Research Paper Mathematics

Braid Group: A Completely New Algorithm

Rahul Singh	Assistant Professor, Ramanujan College, Delhi University. New Delhi-110019.
Maneesh Kumar	Assistant Professor, Motilal Nehru College,Delhi University,New Delhi-110021.
Alok Kumar	Assistant Professor, Deshbandhu College, Delhi University.New Delhi-110019.
Dheeraj Tiger	Assistant Professor,Rajdhani College,Delhi University,New Delhi-110015

ABSTRACT

Notion of braids and braid group was first considered when orbit of the first observed asteroid was studying. In the present paper a quick algorithm in comparison with other methods for short braid words is obtained. In this algorithm for short braid words over a large number of strings a very short description of the -base, that yields a very fast algorithm is obtained .This means that a very useful and practical algorithm is developed in the present paper. The present algorithm is also highly implemented on a computer.

KEYWORDS Braid group, Garsides Algorithm, g-base, Dehornoys Algorithm

1. Introduction

For $n \geq 2$ the braid group B_n is defined by the presentation$< \sigma_1, \sigma_2, \dots \dots \dots \dots, \sigma_{n-1} : \sigma_i\sigma_j = \sigma_j\sigma_i$ for $|i-j| \geq 2, \sigma_i\sigma_{i+1}\sigma_{i=}\sigma_{i+1}\sigma_i\sigma_{i+1}$for $|i-j| = 1 >$. This presentation is called the Artin presentation and the generators are called Artins generators.

An element of B_n will be called an - n braid. For each n, the identity mapping on $\{\sigma_1, \sigma_2, \dots \dots \dots \dots, \sigma_{n-1}\}$ induces an embedding of B_n into B_{n+1} , so that we can consider an - n braid as a particular $n+1$ braid. Using this, one can define the limit group B_∞ .

Note that B_2 is an infinite cyclic group,and hence it is isomorphic to the group Z of integers. For $n \geq 3$, the group B_n is not commutative and its center is an infinite cyclic subgroup. Every n-braid is an equivalence class of n- braid words under the congruence generated by the relations in Presentation.

2.1 Braid based scheme

2.1.1 Key agreement Scheme

Two parties, Alice and Bob,want tosh area secret over public channel.The shared secret can be used as a encryption-decryption key for a secure commu-nication.

Example: Diffie-Hellman Scheme

Alice chooses x and send $f(x)$ to Bob.Similarly Bob choose y and send $f(y)$ to Alice.

Alice and Bob compute the common secret

$$F\big(x, f(y)\big) = F(y, f(x))$$

Here we need an one-way function f and a corresponding trap-door function F.For example,

$$f(x) = a^x,\ F(x,b) = b^x \quad \text{for } a, b, x, \epsilon\, z_p^*$$

2.1.2 Public Key Encryption Scheme

Bob wants to send a message to Alice confidentially over public channel.

Example: El Gamal Scheme

Alice chooses a decryption key x privately and makes the corresponding encryption key $f(x)$ public. Bob chooses a random y and encrypts the message m to produce a ciphertext $\big(f(y), F\big(y, f(x)\big)\big) \oplus m)$ and sends the ciphertext to Alice. Alice recovers the message m by computing $F\big(x, f(y)\big) \oplus F\big(y, f(x)\big) \oplus m = m$. Again we need a one-way function f and a corresponding trap door function F.

3. Previous algorithms for word problem

The complexity of different algorithms varies, but to our knowledge, the best known solution is of complexity of $O(l^2)$ where l is the length of the longer braid. There are two most important algorithm for solving word problem.

3.1 Garsides Algorithm:

Garside gave a solution for the braid word problem in 1969. His solution is based on the definition of positive words, which contain only

generators with positive power. Then, he stated that the fundamental word of the braid group Δ^n has a property that enables to replace all the generators with a negative power. This can be done simply by noticing the fact that for any i, there exists a positive braid word w_i for which $\sigma_i^{-1} = \Delta_n^{-1}.w_i$ Another property of the Δ_n is that for any i we have that $\sigma_i\Delta_n = \Delta_n\sigma_{i-1}$. This gives a method for writing a given braid word w in such away that $w = w_1w_2$ where $w_1 = \Delta_n^r$, $r \leq 0$, which is a negative braid word and w_2 is a positive braid word.

Now, one can write $w_2 = \Delta_n^q w_3$, where q is maximal. By doing this,he can increase r resulting in the minimal way of writing $w = \Delta_n^{r-q} w_3$ By organizing By organizing w_3 in a lexicographicin order,we obtain what is called Garsides normal form of the braid word w. Garside proved, that two braid words w and w' are equal if and only if their normal forms are the same. There are some implementations for solving the braid word using this solution, and variations of it as can be found .For achieving the best complexity by this method, one has to expand the size of the set of generators of the braid group, resulting in the complexity of $O(l^2)$ where l is the length of the longer of the two braid words.

