ のブーリアングレブナー基底とすると,

$$G = G_1 \cup \{x_1h + qh \mid h \in G_2\}$$

は $\langle g \rangle$ のブーリアングレブナー基底である.

証明 g は $g = x_1p + (x_1+1)q$ の形をとる. $pq \cdot g$, $(x_1+q)(p \vee q) \cdot g$ をそれぞれ計算して, $pq \preceq g$, $(x_1+q)(p \vee q) \preceq g$ を得る. $G \backslash G_1 \subseteq \langle (x_1+q)(p \vee q) \rangle$ に注意. したがって $G \subseteq \langle g \rangle$.

任意の $f \in \langle g \rangle$ をとる. $p' = f(1, x_2, \ldots, x_n)$, $q' = f(0, x_2, \ldots, x_n)$ とおき, f を g と同様の形に分解する. このとき, f は $f = x_1p' + (x_1+1)q' = x_1(p'+q') + q'$ の形をとる. $f \preceq g$ より $p' \preceq p$, $q' \preceq q$, すなわち $p' \in \langle p \rangle$, $q' \in \langle q \rangle$ がそれぞれ従う. $x_1 \nmid \mathrm{LT}(f)$ のとき, $f = p' = q'$ であり, 特に $f \in \langle pq \rangle = \langle G_1 \rangle$. G_1 はブーリアングレブナー基底であるから, $\mathrm{LT}(g_1) \mid \mathrm{LT}(f)$ を満たす $g_1 \in G_1 \subseteq G$ が存在する. 他方, $x_1 \mid \mathrm{LT}(f)$ のとき, $p' \neq q'$ であり, $\mathrm{LT}(f) = x_1\mathrm{LT}(p'+q')$. $p'+q' \in \langle p, q \rangle = \langle p \vee q \rangle = \langle G_2 \rangle$ に注意する. G_2 はブーリアングレブナー基底であるから, $\mathrm{LT}(h) \mid \mathrm{LT}(p'+q')$ を満たす $h \in G_2$ が存在する. ここで, $g_2 = x_1h + qh \in G$ とおくと, h が x_1 を含まないことから $\mathrm{LT}(g_2) = x_1\mathrm{LT}(h)$ がいえる. 特に, $\mathrm{LT}(g_2) = x_1\mathrm{LT}(h) \mid x_1\mathrm{LT}(p'+q') = \mathrm{LT}(f)$.

ゆえに, G は $\langle g \rangle$ のブーリアングレブナー基底である. ∎

$\mathbb{B} = \mathbb{Z}_2$ のとき, $\mathbb{Z}_2(x_1, \ldots, x_n)$ と $\mathbb{Z}_2^n$ から $\mathbb{Z}_2$ へのブール関数全体との間に対応が存在し, $f \in \mathbb{Z}_2(x_1, \ldots, x_n)$ を n 変数論理式とみなすことができる. この定理は, $\langle f \rangle$ のブーリアングレブナー基底を, f の部分論理式を用いて再帰的に構成できることを主張している.

定理 10 に基づいたブーリアングレブナー基底構成は, 一般に, その元の数の意味でも, その元の持つ項の意味でも冗長さを伴う. 以下の系によって, 極小性, 既約性を保った構成が可能になる.

系 11
定理 10 の条件に加えて, G_1, G_2 の極小性を仮定すると,

$$G = G_1 \cup \{x_1 h + qh \mid h \in G_2,\ \mathrm{LT}(h) \notin \mathrm{LT}(G_1)\}$$

は $\langle g \rangle$ の極小ブーリアングレブナー基底である.

系 12
定理 10 の条件に加えて, G_1, G_2 の既約性を仮定すると,

$$G = G_1 \cup \{x_1 h + \mathrm{nf}(qh, G_1) \mid h \in G_2,\ \mathrm{LT}(h) \notin \mathrm{LT}(G_1)\}$$

は $\langle g \rangle$ の簡約ブーリアングレブナー基底である. ここで, $\mathrm{nf}(qh, G_1)$ は qh の G_1 に関する正規形を表す.

証明 いずれの場合も, G_1 の任意の元は x_1 を含まず, $G_2' := G \backslash G_1$ の任意の元は x_1 を含む先頭項を持つ. したがって, $g \in G_1$ のもつ極小性 (既約性) は保たれる. 残るは $g \in G_2'$ に対して g の持つべき性質を確認することである. G_2 の極小性 (既約性) から, これは $\mathrm{LT}(g) \notin \mathrm{LT}(G_1)$ (g が G_1 に対して既約) であることを示せばよい. これは, G_2' の定義から従う. ∎

系 12 から従うブーリアングレブナー基底構成手法を以下に示す.

