

ブレイド群における共役判定プログラムの試作

A program for conjugacy problem of braids

谷 研究室 三角 哲也

Tetsuya Misumi

概要

2000 年に、ブレイド (組み紐) 群における共役問題の難しさを利用した公開鍵暗号系が提案された。本研究ではブレイドで公開鍵暗号の安全性を検証する上で必要となるプログラムを実装する。任意のブレイド V において、 V と共役なもの集合は無限集合であるため、その有限集合 summit set を求めるアルゴリズムを実装した。

1 はじめに

ネットワーク化が進んだ今日、情報の隠蔽を目的とした暗号技術は、数々のセキュリティ技術の中核を成すものの 1 つとして位置付けられている。暗号とは、当事者以外には意味がわからないように、鍵を使い決められた手順に従って変換した特殊な記号や文字、または方式のことをいう。

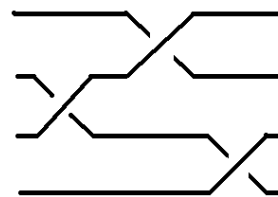
現在最も広く使われている公開鍵暗号系は、素因数分解の難しさを利用した RSA 暗号系である。この暗号系では、2 つの大きな素数の積を鍵としている。安全性を高めるためにより長い鍵を使うことが求められているが、携帯電話や IC カードのように限られた環境で大きな整数の演算を行うことは困難であり、RSA 暗号系はそのような環境に適していない。そこで、楕円曲線暗号などの新しい公開鍵暗号系が研究されている。

本研究ではブレイド群を使った公開鍵暗号を考えるに当たり、安全性の根拠となる共役問題の難しさを検証する手段の一つとして、計算可能な有限集合 summit set を求めるアルゴリズムを実装し、2005 年に松葉氏によって行われた実験と比較検討した。2 章ではブレイド群の基本的な定義の解説、3、4、5 章ではその応用とブレイド群における困難な問題の紹介、6 章ではアルゴリズムを紹介し、7、8 章では実験と今後の課題などを示した。

2 ブレイド群の定義

2.1 ブレイドとは

n -ブレイドとは平行に並ぶ絡み合った n 本の紐であり、右から出ている紐をたどると常に左へ向かう。 n -ブレイド群のことを B_n と表す。



4-ブレイドの例

2.2 単位元 e

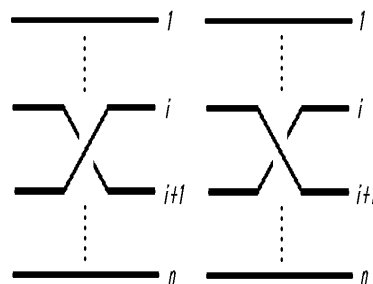
単位元 e とは、ブレイドのうち、紐が一度も交差しないもの。

2.3 生成元 t

単位元 e において上から数えて i 番目と $i+1$ 番目の紐のみが交差しているブレイドで、 i 番目の紐が手前にきているものを t_i と表現する。これを生成元と呼ぶ。

2.4 逆元

t_i^{-1} は、 t_i の紐の上下関係を逆にしたブレイドであり、 t_i の逆元となっている。



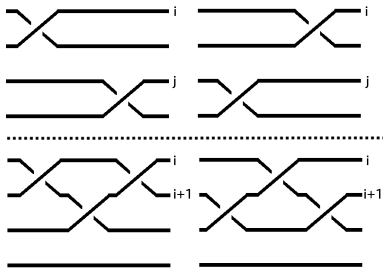
(a) t_i

(b) t_i^{-1}

2.5 ブレイドの同値

2 つのブレイド V, W が同値とは、一方のブレイドの紐の両端を固定し、上下関係を維持したまま動かしもう一方のブレイドと同じものにできる時、2 つのブレイドは同値であるという。

2.6 生成元における同値関係



$$1. t_i t_j = t_j t_i \quad (|i - j| > 1)$$

$$2. t_i t_{i+1} t_i = t_{i+1} t_i t_{i+1}$$

2.7 積

2つのブレイド V, W の積 VW とは, V の右端と W の左端を繋げたブレイドのことをいう。

2.8 正ブレイド

t_i^{-1} を含まないブレイド。正ブレイドの集合を B_n^+ と表す。

2.9 非冗長ブレイド

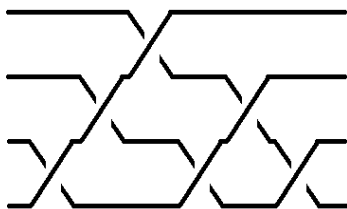
正ブレイドの中でどの二本の紐も高々一回だけ交差するもの。非冗長ブレイドの集合を D_n と表す。