3.2 Dehornoys Algorithm:

Dehornoy used a different approach for solving the problem. His approach is based on a definition of a σ reduced braid word, which is a braid word that for any integer i, any occurrence of the letter σ_i is separated from any occurrence of the letter σ_i^{-1} by at least one occurrence of a letter σ_j^{+1} with $j < i$. Dehornoy presented an algorithm for transforming any braid word to its reduced form He proved that the reduced form of a braid word w is Id (i.e.the null braid word) if and only if w is the identity word. This gives a simple way of checking whether two braid w and w' are equal, simply by writing $w'' = w(w')^{-1}$ and reducing w''. If the reduced form of w'' is Id, it means that $= w'$. The reduction process is actually a type of an unknotting process that unties the twisted strings in a braid, by adding proper sequences and transforming locally twisted strings into an untwisted state.

Dehornoy conjectured that the complexity of his algorithm is bounded by $O(l^2)$ where l is the length of the longer braid word.

In the next section we will present our algorithm, which is based on a completely different approach.

4. The mixed algorithm: A compltely new approach

The algorithm that we are going to develop in the present paper in order to solve the word problem in the braid group is based on the interplay between its two definitions. We will fix the standard frame and the standard g-base that will be used as a starting position. We associate the generator σ_i to the half-twist H_i in the standard frame for every $i = 1,2, \ldots\ldots\ldots\ldots\ldots n-1$ By using our two algorithms and encoding the g -bases in a unique way, and by using an algorithmic way to explore the changes that happen to the standard g -base while the braid word acts on it, we produce a practical algorithm for the word problem.

Mathematically, we compare two braid words by taking one braid word and compute the result of its action on the standard g-base of the fundamental group. Then, we take the other braid word and compute the same result. The two braid words are equal if and only if the two resulted g-bases are identical. The computerized implementation of the g -base: we will describe the way we encode the g-base.It involves some conventions.

Recall that D is the closed unit disk, the point u is the point $(0,-1)$ and the points in K are on the x-axis. In order to encode the path in D, which is an element of the g-base, we will distinguish some positions in D. We will denote by $(i,1)$ a point close to k_i but above it, $(i,-1)$ a point close to k_i but below it, and $(i,0)$ the point k_i itself. We will also denote the point u by $(-1,0)$

(which is not its position in D, rather only a notation).

To represent a path in D, we will use a linked list which its links are based on the notations above, which represents the position of the path in relation to the points $u\ and\ k_i,\ i = 1,2 \ldots\ldots\ldots\ldots\ldots\ldots n$. Each link of the list holds the two numbers as described above. Now we are ready to present the algorithm:

4.1 Algorithm 1

Process Word (w) input: w -a braid word.

Output: a list which represents the g -base resulted after the action of the words letters on the standard g -base.

Process Word (w):

$g \leftarrow$ List that represents the standard g -base.

For every letter σ_i in w do

1. Act on g using σ_i by applying (positive/Negative) Half Twist (σ_i , g) function.
2. Reduce g to its unique form using Reduce (g) function.

Return g .

Now, we will present the Positive Half Twist (σ_i , g) function.

4.2 Algorithm 2

Positive Half Twist (σ_i , g)

Input:

σ_i -the generator of the braid group acting on the g -base. g -the list representing the g -base.

Output: A list representing the g -base after the action of $\sigma_i$on .

Positive Half Twist (σ_i , g): for each sequence of links in g of the type (i, e) or $(i + 1, e)$ or $e \in \{-1,0,1\}$ do

Before Section ← The link just before the first link in the sequence

After Section ← The link after the last link in the sequence

First Link ← The first link in the sequence

Second Link ← the second link in the sequence

If Before Section $= (-1,0)$ then

act upon one of the following cases:

If First Link.Point = i and SecondLink.Point= 0 then add the link $(i, -1, -1)$ after Before Section

Before Section ← the new link

If First Link.Point $= i + 1$ and FirstLink.Position $= 0$ then add the link $(i + 2, -1)$ after Before Section

Before Section ← the new link

If First Link.Point = i and SecondLink.Point = $i + 1$ then add the link $(i, -1, -1)$ after Before Section

Before Section ← the new link

If First Link.Point = i and SecondLink.Point = $i - 1$ then add the links $(i, -1, -1) \rightarrow (i, -1)$ after Before Section

Before Section ← the first new link

If First Link.Point = $i + 1$ and SecondLink.Point = $i + 2$ then add the links $(i + 2, -1) \rightarrow (i + 1, -1)$ →after BeforeSection

Before Section ← the first new link

If First Link.Point $= i + 1$ and Second Link.Point = i then add the link $(i + 2, -1)$ after Before Section

