Solving The Word Problem

Paul Vrbik

April 7, 2006

Introduction

Dealing with braids or knots, is something very natural and intuitive for a human being to understand. However, describing these very abstract topological objects to a computer has been somewhat of a challenge. Motivated by this, Emil Artin introduced an algebraic representation of the braids in an effort to study their characteristics. In particular, Artin was interested in determining if two braids were equivalent. This paper will outline two approaches to solving the braid equivalence problem, or the *word problem* as it is called amongst knot theorists.

Properties of Braids

It is first necessary to define a braid and the operations that we can do on a braid.

Braid The most typical way to describe a braid is to regard it as pieces of intertwining string that are attached to top and bottom parallel "bars" in such a way that no string can loop back on itself. That is, if you were to regard each string as the path of some particle, this particle's distance from the bar of origin would be strictly increasing.

Braid Equality We say two strings as equal if we can rearrange the strings in the two braids to look the same without passing any string through one another or themselves.

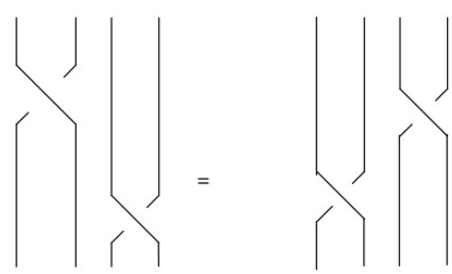

Figure 1: Two equivalent four braid strands

Product of Braids The product of two braids ($B_1 \times B_2$) with an equivalent amount of strings is obtained by attaching the "bottom" of B_1 with the "top" of B_2.

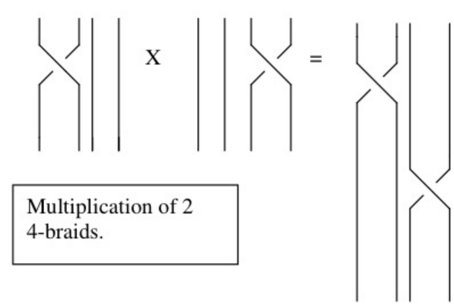

Figure 2: Braid Product

Braid Inverse The inverse of a braid is its reflection in the horizontal axis.

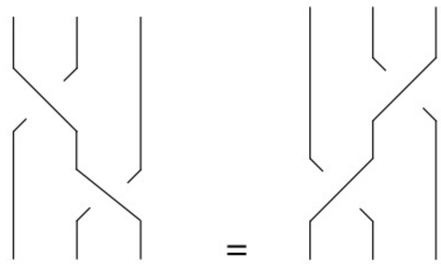

Figure 3: Braid inverse

Identity Braid The identity braid of n strings is the braid whose strings do not cross.

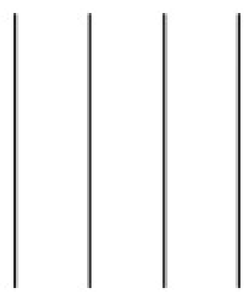

Figure 4: The identity braid on four strings

We now have all the ingredients to form an algebraic group. Moreover, we can present the braid group completely in terms of generators as defined by Artin in (1925).

Artin's Presentation

B_n, the braid group consisting of braids with n strings each are given by the following generators and relations. These relations are illustrated in Figure 5 through Figure 6.

Generators $\sigma_1, \sigma_2, \sigma_3, ...\sigma_{n-1}$ where σ_i is defined as passing the ith string underneath the $i^{th}+1$ string where all other strings remain uncrossed.

Relations .

1. $\sigma_s \sigma_t = \sigma_t \sigma_s$ for $|t-s| > 1$
2. $\sigma_s \sigma_t \sigma_s = \sigma_t \sigma_s \sigma_t$ for $|t-s| = 1$

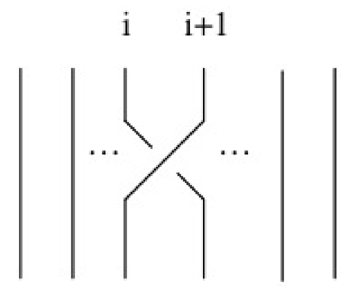

Figure 5: σ_i

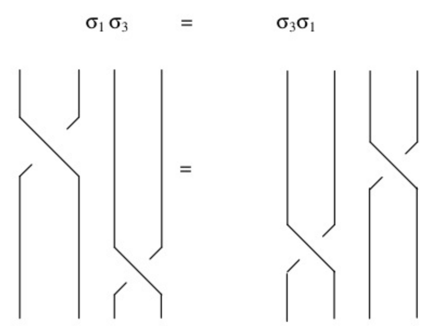

Figure 6: Example of relation 1

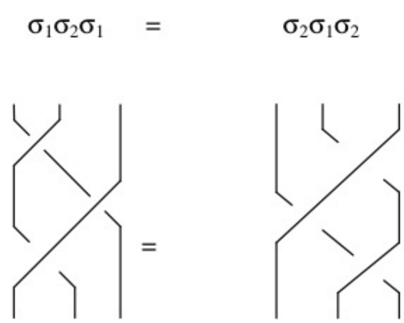

Figure 7: Example of relation 2

BKL-Presentation (Birman, Ko, Lee)

It is worth mentioning that there is an alternate presentation of the braid group B_n with generators a_{ts}, $1 \leq s < t \leq n$ where a_{ts} represents the braid in which the t^{th} string crosses over the s^{th} string while the s^{th} and t^{th} string cross in-front of all intermediate strings.

