2003

Braids and Juggling Patterns

Matthew Macauley
Harvey Mudd College

Braids and Juggling Patterns

by
Matthew Macauley
Michael Orrison, Advisor

Advisor: ____

Second Reader: ____
(Jim Hoste)

May 2003
Department of Mathematics
Harvey Mudd College

Abstract

Braids and Juggling Patterns

by Matthew Macauley

May 2003

There are several ways to describe juggling patterns mathematically using combinatorics and algebra. In my thesis I use these ideas to build a new system using braid groups. A new kind of graph arises that helps describe all braids that can be juggled.

Table of Contents

List of Figures

Acknowledgments

I would like to thank my advisors Michael Orrison and Jim Hoste for helping me with my research. I would also like to thank fellow jugglers and mathematicians Ron Graham and Will Murray for letting me bounce ideas off of them and giving me feedback.

Chapter 1

Introduction

Mathematics and juggling have both been around for thousands of years. The oldest known record of juggling was recovered from a burial site in Egypt that is nearly four thousand years old. Evidence of juggling has been uncovered in the histories of many different civilizations, including ancient China, Europe, Asia, and the Middle East. Though both have elaborate histories, mathematics and juggling have only become intertwined within the last few decades. The big breakthrough came in 1985 when three different sources independently invented a mathematical notation for juggling patterns. These groups were Caltech students Bengt Magnusson and Bruce Tiemann, the trio of Mike Day, Colin Wright and Adam Chalcraft from Cambridge, and Paul Klimak at the University of California, Santa Cruz. The notation, called siteswap notation, describes a juggling pattern by a sequence of digits that denote the height of each throw.

There are several different mathematical aspects of juggling that have been examined. One natural topic is the physics of juggling. Magnusson and Tiemann published a paper [2] on this subject in 1989. A decade later, Jack Kalvin, a professional juggler who holds a mechanical engineering degree from Carnegie Mellon University, wrote two papers [6, 7] in the late 1990s about the physics of juggling. One of his results is determining how many balls a human being can physically juggle. He uses the first four time derivatives of the motion of the juggler's throwing hand to conclude that it should be physically possible for a human being to juggle up to fifteen balls. However, the current record stands at ten, where the juggler must make $2n$ catches of n balls for it to qualify as a "juggle." Several individuals have been able to "flash" twelve balls, which means each ball is thrown and caught once. Recently, Albert Lucas successfully flashed fourteen rings. Nobody else has flashed more than twelve.

There are connections between siteswap notation and physics. However, siteswap notation in itself poses many interesting algebraic and combinatorial questions. One of the big papers in this area was co-authored by Joe Buhler, David Eisenbud, Ron Graham and Colin Wright in 1994 [3]. They used some innovative techniques to count the number of siteswap patterns of a fixed length given a certain number of balls. They wrote a second paper that generalized the mathematics they had invented in their first paper to any arbitrary partially ordered set. Another important paper was written by Richard Ehrenborg and Margret Readdy in 1996. Siteswap notation can be generalized to describe patterns, called multiplex patterns, where a hand can throw more than one ball at a time. Ehrenborg and Readdy provided connections between multiplexed patterns and Stirling numbers of the second kind, and the affine Weyl group $\tilde{A}_{p-1}$.

Eighteen months prior to writing this thesis, I had the idea of studying siteswap patterns by looking at the braid formed by attaching strings to the ends of the balls. A more natural way to make a braid when juggling is to walk forward and look at the braid formed by the paths of the balls traced out in space. I searched far and wide to see if this had been done before and found nothing to suggest that it had. A year later when I began this thesis, I searched again. This time, I found a website linked from the juggling club at Brown University [8]. For a project in an undergraduate topology class, two students had looked at braid groups and realized that juggling patterns can be represented as braids in this manner. They gave a few examples of the braids of some simple siteswap patterns and discussed some general concepts. A few months later, Burkard Polster published a book about the mathematics of juggling that was intended to be a collection of just about everything that has been done so far with mathematics and juggling [9]. He talks about braids and juggling for about five pages, mostly summarizing what had been said in [8], and proving the theorem that with enough hands, any braid can be juggled. The result is intuitive, and follows from the fact that any braid can be generated by a series of crossings of adjacent strings.

In this paper I give some background of the mathematics of juggling needed to study the braids of juggling patterns, which is the focus of Chapter 6. At the end of each chapter, I pose some questions that arose when working on this paper. They are not necessarily extremely difficult, but just ideas that I had but never got around to when working on this paper. Some of them are natural generalizations that may or may not have promise. However, I am confident that there is lots of room for future research about the mathematics of juggling.