2.10 基本ブレイド

非冗長ブレイドの中でどの二本の紐も一回だけ交差するもの。このブレイドは

$$\Omega_n = (t_1 t_2 \dots t_{n-1})(t_1 t_2 \dots t_{n-2}) \dots (t_1 t_2) t_1$$

と表せる。



基本ブレイドの例

3 canonical form

3.1 分割

$X \in D_n, A \in B_n^+$ に対して, $P = XA$ となるとき, XA を P の分割という。

3.2 Left-weighted

P の分割 XA が Left-weighted であるとは, 任意の正ブレイド P の分割 $X'A'$ に対して, $X' \leq X$ となることをいい, $X \triangleright A$ と表す。

3.3 標準形表現

標準形表現とは, $V \in B_n$ に対して, $p \in \mathbb{Z}$ と正ブレイド $A_1, A_2, \dots, A_q \in D_n$ を用いた

$$V = \Omega^p A_1 A_2 \dots A_q$$

という表現をするもので, $1 \leq i \leq q-1$ となる各 i に対し、

$$A_i \triangleright A_{i+1}$$

が成り立つ。

3.4 head, tail

$a, b, c \in B_n^+$ において $ab = c$ のとき, a は c の head といい, b は c の tail という。

3.5 maximal head

P の head H が maximal head であるとは, H と H を取り除いた P の tail が Left-weighted となることをいう。

3.6 標準形の作り方

任意のブレイド V から標準形を作るには, V を maximal head と tail に分け, 更にその tail を maximal head と tail に分け, それを繰り返した後, 出てきた maximal head を順に表し, 最後に残った tail を表す。

4 ブレイド群における計算問題

ブレイド X と Y が共役であるとは, $Y = A^{-1}XA$ となるブレイド A が存在するときを言う。共役に関して、困難であると思われる計算問題に以下のようなものがある。 B_m を $m < n$ に対して $t_1, \dots, t_{m-1}$ で表される B_n の部分群とする。

4.1 共役決定問題

入力: $(X, Y) \in B_n \times B_n$

出力: X と Y が共役であるか?

4.2 共役探索問題

入力: 互いに共役であるような $(X, Y) \in B_n \times B_n$

出力: $Y = A^{-1}XA$ となるような $A \in B_m$

4.3 一般共役探索問題

入力 : $B \in B_m, m \leq n$ に対して $Y = B^{-1}XB$ となるような $(X, Y) \in B_n \times B_n$

出力 : $Y = A^{-1}XA$ となるような $A \in B_m$

4.4 共役分解問題

入力 : $B \in B_m, m \leq n$ に対して $Y = B^{-1}XB$ となるような $(X, Y) \in B_n \times B_n$

出力 : $Y = A'XA''$ となるような $A', A'' \in B_m$

5 Summit set

5.1 τ

任意のブレイド V に τ を施すとは, V に対して, $\Omega^{-1}V\Omega$ という積を作ること.

5.2 τ^{-1}

任意のブレイド V に τ^{-1} を施すとは, V に対して, $\Omega V\Omega^{-1}$ という積を作ること.

5.3 cycling, decycling

cycling とは, ブレイド V の標準形表現が

$$V = \Omega^p W_1 W_2 \cdots W_q$$

のとき,

$$V = \Omega^p W_2 \cdots W_q \tau^{-p}(W_1)$$

とするもので, これを $cyc(V)$ と表す.

5.4 decycling

decycling とは, ブレイド V の標準形表現が

$$V = \Omega^p W_1 W_2 \cdots W_q$$

のとき,

$$V = \Omega^p \tau^p(W_q) W_1 \cdots W_{q-1}$$

とするもので, これを $decyc(V)$ と表す.