アルゴリズム 1 再帰的ブーリアングレブナー基底構成 $\textsc{Recbgb}(g, x_1, \ldots, x_n)$

Require: $g \in \mathbb{Z}_2(x_1, \ldots, x_n)$, variables $x_1, \ldots, x_n$.
Ensure: $G \subseteq \langle g \rangle$ the reduced Boolean Gröbner basis for $\langle g \rangle$ w.r.t. lex order on $x_1 > \cdots > x_n$.

```
1: if g ∈ Z_2 then G := {g}\{0}
2: else
3:     G_1 := ∅; G_2 := ∅
4:     p := g(1, x_2, ..., x_n); q := g(0, x_2, ..., x_n)
5:     G_1 := Recbgb(pq, x_2, ..., x_n)
6:     if p ≠ q then
7:         G_2 := Recbgb(p ∨ q, x_2, ..., x_n)
8:         for all h ∈ G_2 do
9:             if LT(h) ∉ LT(G_1) then
10:                G_2 := G_2 ∪ {x_1 h + nf(qh, G_1)}
11:            end if
12:        end for
13:    end if
14:    G := G_1 ∪ G_2
15: end if
16: return G
```

停止性は再帰呼び出し時の変数の減少性による.

また, 定理 10 により得られたブーリアングレブナー基底の構造から, 正規形に対しても再帰的な構成が可能になる.

定理 13
$f, g \in \mathbb{Z}_2(x_1, \ldots, x_n) \backslash \{0\}$ に対し, $\phi_1, \phi_2, \psi_1, \psi_2 \in \mathbb{Z}_2(x_2, \ldots, x_n)$ が $f = x_1\phi_1 + \phi_2$, $g = x_1\psi_1 + (x_1 + 1)\psi_2$ を満たすとする. このとき, $r_1 = \mathrm{nf}(\phi_1, \langle \psi_1 \vee \psi_2 \rangle)$, $r_2 = \mathrm{nf}(\phi_2 + \psi_2(\phi_1 + r_1), \langle \psi_1 \psi_2 \rangle)$ とおくと, $\mathrm{nf}(f, \langle g \rangle) = x_1 r_1 + r_2$.

証明 まず, $r = x_1 r_1 + r_2$ の $\langle g \rangle$ に対する既約性を示す. 明らかに $x_1 r_1$ と r_2 は共通項を持たず, $x_1 r_1$ と r_2 の既約性をそれぞれ示せばよい. r_1 の $\langle \psi_1 \vee \psi_2 \rangle$ に対する既約性は, $x_1 r_1$ の $\langle \psi_1 \vee \psi_2 \rangle$ に対する既約性を伴う. $\langle g \rangle \subseteq \langle \psi_1 \vee \psi_2 \rangle$ に注意して, $x_1 r_1$ は $\langle g \rangle$ に対して既約である. r_2 は $\langle \psi_1 \psi_2 \rangle$ に対して既約であり, 定理 10 から, このとき r_2 は $\langle g \rangle$ に対しても既約である.

次に, $r = f + h$ なる $h \in \langle g \rangle$ の存在を示す. r_1 の定義から, $r_1 = \phi_1 + h_1$ なる $h_1 \in \langle \psi_1 \vee \psi_2 \rangle$ が存在し, r_2 に対しても同様に $r_2 = (\phi_2 + \psi_2(\phi_1 + r_1)) + h_2$ なる $h_2 \in \langle \psi_1 \psi_2 \rangle \subseteq \langle g \rangle$ が存在する. このとき,

$$r = x_1(\phi_1 + h_1) + (\phi_2 + \psi_2(\phi_1 + r_1)) + h_2 = f + (x_1 + \psi_2)h_1 + h_2.$$

定理 10 の証明でもみたとおり $(x_1 + \psi_2)h_1 \in \langle g \rangle$ であり, $h = (x_1 + \psi_2)h_1 + h_2 \in \langle g \rangle$. ∎

定理 2 から従う簡約手法を以下に示す.