Chapter 2

Siteswap Notation

In order to examine the mathematics of juggling we must set some rules for what constitutes a valid juggling pattern with b balls. First we need a notion of equal time intervals, or beats. Throws may only be made on a beat, and at most one ball is thrown or caught each beat. In practice, once a ball lands, it remains in the juggler's hand for at least a beat before it is thrown again. However, once a ball is thrown we will only concern ourselves with the number of beats before that ball is thrown again. Any juggling pattern can be described by the function $f : \mathbb{Z} \longrightarrow \mathbb{Z}$ where the ball thrown at time t is thrown next at time $f(t)$. If no ball is thrown at time t_0, then $f(t_0) = 0$. Because balls are not thrown back in time, to each pattern we assign a non-negative height function

$$h : \mathbb{Z} \longrightarrow \mathbb{N} \cup \{0\}$$

defined by

$$h(t) = f(t) - t.$$

For every beat t, $h(t)$ is the number of beats from when the ball thrown at time t will be thrown again. Notice that a height of 0 means that there was no throw on that beat. In this paper we shall only consider patterns with periodic height functions, which means that for some $n > 0$, $h(t) = h(t+n)$. For any positive n that satisfies this condition, we define the function

$$\overline{h} : \mathbb{Z}/n\mathbb{Z} \longrightarrow \mathbb{N} \cup \{0\}$$

by $\overline{h}(t) = h(t)$. Every n-periodic pattern can be described as a length-n string of non-negative digits, namely

$$\overline{h}(0)\ \overline{h}(1) \dots \overline{h}(n-1).$$

This is called *siteswap notation*, and a juggling pattern in this form is called a *siteswap pattern*. In practice, most siteswap patterns that people juggle do not have throws higher than a seven. Even the most advanced jugglers rarely will juggle a pattern with a throw higher than a nine. There are a few exceptions, and for these patterns, letters are used for higher digits, like A=10, B=11, C=12, and so on.

Several properties are immediate consequences of the construction of siteswap notation. Since two balls cannot land on the same beat, for any positive integer i, $h(t) - h(t+i) \neq i$. Also, with b balls, the average height must be b, and so the average of the digits of a siteswap pattern is the number of balls in that pattern. To give a few examples, some common three-ball patterns are 3, 51, 423, 441, 504, 531, and 51414.

One way to think of a siteswap pattern is to use a *profile braid*. For a siteswap pattern with height function h, an arc is drawn on the real line from t to $t + h(t)$ for each integer t. The profile braid depicts the paths of the balls of a pattern as seen from the side as the jugglers walks forward. Figure 2.1 is the profile braid of the pattern 441. Profile braids will be discussed in more detail in Chapter 5. Notice that the patterns 441, 414, and 144 all yield the same profile braid, only shifted. We will call two siteswap patterns with height functions h_1 and h_2 *equivalent* if for some integer k, $h_1(t+k) = h_2(t)$ for all t. Hence, 441, 414, and 144 are all equivalent.

Siteswap notation does not describe how many hands are used to juggle the pattern or the locations of the hands. The standard juggling method uses two hands that alternate throwing the balls. Balls are caught from the outside of the pattern and thrown from the inside. Observe that using this convention, throws with even heights don't switch hands while throws with odd heights do. In this

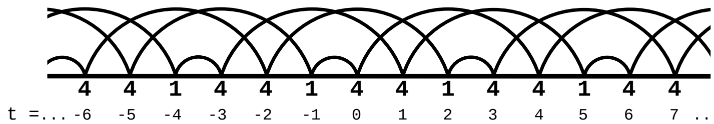


Figure 2.1: Profile braid of the pattern 441.

paper we will consider the simplest model, one-handed juggling.

There is a theorem attributed to Buhler, Eisenrod, Graham, and Wright first published in [3] that counts the number of siteswap patterns of a given length.

Theorem 2.1 *The number of distinct height functions of all length-n siteswap patterns with less than b balls is b^n.*

Though this formula is pretty, it is impractical because it counts equivalent patterns separately. A short proof of Theorem 2.1 will be given in the next chapter after laying out some more definitions. However, there is an immediate corollary. We define the *period* of a siteswap pattern to be the smallest positive integer k such that $\overline{h}(t) = \overline{h}(t+k)$ for all t. For example, the length-six pattern 441441 has period 3.