The relations of the BKL-Presenation are the following:

1. $a_{ts}a_{rq} = a_{rq}a_{ts}$ if $(t-r)(t-q)(s-r)(s-q) > 0$

2. $a_{ts} = a_{sr} = a_{tr}a_{ts} = a_{sr}a_{tr}$

Although the BKL-Presentaion has no more power then the Artin Presentation, some find it easier to work with.

Decision Problems

Whenever we are given a group presentation we are led to considering "decision problems" for the group. That is, we would like to know if a generated element is the identity or not. The deciding the braid group is called the *word problem* and it is defined as follows.

Word Problem Given a presentation of the braid group B_n and a braid $b \in B_n$ presented as a product of the generators and their inverses, can we decide in a *finite* number of steps if b is the identity in B_n?

Note that this problem is equivalent to solving braid equivalency since $b_1 = b_2 \Leftrightarrow b_1 b_2^{-1} = identity$.

Artin Combing

Every braid on n strands consists of a one-to-one correspondence between two sets of n items, (plus some topological information). If we forget this topological information, then every braid yields a one-to-one correspondence of n-items. These are precisely the elements of S_n. It turns out this assignment is in fact a surjective group homomorphism $B_n \to S_n$. The Kernel of this group is called the pure braid group.

More simply, if we numbered the spots on the "top" and "bottom" bars where the strings attach, a braid in which each string is attached to the same position on the "bottom" as it is on the "top" is pure braid. By studying this subset Artin found a solution to word problem called Artin Combing.

There is a iterative algebraic method for doing this combing however intuitively we are doing the following:

1. For any braid $b \in B_n$ remove the $n-th$ band of the braid

2. "Comb" the remaining strings by straightening each strand, deviating from this straightening only when it is necessary to cross under or over another string.

3. Re-add the n^{th} string in a manner which follows the straightening described in (2)

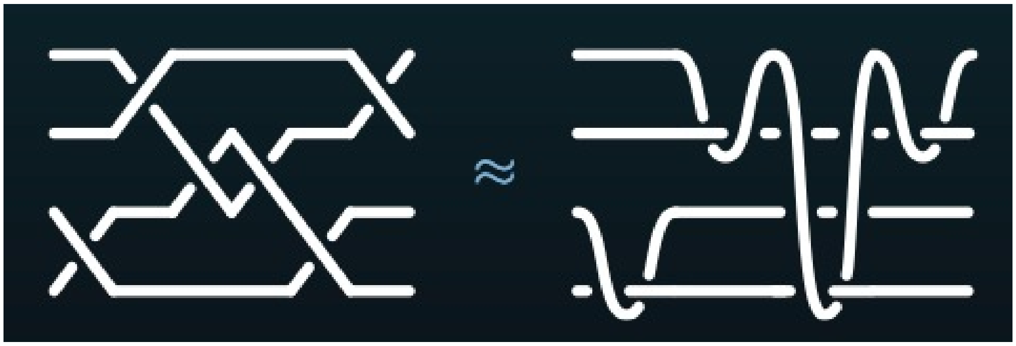

Figure 8: Example of Artin combing

This process is illustrated in Figure 8.

Artin proved that this process would result in a unique presentation of the braid. Which of course solves the braid word problem. The next natural question is to ask how to implement Artin Combing as a computer program. The answer to this is simple, we don't. It has been shown that any algorithm that models Artin Combing will have complexity given by n^n where n is the number of strands in the braid. This is a computational disaster, for example to determine if two braids from B_{15} were equal would require 15^{15} calculations. Suppose we had a super fast computer that could do 1,000,000 calculations a second, approximating $15^{15} = 437,893,890,380,859,375 > 4 \times 10^{17}$ it would take

$$\frac{4 \times 10^{17}}{10^6 \times 60 \times 60 \times 24} = 4,629,629 : days = 12,718 : years$$

to compute a solution.

Word reversing

A more novel approach to deciding braids uses word reversing. We say that $w \curvearrowright w'$ (said w reverses to w') holds if w' is obtained by iteratively applying:

- deleting some $\sigma_i^{-1}\sigma_i$
- replacing some σ_i^{-1} with $\sigma_j\sigma_i^{-1}$ if $|i-j| \geq 2$
- replacing some σ_i^{-1} with $\sigma_j\sigma_i\sigma_j^{-1}\sigma_i^{-1}$ if $|i-j| = 1$

Then using Garside Theory we can show that

- Every braid reverses to a unique word of the form uv^{-1} with u, v positive (that is u and v are words containing no inverses).
- the braid word w represents the identity in $B_n \Leftrightarrow$ for some positive u, v, $w \curvearrowright uv^{-1}$ and $v^{-1}u \curvearrowright identity$

It is easy to see how such an algorithm could be implemented by a computer and in fact one has been done with a complexity of n^2 for braids in B_n which is good by any standards.