アルゴリズム 2 再帰的簡約手法 $\mathrm{nf}(f, g, x_1, \ldots, x_n)$

Require: $f, g \in \mathbb{Z}_2(x_1, \ldots, x_n)$, variables $x_1, \ldots, x_n$.
Ensure: $r \in \mathbb{Z}_2(x_1, \ldots, x_n)$ the normal form of f for $\langle g \rangle$ w.r.t. lex order on $x_1 > \cdots > x_n$.

1: **if** $f = 0$ **or** $g = 1$ **or** $f = g$ **then** $r := 0$
2: **else if** $f = 1$ **or** $g = 0$ **then** $r := f$
3: **else if** $f = g + 1$ **then** $r := 1$
4: **else**
5: $\quad \phi_2 := f(0, x_2, \ldots, x_n)$; $\phi_1 := (f - \phi_2)/x_1$
6: $\quad \psi_1 := g(1, x_2, \ldots, x_n)$; $\psi_2 := g(0, x_2, \ldots, x_n)$
7: $\quad r_1 := \mathrm{nf}(\phi_1, \psi_1 \vee \psi_2, x_2, \ldots, x_n)$
8: $\quad r_2 := \mathrm{nf}(\phi_2 + \psi_2(\phi_1 + r_1), \psi_1 \psi_2, x_2, \ldots, x_n)$
9: $\quad r := x_1 r_1 + r_2$
10: **end if**
11: **return** r

アルゴリズム 1 と同様, 停止性は再帰呼び出し時の変数の減少性による. この手法は, アルゴリズム 1 の 10 行目における正規形計算に用いることができる. すなわち, すでに計算された pq を用いて $\mathrm{nf}(qh, pq, x_1, \ldots, x_n)$ とすることで, qh の G_1 に対する正規形が計算できる.

4 計算実験

ここでは, 3 章の結果を元にした実装, 計算実験について述べる.

C++によるプログラムで, アルゴリズム 1, 2 を実装した. 多項式を表すデータ構造として Binary Decision Diagram (BDD) [1], Zero-suppressed BDD (ZBDD) [6] を用いた. BDD とは論理関数を効率的に扱う構造であり, 部分関数の演算, 論理和, 論理積演算を多用し, 再帰的な単項演算, 二項演算を行う本手法と相性がよい. このため, 主な計算はすべて BDD を用いて行っている. ZBDD とは集合族を表す構造であり, ブール単項式を $\{x_1, \ldots, x_n\}$ の部分集合とみなし, ブール多項式をその族として表すことができる. BDD はブール多項式の論理関数としての表現であり, 多項式の項を多く参照する処理は ZBDD を用いたほうが効率がよい. 本実装では入出力を行う際に ZBDD を用いている. BDD, ZBDD の処理系として SAPPORO BDD ライブラリ [9] を用いた.

アルゴリズム 1 の 10 行目における正規形計算について, この時点で $\langle pq \rangle$ のブーリアングレブナー基底は計算されており, 通常の多項式除算を行うこともできる. しかし, 今回の実装では, 多くの場合でアルゴリズム 2 を用いた方が効率的であり, この再帰簡約手法を採用した.

計算例として, まず life[5] を用いた. これは, 対称多項式によって定義されるブール多項式であり, 単集合からブーリアングレブナー基底計算を行う提案手法に合致した例である. この他に cyclic, MQ Problem (以下 MQP と略記) Type I, IV[8] を用いた. これらはいずれも $\mathbb{Z}_2$ 係数のブール多項式とみなして計算を行った. これらは複数の多項式によって定義されるため, これを $f_1, \ldots, f_s$ としたとき, アルゴリズム 1 の実行前に $g = f_1 \vee \cdots \vee f_s$ の演算を行っている. MQP Type I 及び Type IV は $\mathbb{Z}_2$ 係数の最高次 2 次の多項式系であり, Type I はその多項式系がほぼ唯一解を持つように定義される.

以上の計算例に対して, 本手法 RecBGB, PolyBoRi[3] Groebner メソッド (オプション無し), 及び Risa/Asir[10] `nd_f4` との間で辞書式順序に関する簡約ブーリアングレブナー基底の計算時間を比較した. MQP の係数はランダムに決定されるため, 10 通りの計算を行い平均をとった. PolyBoRi は ZBDD を用いたブール多項式処理系であり, Groebner メソッドはブッフバーガーアルゴリズムをもとに, $\mathbb{Z}_2[x_1, \ldots, x_n]$ 上での基準を用いて, 特に辞書式順序に関して計算が効率化されている. また, Risa/Asir による計算では, $\{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$ を基底に加えて $\mathbb{Z}_2$ 上のグレブナー基底計算を行っている. 計算環境は Linux Mint 19.2 Tina 64-bit, Intel Core i3-8100 3.60GHz, RAM 16GB, gcc-7.4.0 である. また, PolyBoRi は SageMath 8.1 を, Risa/Asir は 20191120 (Kobe Distribution) を用いた.