Corollary 2.2 *The number of siteswap patterns of period n with exactly b balls, up to equivalence, is*

$$\frac{1}{n}\sum_{d|n} \mu(n/d)\left((b+1)^d - b^d\right).$$

Here, μ is the Möbius function, which is defined on the natural numbers by

$$\mu(n) = \begin{cases} 1 & \text{if } n \text{ is square-free with an even number of distinct prime factors.} \\ -1 & \text{if } n \text{ is square-free with an odd number of distinct prime factors.} \\ 0 & \text{if } n \text{ is not square-free.} \end{cases}$$

An integer n is said to be square-free if there does not exist a square of a smaller integer that divides n.

Proof: By Theorem 2.1, there are $(b+1)^n - b^n$ different height functions that describe a b-ball juggling pattern. However, this over-counts the number of siteswap patterns because different height functions can correspond to equivalent siteswap patterns. Let $M(n,b)$ be the number of siteswap patterns of period n with exactly b balls. For every divisor d of n, there are n/d equivalent patterns of period d, related by shifting the digits. We can count the length-n height functions by summing over all height functions of length d for each $d|n$. Thus

$$(b+1)^n - b^n = \sum_{d|n} d\ M(d,b)$$

We can solve for $M(n,b)$ using a combinatorial technique called a Möbius inversion to get

$$M(n,b) = \frac{1}{n} \sum_{d|n} \mu(n/d) \left((b+1)^d - b^d\right).$$

Möbius inversion is described in [13]. □

Chapter 3

Symmetric Groups

3.1 Siteswap Permutations

Let $\mathbf{S}_n$ be the group of all permutations of the set $[n] = \{0, 1, \ldots, n-2, n-1\}$. Every periodic height function $\overline{h}$ of a length-n pattern naturally corresponds to a permutation that sends each integer $i \in [n]$, the domain of $\overline{h}$, to $i + \overline{h}(t)$ modulo n. One way to think of this is a beat $t \equiv i \pmod{n}$ is sent to the beat reduced modulo n where the ball thrown on t will be thrown next. Notice that if a no ball is thrown on beat i, then the permutation sends i to itself. This is a permutation because no more than one ball is thrown each beat and at most one ball lands each beat. We can define this map

$$p : \{\text{length } n \text{ siteswap patterns}\} \longrightarrow \mathbf{S}_n$$

because each siteswap pattern corresponds to a unique $\overline{h}$ function. However, p is not injective. Several siteswap patterns can give rise to the same permutation. As an example, we'll compute the permutation of the pattern 441, which is given by the $\overline{h}$ function

$$\overline{h}(t) = \begin{cases} 4 & t \equiv 0 \pmod{3} \\ 4 & t \equiv 1 \pmod{3} \\ 1 & t \equiv 2 \pmod{3} \end{cases}$$

A ball thrown on a beat $t_0 \equiv 0 \pmod{3}$ is thrown again on a beat $t_0 + h(t_0) \equiv 1 \pmod{3}$. A ball thrown on a beat $t_0 \equiv 1 \pmod{3}$ is thrown again on a beat $t_0 + h(t_0) \equiv 2 \pmod{3}$. This means that the permutation of 441 is $(0\ 1\ 2)$. However, the permutations of 441, 741, and 471 are all $(0\ 1\ 2)$. Not only is p not injective, but equivalent juggling patterns may even have different permutations. As an example, the pattern 423 has permutation $(0\ 1)\,(2)$ but the equivalent pattern 342 has permutation $(0)\,(1\ 2)$. In order to have equivalent patterns correspond to the same permutation, we need to put an equivalence relation on the set of permutations.

We shall call two permutations in S_n equivalent if they describe equivalent siteswap patterns. For any siteswap pattern with permutation π, the equivalent siteswap pattern obtained by beginning with the ith digit corresponds to the permutation resulting in incrementing each digit in the cycle notation of π by i modulo n. For example, $p\,(33342) = (0\ 3\ 2)\,(1\ 4)$ and $p\,(33423) = (1\ 4\ 3)\,(2\ 0)$.

Later we will see that if two profile braids as described in Chapter 5 are in the same same orbit of Z_n acting on the set of period-n profile braids, they have the equivalent permutations. The converse is false.

