Factoring Products of Braids via Garside Normal Form

Simon-Philipp Merz *and Christophe Petit*

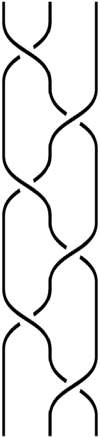
Royal Holloway, University of London

University of Birmingham

April 16, 2019

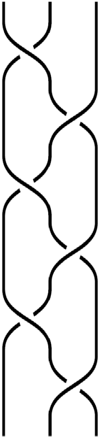
Presentation Contents

Braid Groups



- B_N denotes braid group on $N \in \mathbb{Z}_{>0}$ strands
- elements are equivalence classes of collections of N strands under ambient isotopy

Braid Groups



- B_N denotes braid group on $N \in \mathbb{Z}_{>0}$ strands
- elements are equivalence classes of collections of N strands under ambient isotopy

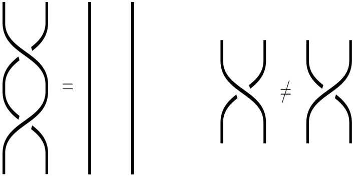


Figure: Equivalence of Braids

Braid Groups

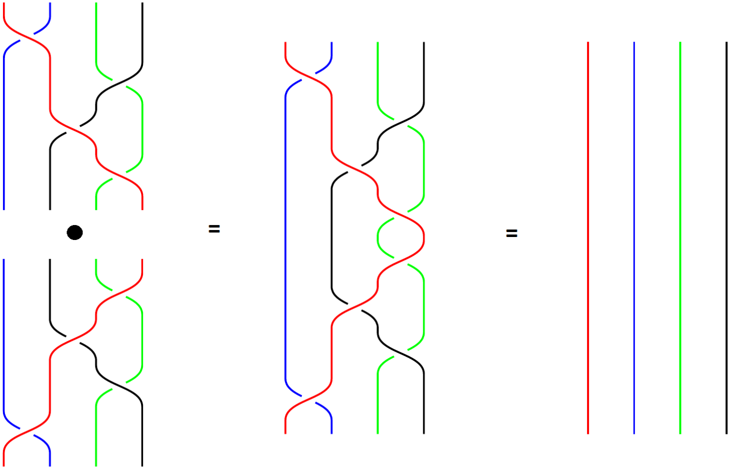


Figure: Concatenating and inverting braids

Algebraic presentation of Braid Groups

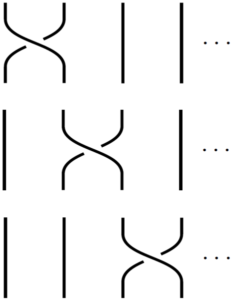


Figure: Artin generators b_1, b_2, b_3

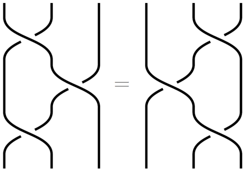


Figure: $b_i b_{i+1} b_i = b_{i+1} b_i b_{i+1}$

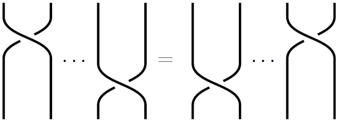


Figure: $b_i b_j = b_j b_i$, if $|i - j| \geq 2$

Algebraic presentation of Braid Groups

Theorem (Artin and Bohnenblust, 1946)

B_N is a group with the algebraic presentation

$$B_N = \left\langle b_1, \ldots, b_{N-1} \,\middle|\, \begin{array}{l} b_i b_{i+1} b_i = b_{i+1} b_i b_{i+1} \\ b_i b_j = b_j b_i \textit{ for } |i-j| \geq 2 \end{array} \right\rangle.$$

The group operation is concatenation of the strands.

Permutation Braids

- there is a natural homomorphism $\pi : B_N \to S_N$ sending braids to the permutation they induce
- braids that can be written as product of positive powers of Artin generators are called *positive braids*
- positive braids for which each each pair of strands cross at most once are called *permutation braids*

$$\{\text{permutation braids}\} \leftrightarrow S_N$$

- *fundamental braid* $\Delta \in B_N$ is the permutation braid for which each pair of strands crosses exactly once

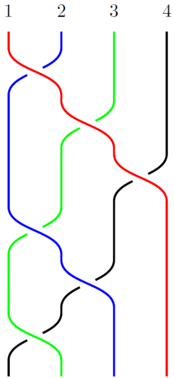


Figure: $\Delta \in B_4$ inducing the permutation $(1\,4)(2\,3)$

Permutation Braids

- there is a natural homomorphism $\pi : B_N \to S_N$ sending braids to the permutation they induce
- braids that can be written as product of positive powers of Artin generators are called *positive braids*
- positive braids for which each each pair of strands cross at most once are called *permutation braids*

$$\{\text{permutation braids}\} \leftrightarrow S_N$$

- *fundamental braid* $\Delta \in B_N$ is the permutation braid for which each pair of strands crosses exactly once