実験結果を表 1, 図 1 に示す. いずれの例も, 提案手法が正常に終了した範囲の結果を示している.

全体的な傾向として, いずれの手法についても, 計算時間は変数の個数に対する指数的な増加を示した. 提案手法と PolyBoRi は ($\mathbb{Z}_2$ 係数の) ブール多項式に特化しているため, どの例に関しても, 汎用的な Risa/Asir と比較して効率的であった.

以降, 提案手法 RecBGB と PolyBoRi を比較する. life, MQP Type IV に対しては提案手法が効率的に計算できていた. これは特に life に顕著で, 図 1(a) から計算時間の増加傾向が緩やかであることがわかる. ある多項式 f から $\langle f \rangle$ のブーリアングレブナー基底を求めるにあたっては, 本手法は有効であるといえる. 一方で, MQP Type I に対して, 提案手法は高速であるものの, 図 1(c) から計算時間の増加傾向をみると, 実験環境を変えて変数の個数をさらに増やした場合には PolyBoRi の方が効率的となり得る. また, cyclic に対しては, PolyBoRi が高速であった.

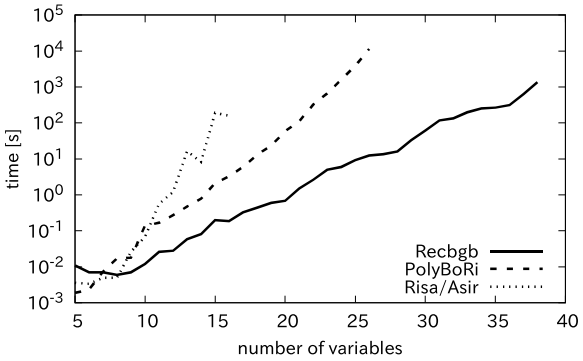
具体的な数値は省略するが, 提案手法の計算時間内訳をみたところ, cyclic, MQP Type I, MQP Type IV のいずれについても, 前処理 $f_1 \vee \cdots \vee f_s$ の計算時間がほとんどを占めていた. 特に, cyclic, MQP Type I に関しては, アルゴリズム 1 にあたる計算時間は 1 ミリ秒未満であり, 計測ができなかった. ここで, 各系の解に注目する. cyclic は変数の数 n が奇数のとき解を持たず, 偶数のとき $(x_1, \ldots, x_n) = (1, \ldots, 1)$ を唯一解として持つ. また, MQP Type I はほとんどの場合唯一解を持つ. このように系が解を持たない場合や唯一解を持つ場合, 単元な基底 $\{g\}$ がわかっていれば, そのブーリアングレブナー基底 (あるいは解) は直ちに求まる. このため, 特に唯一解を調べるにあたっては, 本手法をそのまま適用するには課題があるといえる.

5 結論

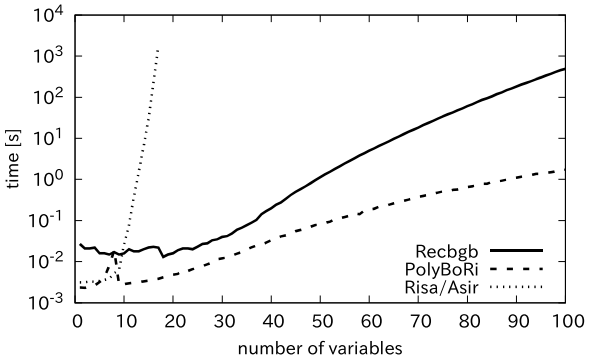
本稿では, 辞書式順序に関する $\mathbb{Z}_2(x_1, \ldots, x_n)$ 上のブーリアングレブナー基底構成について新たな手法を提案した. また, 提案手法の実装を行い, 幾つかの計算例から, 本手法とその実装が有効となる場合があることを確認した. 一方で, 有効性は限定的であり, 2 章におけるブッフバーガーアルゴリズムの拡張も合わせて, 一般のブーリアングレブナー基底計算における効率化, 本手法の活用可能性などが課題である.