3.2 Interesting Questions

1. How many distinct elements of S_n are there up to equivalence?

2. If a pattern has permutation π, then what can we say about patterns that have permutation π^{-1}?

3. Are there any similarities between patterns whose permutations are conjugate?

Chapter 4

Stack Notation

At any time during a juggling pattern, we can make an ordered list of the balls in the air based on the order that they will land. If we assign each ball a unique color, then we can draw a vertical stack of colors in order of the landing times of the balls, with the lowest ball landing first. Each time we throw a ball, that ball gets inserted somewhere into the stack of the other $b-1$ balls. There are b slots to insert the new ball, and if we label them from bottom-to-top $1, 2, \ldots, b$, we can create a length-n sequence for every siteswap pattern. If no ball is thrown at a beat, then the digit at the beat is 0 and the stack remains unchanged. We call one of these sequences the *stack sequence* of a juggling pattern.

Let's take a look at some examples and see how to derive a stack sequence from a siteswap pattern. First write a few periods of the siteswap sequence, and assign each ball a different color. For example, if we have three balls, blue red and green, denoted B, R, and G, the pattern 441 would look like this:

$balls$	R	G	B	B	R	G	G	B	R
$siteswap$	4	4	1	4	4	1	4	4	1

We will write a stack between each beat based on the order that the balls will land. The bottom layer is simply the ball that will be thrown at the next beat. The second layer is the next different colored ball that will be thrown after the next beat. Repeat until all b layers have been filled out. If we do this algorithm with 441, we get 331, as shown in Figure 4.1.

Thus the stack sequence for 441 is 331. To go from the stack sequence to siteswap notation, assign colors to the first stack, and then everything else is determined.

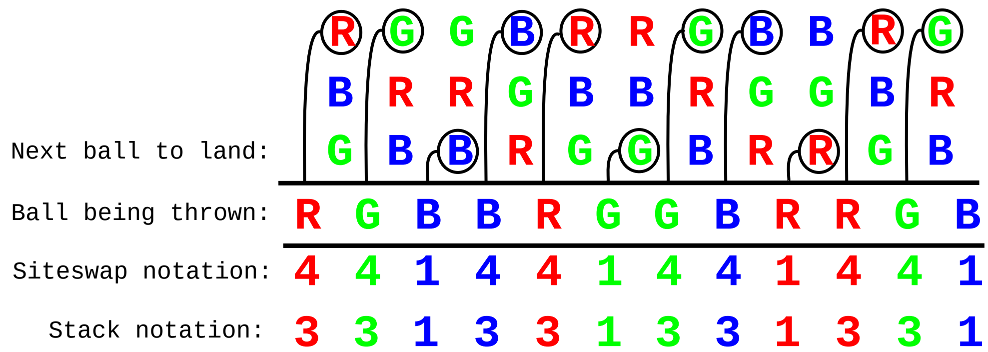


Figure 4.1: Relation between siteswap and stack notation.

The ball at the bottom at the stack is always the next one to land, and the balls above maintain their relative order until they land.

Theorem 4.1 *There is a bijection between the number of length-n siteswap patterns and length-n stack sequences.*

This follows from the existence of the algorithms that produce a unique stack sequence from a siteswap pattern, and a unique siteswap pattern from a stack sequence.

Lemma 4.2 *A length-n juggling pattern has exactly b balls if and only if the highest digit in its stack sequence is b.*

Proof: Consider a b-ball pattern, and let b' be the highest digit in the stack sequence. Since there are b balls in the stack and balls never move up in the stack, the top ball must have been thrown to that position, thus $b' \geq b$. Conversely, since the highest digit in the stack sequence is b', there must be at least b' balls, so $b' \leq b$. This means that $b' = b$. □

Stack notation gives a simple proof to the Theorem from Chapter 1 about counting siteswap patterns, or height functions.

Corollary 4.3 *There are* $(b+1)^n$ *siteswap patterns of length-*n *using at most* b *balls, and* b^n *zeros are disallowed.*

Proof: The number of length-n stack sequences with at most b balls is the number of sequences using the digits $\{0, 1, \ldots, b\}$, which is $(b+1)^n$. If zeros are not allowed, then it is just b^n. Because there is a bijection between stack sequences and siteswap patterns (or height functions), this is also the number of siteswap patterns. $\square$

Chapter 5

Profile Braids

Profile braids were mentioned briefly in Chapter 1. The idea is to draw the paths of the balls of a siteswap pattern as seen from the profile view as the juggler walks forward at a constant rate. We do this by assigning one throw to each integer on the real line. For a throw at time t of height h, we draw an inverted parabola from t to $t + h$. The resulting diagram has n distinct lines. If the pattern has no zeros in its siteswap representation then at each beat, exactly one parabola begins and ends. Two profile braids are equivalent if and only if their siteswap patterns are equivalent. Note that this is the same as being able to move one to the other by a cyclic shift. A full profile braid is one such that there are no empty beats (no zeros in the siteswap sequence). In this paper we are only concerned with periodic juggling patterns. Figure 5.1 shows the profile braid for the pattern 441.