Before Section ← the new link

for any link L between Before Link and After Link do

L.Position ← $-L$.Position

L.Point← $2i + 1 - L$.Point

If Before Section.Point $= i - 1$ then add the links $(i, -1,) \rightarrow (i + 1, -1)$ after Before Section

else

add the links $(i + 1,1,) \rightarrow (i, 1)$ after Before Section

If After Section.Point = $i - 1$ then add the links $(i + 1, -1,) \rightarrow (i, -1)$ after BeforeSection

else

add the links $(i, 1) \rightarrow (i + 1,1)$ after Before Section

In order to obtain the Negative Half Twist (σ_i, g) function, one has to use the Positive Half Twist (σ_i, g) function while replacing the last twoif statements with the following:

If Before Section.Point = $i - 1$ then add the links $(i, 1,) \rightarrow (i + 1,1)$ after Before Section

else

add the links $(i + 1, -1,) \rightarrow (i, -1)$ after Before Section

If After Section.Point $= i - 1$ then

add the links $(i+1,1,) \rightarrow (i,1)$ after BeforeSection
else
add the links $(i,-1,) \rightarrow (i+1,-1)$ after Before Section

5. Results and Discussion

Let σ_i be the negative half-twist acting on the g-base. Then, the prefix sequence we have to add is as follows:
1: $(i,1,) \rightarrow (i+1,1)$, If the local section of the path is connected to a point to the left of the point i.
2: $(i+1,-1,) \rightarrow (i,-1)$, If the local section of the path is connected to a point to the right of the point $i+1$.
The postfix sequence we have to add is as follows:
1: $(i+1,1,) \rightarrow (i,1)$ If the local section of the path is connected to a point to the left of the point i.
2: $(i,-1,) \rightarrow (i+1,-1)$, If the local section of the path is connected to a point to the right of the point $i+1$.
Now, we will consider the case where the link $(-1,0)$ is followed immediately by the local section of the path. In this case, we alter the path homotopically so that the preceding link to the local section of the path will not be $(-1,0)$ .By doing this, we will reduce the problem to the one already proved by the above propositions,hence, we will be able to use the same algorithmic methods in these cases.
We have 2 possible different cases:

(i) If we have the sequence $(-1,0) \rightarrow (i,0)$,then we add a link just below the point to the left of the local sectionw hich is $(i-1,-1)$ .As a result,this point is the one preceding the local section(see figure(a)).

(ii) If we have the sequence$(-1,0) \rightarrow (i+1,0)$, then we add a link just below the point to the right of the local section which is $(i+2,-1)$.As a result, this point is the one preceding the local section.

6. Conclusion

We would like to state here that although for very long braid words this algorithm running time is long, since the complexity of the g-base presentation grows with the length of the braid word, for short braid words we have obtained a quick algorithm in comparison with other methods. This is true because of the fact that Garsides algorithm involves the replacement of the generators in a negative power by a sub word of size $n(n-1)$,where n is the number of strings in the braid group (although, for variations of his algorithm the size of the fundamental word Δ_n,and because each step of reduction in Dehornoys algorithm involves the insertion of atleast two sub words of $O(n)$ length.In our algorithm for short braid words over a large number of strings we obtain a very short description of the g -base,that yields a very fast algorithm.This means that we have presented a very useful and practical algorithm.We also would like to point out that we have an implementation of the algorithm on a computer.

REFERENCES

Artin E. (1947), Theory of braids, Ann.Math., Vol. 48, pg. 101-126. Anshel I, Anshel M, Fisher B & Goldfeld D (2001). New Key Agreement Protocols in Braid Group Cryptography, Topics in Cryptology-CT-RSA (San Francisco, CA) , Springer Berlin, pg. 13-27. Birman J.S., Ko K.H. & Lee S.J. (1998). A new approach to the word and conjugacy problems in the braid groups, Adv.Math., Vol. 13 , pg. 322-353. Dehornoy P. (1995). From large cardinals to braids via distributive algebra, J. Knot Theory and Ramifications, Vol 4, Issue1,3 pg. 3-79 Dehornoy P. (1997). A fast method for comparing braids. Adv. Math., Vol. 125, Issue 2, 200235. Elrifai E.A. & Morton H.R. (1994). Algorithms for positive braids. Quart.J.Math. Oxford Ser.(2). Vol. 145, pg. 479-497. Garside F.A. (1969). The braid group and other groups. Quart.J.Math.Oxford Ser. (2). Vol.78, pg. 235-254. Jacquemard A. (1990). About the effective classification of conjugacy classes of braids. J.Pure. Appl. Alg., Vol. 63, pg. 161-169. Kang E.S., Ko K.H. & Lee S.J. (1997). Band-generator presentation for the 4-braid group. Top. Appl., Vol. 78, pg. 39-60. Moishezon B. & Teicher (1988). M. Braid group techniques in complex geometry I, Line arrangements in P^2 , Contemporary Math, Vol. 78, pg. 425-555.