表 1: 計測時間 (単位秒, 一部抜粋)

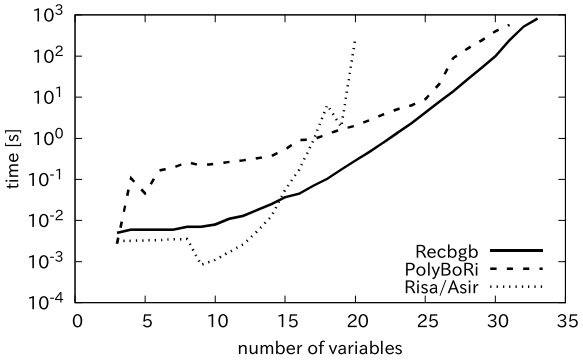
	n	RecBGB	PolyBoRi	Risa/Asir
life	14	0.081	0.785	8.218
	15	0.199	2.114	190.4
	16	0.185	3.289	158
	17	0.326	6.007	
	⋮	⋮	⋮	
	24	5.992	1611.303	
	25	9.233	3894.273	
	26	12.390	11388.881	
	27	13.473		
	⋮	⋮		
	36	313.822		
	37	638.588		
	38	1354.250		
cyclic	15	0.022	0.003	29.7
	16	0.023	0.004	223.077
	17	0.022	0.004	1786.791
	18	0.013	0.004	
	⋮	⋮	⋮	
	98	413.890	1.579	
	99	457.768	1.672	
	100	488.329	1.726	
MQP-I	18	0.104	1.258	6.429
	19	0.175	1.663	1.899
	20	0.289	2.024	271.1
	21	0.462	2.753	
	⋮	⋮	⋮	
	29	51.519	239.525	
	30	99.713	400.420	
	31	242.776	573.427	
	32	517.748		
	33	822.156		
MQP-IV	14	0.013	1.697	0.9764
	15	0.018	2.595	5.735
	17	0.040	8.268	475.7
	18	0.057	35.710	
	⋮	⋮	⋮	
	20	0.165	108.933	
	21	0.255	192.241	
	23	0.931	2246.373	
	24	1.705		
	⋮	⋮		
	30	93.921		
	32	423.295		
	33	1033.685		



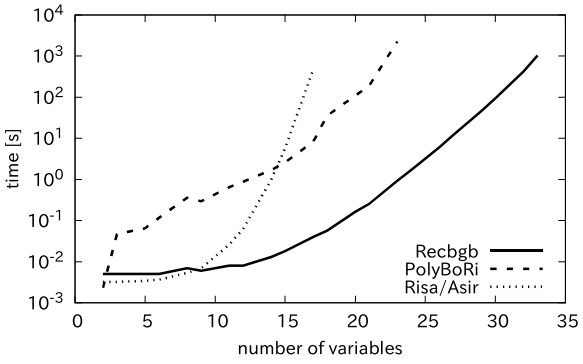
(a) life



(b) cyclic



(c) MQP Type I



(d) MQP Type IV

図 1: 計測時間の比較

参 考 文 献

[1] S. B. Akers, Binary Decision Diagrams, IEEE Transactions on Computers, **C-27** (1978), pp.509–516.

[2] M. Brickenstein, Boolean Gröbner bases — Theory, Algorithms and Applications, Logos Verlag Berlin GmbH, 2010.

[3] M. Brickenstein, A. Dreyer, A framework for Gröbner-basis computations with Boolean polynomials, J. Symb. Comput., **44** (2009), pp.1326–1345.

[4] D. A. Cox, et al., Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 4th edn., Springer, 2015.

[5] V. P. Gerdt, M. V. Zinink, A Pommaret division algorithm for computing Gröbner bases in boolean rings, Proceedings of the 21st International Symposium on Symbolic and Algebraic Computation, 2008, pp.95–102.

[6] S. Minato, Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, 30th ACM/IEEE Design Automation Conference, 1993, pp.272–277.

[7] Y. Sato, et al., Boolean Gröbner bases, J. Symb. Comput., **46** (2011), pp.622–632.

[8] T. Yasuda, et al., `https://www.mqchallenge.org`.

[9] `https://github.com/takemaru/graphillion/tree/`

[10] `http://www.math.kobe-u.ac.jp/Asir/asir-ja.html`