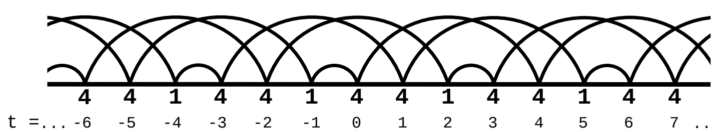


Figure 5.1: Profile braid of the pattern 441.

Profile braids bring up lots of combinatorial questions. First we will note a few properties of profile braids. We can assume that if a parabola in a profile braid crosses another ball path, then there are exactly two points of crossing of the path and the parabola. This is shown in Figure 5.2.

If a ball path crosses into the interior of a parabola, then it must eventually

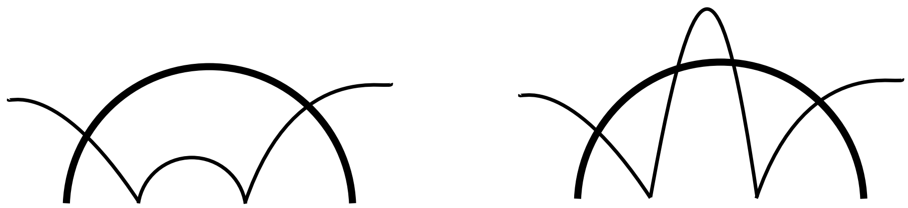

(a) (b)

Figure 5.2: A ball crossing the parabola of another ball's path.

leave, as shown in (a). The laws of physics dictate that a ball that is thrown after and lands before another ball must have been thrown to a lower height. Thus the situation in (b) is impossible. This means that each parabola can have either zero or two points of intersection with any given path in the profile braid. It follows that each parabola has an even number of intersections. We can use a juggling pattern's stack sequence to count the number of crossings per cycle of the profile braid of a juggling pattern.

Theorem 5.1 *Given a stack sequence of a juggling pattern, the sum of one less than each digit equals the number of crossings per cycle of that pattern's profile braid.*

Proof: Suppose a ball is thrown to the ith place in the stack. It crossed each of the $i-1$ balls beneath it. This means that each digit i in the stack sequence corresponds to $i-1$ crossings. This does not double-count crossings or miss any crossings, because of the impossibility of the situation in Figure 5.2(b). □

5.1 Polya Theory

Polya theory is a branch of mathematics where a counting problem can be modeled with a group acting on a set of objects. Two elements are considered equivalent if and only if they're in the same orbit of the group action. We now present an example of a theorem in Polya theory that will be useful later in counting profile braids. A *k-ary necklace* is an ordered set of n elements, each assigned a color from the set $\{C_1, C_2, \ldots, C_k\}$. Think of a k-ary necklace as a circular arrangement of colored beads. Two necklaces N_1 and N_2 are equivalent if there exists a circular rotation of N_1 so that N_1 is colored just like N_2. Mathematically, this means that there is an element r in the additive group Z_n acting on the set of necklaces, such that $rN_1 = N_2$.

Theorem 5.2 *The number of distinct n-bead k-ary necklaces up to equivalence is*

$$\frac{1}{n}\sum_{d|n}\phi(n/d)k^d.$$

Proof: Let X be the set of all colorings of n-bead necklaces using at most k colors. Two colorings are equivalent if and only if there is a rotation that maps one to the other, which means they are in the same orbit of the group Z_n acting on the set of colorings. To determine the number of colorings up to equivalence, we need to count the number of orbits, which will be denoted $|\mathcal{O}|$. An algebra theorem tells us that

$$|\mathcal{O}| = \sum_{x\in X}\frac{1}{|\operatorname{orb}(x)|} = \sum_{x\in X}\frac{|\operatorname{stab}(x)|}{|Z_n|} = \frac{1}{n}\sum_{x\in X}|\operatorname{stab}(x)|$$

where $\operatorname{orb}(x)$ is the orbit containing x and $\operatorname{stab}(x)$ is the stabilizer of x. Instead of counting the size of the stabilizer of a necklace summed over all necklaces, we can count the number of necklaces fixed by each rotation and sum over all rotations:

$$|\mathcal{O}| = \frac{1}{n}\sum_{r\in Z_n}|\text{fixed pts(r)}|.$$

Let r be the element in Z_n corresponding with a rotation by $2\pi/n$. Only the one-color necklaces will be fixed by r, or by any power of r relatively prime to n. For an integer i such that $\gcd(n,i) = d \neq 1$, the rotation r^i fixes exactly the necklaces where every dth bead is the same color. In this case there are d different sets each containing n/d beads, and so we have k^d possible colorings that are fixed by r^i. There are $\phi(n/d)$ rotations r^i such that $\gcd(n,i) = d$. Summing over all rotations, we get

$$\frac{1}{n}\sum_{d|n}\phi(n/d)k^d$$

which proves the theorem. □

We can count the number of distinct profile braids of a given period by constructing a bijection between profile braids and necklaces. This leads to the following result.

Theorem 5.3 *The number of distinct full profile braids with period n and at most b balls is*

$$\frac{1}{n}\sum_{d|n}\phi(n/d)b^d$$

Proof: We can view a stack sequence as a necklace; two stack sequences lead to equivalent profile braids if and only if they are related by cyclic shift. Thus the number of distinct profile braids of period n is the number of length-n necklaces using b colors. The result follows immediately. □

Profile braids are important because if we include information about how the paths intertwine with each other, we can put an algebraic structure on the juggling patterns by using braid groups. In one sense, this provides a way to describe the topology of juggling patterns. This will be the focus of the next chapter. But first, there is one more interesting combinatorial theorem about profile braids. For all period-n profile braids, we want to determine the average number of crossings per period. To find this, we start by counting the number of crossings per period and sum this over all profile braids, and then just divide by the number of profile braids as given by Theorem 5.3.

Theorem 5.4 *: Summing over all length-n profile braids with at most b balls, the number of crossings per period is*

$$\frac{b-1}{2}\sum_{d|n}\phi(n/d)b^d$$

Proof: The number of crossings per period of a length-n profile braid is given by

$$\sum_{i=0}^{n-1}(s_i-1)$$

where $s_1 s_2 \ldots s_n$ is the pattern's stack sequence. Because of the bijection between stack sequences and necklaces, we can represent a pattern that has stack sequence $s_1 s_2 \ldots s_n$ as a necklace where we label (or "color") the ith digit $s_i - 1$. The number of crossings per period in a pattern's profile braid is equal to the sum of the labels of the corresponding necklace. Summing the number of crossings over all patterns of period n amounts to adding up the sums of each n-bead necklace over all "colorings" from the set $\{0, 1, \ldots, b-1\}$. By symmetry, each digit occurs with equal frequency. Since each pattern has n digits, there are $\sum \phi\left(\frac{n}{d}\right) b^d$ digits, with each value from $\{0, 1, \ldots b-1\}$ occurring with equal frequency. Thus the average value is $\frac{b-1}{2}$ and summing over all necklaces gives us the desired result. □

Corollary 5.5 *The average number of crossings per period of a length-n profile braid with at most b balls is n(b-1)/2.*

Proof: Dividing the sum from Theorem 5.4 by the sum from Theorem 5.3 gives the desired result. □

5.2 Interesting Questions

1. The profile braid of the pattern 411222 is symmetric. What property must a juggling pattern have to be symmetric?

2. The profile braids of 411231 and 411321 are mirror images of each other. Find necessary and sufficient conditions for two juggling patterns to have profile braids that are mirror images of each other.

3. We can model bounced throws by allowing both regular and inverted parabolas in the profile braid. Prove similar results for this generalization.

Chapter 6

Braids and Juggling

Suppose we juggle an n-ball siteswap pattern as we walk forward. The paths of the balls will trace out a braid in 3-space with n strings. A braid can be represented algebraically as an element of a braid group. Braid groups give us a way to study the topology of juggling patterns. For more on braid groups, see [5].

6.1 The Braid Group

Definition 6.1 *Consider two planar parallel segments X and Y in $\mathbb{R}^3$ each containing n distinct points, $\{x_i\}$ and $\{y_i\}$. An n-braid is a collection of n curves $\{b_i\}$, where $b_i : [0,1] \longrightarrow \mathbb{R}^3$ for each b_i and the following conditions hold:*

1. *Each b_i has one endpoint at one of the x_i's and one endpoint a y_j.*
2. *All the b_i's are pairwise disjoint.*
3. *Every plane parallel to X and Y and normal to the plane containing them either intersects each b_i at exactly one point or is disjoint from all of them.*

The easiest way to draw a braid is to draw its projection onto a plane and denote which strand is on top at each crossing. For each braid, we can choose a projection such that no three strands meet at any one point, and any two strands intersect at a finite number of points. The first diagram in Figure 6.1 is a braid, but the second is not because it violates the third property.