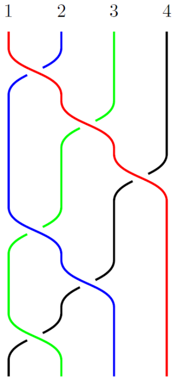


Figure: $\Delta \in B_4$ inducing the permutation $(1\,4)(2\,3)$

Garside Normal Form of B_N

Theorem (Garside left normal form)

Every braid β can be represented uniquely by a braid word

$$\Delta^r A_1 \cdots A_k,$$

where $r \in \mathbb{Z}$, A_i are permutation braids and $A_i A_{i+1}$ is a left-weighted product for $1 \leq i \leq k$.

Multiple variants to compute Garside normal forms [Th92, EM94].

Normal Forms of Products of Braids

Let A, $B \in B_N$ with left normal forms $\Delta^a{\cdot}A_0 \dots A_n$ and $\Delta^b{\cdot}B_0 \dots B_m$ respectively and let $\tau : B_N \to B_N,\ m \to \Delta m \Delta^{-1}$.

Theorem

The left normal form of AB is

$$\Delta^{a+b+k} \cdot \tau^i(A_0) \dots \tau^i(A_{n-c_1}) \cdot X_1 \dots X_l \cdot B_{c_2} \dots B_m,$$

for some integers $k \in \mathbb{Z}$, $0 \le c_1 \le n$, $0 \le c_2 \le m$, $i \in \{0,1\}$ *and permutation braids* $X_1, \dots, X_l$.

Normal Forms of Products of Braids

Let A, $B \in B_N$ with left normal forms $\Delta^a \cdot A_0 \dots A_n$ and $\Delta^b \cdot B_0 \dots B_m$ respectively and let $\tau : B_N \to B_N$, $m \to \Delta m \Delta^{-1}$.

Theorem

The left normal form of AB is

$$\Delta^{a+b+k} \cdot \tau^i(A_0) \dots \tau^i(A_{n-c_1}) \cdot X_1 \dots X_l \cdot B_{c_2} \dots B_m,$$

for some integers $k \in \mathbb{Z}$, $0 \leq c_1 \leq n$, $0 \leq c_2 \leq m$, $i \in \{0, 1\}$ and permutation braids $X_1, \dots, X_l$.

Penetration Distance

Definition

For two braids A and B, the *penetration distance* $\mathrm{pd}(A, B)$ for the product AB is the number of permutation braids at the end of the normal form of A which undergo a non-trivial change in the normal form of the product.

Conjecture

Let $A, B \in B_N$ be braid words that are picked uniformly at random from all freely reduced braid words of length k. Then there exists a $C_N \in \mathbb{N}$ such that for all k, we have

$$\mathbb{E}(\mathrm{pd}(A, B)) < C_N.$$

Factoring Products of Braids

Let A, $W = \Delta^w \cdot W_1 \cdots W_m = \mathcal{W}_1\mathcal{W}_2\mathcal{W}_3$ and C be randomly chosen braids.
If $m > 2C_N$, we expect the left normal form of AWC to be of the form

$$\Delta^j \cdot \mathcal{X} \cdot \tau^i(\mathcal{W}_2) \cdot \mathcal{Y}$$

for some product of permutation braids $\mathcal{X}, \mathcal{Y}$, $j \in \mathbb{Z}$ and $i \in \{0, 1\}$, such that

$$\begin{aligned} A' &= \Delta^j \cdot \mathcal{X} \cdot \tau^i(\mathcal{W}_1)^{-1} \\ &\equiv A \pmod{\Delta^2} \\ \text{and} \qquad C' &= \tau^i(\mathcal{W}_3)^{-1} \cdot \mathcal{Y} \\ &\equiv C \pmod{\Delta^2} \end{aligned}$$

Factoring Products of Braids

Let A, $W = \Delta^w \cdot W_1 \cdots W_m = \mathcal{W}_1\mathcal{W}_2\mathcal{W}_3$ and C be randomly chosen braids.
If $m > 2C_N$, we expect the left normal form of AWC to be of the form

$$\Delta^j \cdot \mathcal{X} \cdot \tau^i(\mathcal{W}_2) \cdot \mathcal{Y}$$

for some product of permutation braids $\mathcal{X}, \mathcal{Y}$, $j \in \mathbb{Z}$ and $i \in \{0, 1\}$, such that

$$\begin{aligned} A' &= \Delta^j \cdot \mathcal{X} \cdot \tau^i(\mathcal{W}_1)^{-1} \\ &\equiv A \pmod{\Delta^2} \\ \text{and} \qquad C' &= \tau^i(\mathcal{W}_3)^{-1} \cdot \mathcal{Y} \\ &\equiv C \pmod{\Delta^2} \end{aligned}$$