5.5 inf

標準形表現 $W = \Omega^p W_1 W_2 \cdots W_q$ において

$$\inf(W) = p$$

5.6 sup

標準形表現 $W = \Omega^p W_1 W_2 \cdots W_q$ において

$$\sup(W) = p + q$$

5.7 summit set

任意のブレイド V において, V と共役なものの集合のうち,

$$C^{sum}(V) := \left\{ S \mid C(V) : \begin{array}{l} \inf(S) = \max\{\inf(S') : S' \in C(V)\} \\ \sup(S) = \min\{\sup(S') : S' \in C(V)\} \end{array} \right\}$$

を満たすものを summit set という.

6 アルゴリズム

6.1 summit set の元の 1 つを求めるアルゴリズム

```
(INPUT) W      B_n
(OUTPUT) W の summit set の元の一つ
i ← 1
while i ≤ |Ω|-1 do
  W ← cyc(W).
  i ← i+1.
od
i ← 1
while i ≤ |Ω|-1 do
  W ← decyc(W).
  i ← i+1.
od
return W.
```

6.2 任意の 2 つの summit set の元が共役であるかを判定するアルゴリズム

```
(INPUT) V', C^sum(V), W', C^sum(W)
(OUTPUT) ACCEPT もしくは REJECT
if inf(V') ≠ inf(W') or sup(V') ≠ sup(W')
then return REJECT.
repeat
  guess C      D_n.
  if C^{-1} V' C = C^sum(V).
  then V' ← 標準形の C^{-1} V' C.
  else return REJECT.
until V' ≡ W'
return ACCEPT.
```

7 実験

7.1 実験内容

松葉氏により紐の本数を 5 本、語長が 1 ~ 60 の時のブレイドをランダムに生成し、それに対する最大の summit set の大きさを調べるという実験が行われている。私は語長が 1 ~ 10 の時に限定しすべての組み合わせを列挙し、それに対する summit set の大きさを調べた。

7.2 実験結果

		SUMMIT SET の元の canonical length									
Max summit size		1	2	3	4	5	6	7	8	9	10 最大
語長	1~5	12	12	18	4						18
	6~10	10	60	104	156	96	120				156
	11~15	12	60	460	728	644	384	378	288	54	728
	16~20	8	120	344	1320	1160	1488	2056	2168	608	364
	21~25	12	120	460	816	1960	2236	1580	1218	1356	1558
	26~30	10	60	260	832	2760	3100	6494	4560	4926	2928
	31~35	10	32	132	765	1848	3286	3682	4442	2034	3072
	36~40			30	438	656	2812	3842	2956	3308	2324
	41~45	8			146	214	2104	990	3644	5356	2208
	46~50					34	44	1272	1832	976	3614
	51~55						250	468	244	1522	678
	56~60								1340	470	450
最大		12	120	460	1320	2760	3286	6494	4560	5356	3614

松葉氏データ

		SUMMIT SET の元の canonical length									
Max summit size		1	2	3	4	5	6	7	8	9	10 最大
語長	1	2	0	0	0	0	0	0	0	0	2
	2	6	2	0	0	0	0	0	0	0	6
	3	10	3	2	0	0	0	0	0	0	10
	4	10	11	3	2	0	0	0	0	0	11
	5	12	16	10	3	2	0	0	0	0	16
	6	8	17	15	10	3	2	0	0	0	17
	7	8	25	16	15	10	3	2	0	0	25
	8	6	34	33	16	15	10	3	2	0	34
	9	4	39	40	33	16	15	10	3	2	40
	10	0	30	49	40	33	16	15	10	3	49
最大		12	39	49	40	33	16	15	10	3	49

三角データ

7.3 考察

私のデータは全列挙しているので、語長が 1 ~ 10 のときの松葉氏のデータの値と同じかより大きくなるはずだが、値が小さくなってしまふ部分があるので、松葉氏のプログラムと比較検討する必要がある。

8 今後の課題

- 語長を増やした場合で全列挙しようとするデータが膨大になるので、summit set の大きさが大きくなるようなデータの取り方を考える

参考文献

- [1] D.B.A.Epstein, James W. Cannon, Word Processing in Group, 181-209, 1992
- [2] 日本大学大学院総合基礎科学研究科地球情報数理科学専攻 平成 15 年度 修士論文, 松葉正志, ブレイド群における共役問題の計算量について, 2005
- [3] Masashi Matsuba and Seiichi Tani: "On Computational Complexity of the Conjugacy Problem for Braids", 信学技報 COMP-2003-88 (2004-03), pp.17-23.