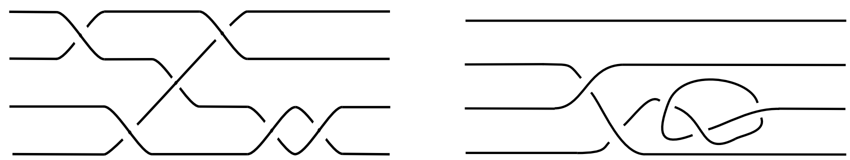


Figure 6.1: A braid on four strings, and an illegal braid.

This will be our conventional way of drawing braids. We can put an algebraic structure on the set of braids on n strings, or n-braids, with a finite number of crossings when projected onto a plane. Any braid can be generated by repeatedly crossing adjacent strings. Starting from one end of the braid and moving to the other, we can list all the crossings one at a time as given by the following rules: At any point, if the current ith strand from the bottom crosses under the $(i+1)$th strand, call it σ_i. If it crosses over, call it σ_i^{-1}. Figure 6.2 is an example of this. Any braid can be expressed as a word of the σ_i's and σ_i's.

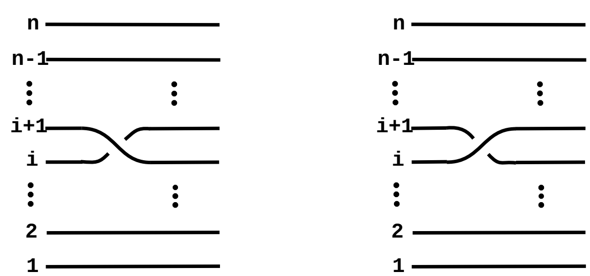


Figure 6.2: The ith generator of the braid group, σ_i, and its inverse.

Two braids are considered equivalent if they can be expressed by the same word. There are two relations that can be useful when determining whether two braids are equivalent. The first braid relation is $\sigma_i\sigma_j = \sigma_j\sigma_i$ if $|i-j| \geq 2$. This is intuitive, because two crossings far enough apart can be moved horizontally independently as shown by the diagram in Figure 6.3.

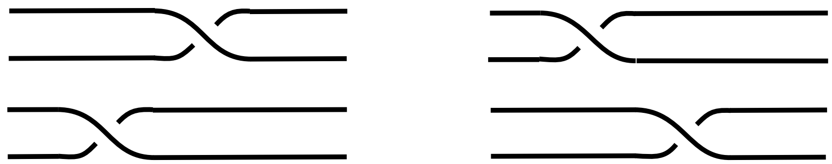


Figure 6.3: The first braid relation: $\sigma_i\sigma_j = \sigma_j\sigma_i$ if $|i-j| \geq 2$.

The second braid relation is $\sigma_k\sigma_{k+1}\sigma_k = \sigma_{k+1}\sigma_k\sigma_{k+1}$. In knot theory, this is called the third Reidemeister move, which allows a strand to be moved past a crossing. Figure 6.4 gives an example of this relation.

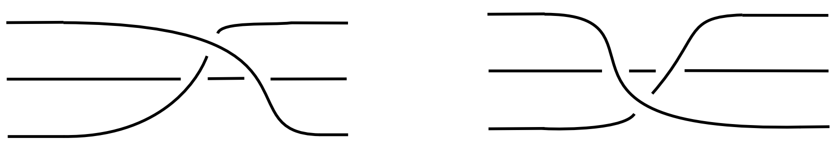


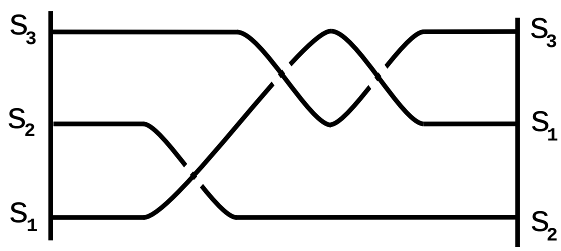
Figure 6.4: The second braid relation: $\sigma_k\sigma_{k+1}\sigma_k = \sigma_{k+1}\sigma_k\sigma_{k+1}$.