Factoring *AWC*

Knowing W, can recover A and C up to the centre of B_N, $\langle \Delta^2 \rangle$:

- compare permutation braids in GNF of $\tau^i(W)$ with the one of *AWC* to find i and common contiguous subsequence $\tau^i(\mathcal{W}_2)$.
- compute

$$A' \equiv A \pmod{\Delta^2}$$
$$C' \equiv C \pmod{\Delta^2}.$$

Figure: Untangling the braid

Conjugacy Search Problem

Definition

Given $X, Y \in B_N$, where $Y = C \cdot X \cdot C^{-1}$ for some $C \in B_N$, the conjugacy search problem (CSP) in braid groups is to find $\tilde{C} \in B_N$ such that $Y = \tilde{C} \cdot X \cdot \tilde{C}^{-1}$.

Recovering $\tilde{C} \equiv C \pmod{\Delta^2}$ will do!

Cryptanalysis of WalnutDSA

- WalnutDSA is a signature scheme submitted to the NIST PQC project
- signatures are braids
- has been attacked (and fixed?) multiple times
- ! is pushed for use in the real world !

Figure: Cracking a Walnut

Cryptanalysis of WalnutDSA

- signatures are braids with a representative

$$S_1 \cdot E(\mathrm{msg}) \cdot S_2,$$

  where S_1, S_2 are secret braids with some randomness added and $E(\mathrm{msg})$ is a deterministic encoding of msg

- before appending signature to message a rewriting algorithm is applied to “obfuscate” single factors

Cryptanalysis of WalnutDSA

Universal forgery attack:

- use our factoring algorithm to recover S_1 and $S_2 \pmod{\Delta}^2$ from any message-signature pair
- splice encoding $E(\text{msg'})$ in between to obtain a valid signature for any msg'
- works on 99.8% of random message-signature pairs for 128-bit parameters (< 1s) and 100% of random message-signature pairs for 256-bit (≈ 3s) parameters
- widely independent of WalnutDSA parameters

Countermeasure:

- randomize encoding of messages sufficiently to prevent adversaries from finding matching permutation braids in signature

Cryptanalysis of WalnutDSA

Universal forgery attack:

- use our factoring algorithm to recover S_1 and S_2 (mod $\Delta)^2$ from any message-signature pair
- splice encoding E(msg') in between to obtain a valid signature for any msg'
- works on 99.8% of random message-signature pairs for 128-bit parameters (< 1s) and 100% of random message-signature pairs for 256-bit (≈ 3s) parameters
- widely independent of WalnutDSA parameters

Countermeasure:

- randomize encoding of messages sufficiently to prevent adversaries from finding matching permutation braids in signature

Cryptanalysis of WalnutDSA

Universal forgery attack:

- use our factoring algorithm to recover S_1 and S_2 (mod $\Delta)^2$ from any message-signature pair
- splice encoding E(msg') in between to obtain a valid signature for any msg'
- works on 99.8% of random message-signature pairs for 128-bit parameters ($<$ 1s) and 100% of random message-signature pairs for 256-bit ($\approx$ 3s) parameters
- widely independent of WalnutDSA parameters

Countermeasure:

- randomize encoding of messages sufficiently to prevent adversaries from finding matching permutation braids in signature

Summary and Concluding Remarks

- secret braids are not necessarily "hidden" in the product of multiple braids
- provided algorithm to recover single factors up to the centre of B_N for products AWC of randomly chosen braids, if W is known and sufficiently long
- can solve CSP instances in braid groups and universally forge signatures of WalnutDSA
- plenty of structure in braid groups is not fully explored yet

Summary and Concluding Remarks

- secret braids are not necessarily "hidden" in the product of multiple braids
- provided algorithm to recover single factors up to the centre of B_N for products AWC of randomly chosen braids, if W is known and sufficiently long
- can solve CSP instances in braid groups and universally forge signatures of WalnutDSA
- plenty of structure in braid groups is not fully explored yet

Summary and Concluding Remarks

- secret braids are not necessarily "hidden" in the product of multiple braids
- provided algorithm to recover single factors up to the centre of B_N for products AWC of randomly chosen braids, if W is known and sufficiently long
- can solve CSP instances in braid groups and universally forge signatures of WalnutDSA
- plenty of structure in braid groups is not fully explored yet