The set of all n-braids forms the *braid group*, and these two relations in fact generate the braid group. Thus the braid group on n strings, denoted $\mathbf{B}_n$, has presentation

$$\mathbf{B}_n = \left\langle \sigma_1, \ldots \sigma_{n-1} \;\middle|\; \begin{array}{ll} \sigma_i\sigma_j = \sigma_j\sigma_i \text{ iff } |i-j| \geq 2 & i,j \in \{1,\ldots,n-1\} \\ \sigma_k\sigma_{k+1}\sigma_k = \sigma_{k+1}\sigma_k\sigma_{k+1} & k \in \{1,\ldots,n-2\} \end{array} \right\rangle.$$

The proof that these two relations generate the braid group is quite involved and will not be given here. A proof can be found in Chapter 1, Section 3, of [5]. To each braid we can assign a permutation based on the order of the strings at the end of the braid. A braid is called a *pure braid* if its permutation is the identity. The identity of the braid group, the unbraid, is an example of a pure braid. The set of all pure braids on n strings, denoted $\mathbf{P}_n$, is a normal subgroup in $\mathbf{B}_n$ (see Chapter 1, Proposition 4.5 in [5]).

There is a simple but useful braid invariant for pure braids called the crossing number. If we number each string, then we can define $\mathrm{cr}(i,j)$ to be the number of times the i^{th} string passes behind the j^{th} string from below, minus the number of times the i^{th} string passes behind the j^{th} string from above. Though crossing number is a braid invariant for pure braids, it will be a very useful tool later in the chapter. An example of the crossing numbers of a braid is given in Figure 6.5.



$\mathrm{cr}(1,2) = 0$	$\mathrm{cr}(2,1) = -1$
$\mathrm{cr}(1,3) = 1$	$\mathrm{cr}(3,1) = 1$
$\mathrm{cr}(2,3) = 0$	$\mathrm{cr}(3,2) = 0$

Figure 6.5: An example of the crossing numbers.

6.2 Braids of Juggling Patterns

If we want to examine the braids of juggling patterns we have to set a standard for the number of hands and the throwing and catching locations of the balls. Siteswap notation does not distinguish this, and varying this will change the flight paths of the balls and possibly the braid. We will start with a simple one-hand model. Balls are caught at a fixed location and throws can be made from either side.

The best way to analyze this braid is to construct it from a profile braid. This works nicely because we can determine the over/under crossings straight from the stack sequence. There are two types of throws that determine whether strands cross over or under the others. Since the profile braid is determined by the stack sequence, we need to be able to denote throws from the back from throws from the front. We'll use α_i and ω_j to denote a throw from the back and a throw from the front, respectively. The subscript refers to the height of the throw in stack notation. Figure 6.6 is an example of a back throw and a front throw in a five-ball juggling pattern. Both throws in Figure 6.6 correspond to a 4 in the stack sequence.

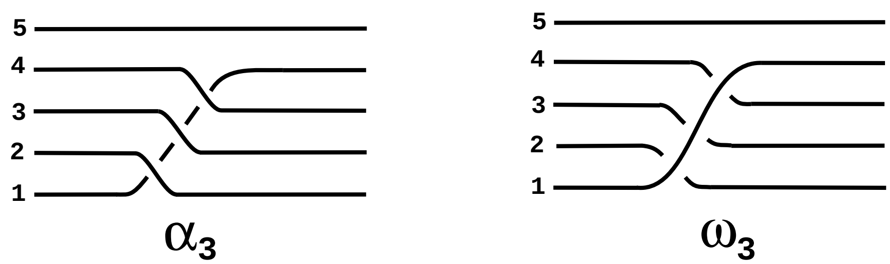


Figure 6.6: The two types of one-handed juggling throws.

Notice that α_i and ω_i can be expressed as

$$\alpha_i = \sigma_1\sigma_2\ldots\sigma_i$$
$$\omega_i = \sigma_1^{-1}\sigma_2^{-1}\ldots\sigma_i^{-1}.$$

We'll call the set of words generated by all jugglable braids with b balls M_b and M_b^+. Elements in M_b are those that can be expressed as words in the α_i's and elements in M_b^+ are those that can be expressed as words in the α_i's and ω_j's. M_b and M_b^+ are monoids. A monoid is a set and binary operation with all the properties of a group except the existence of inverses is not guaranteed. The set of natural numbers under addition is an example of a monoid.

Lemma 6.1 *For any non-trivial word $\beta_1 \in M_b$, there does not exists a word $\beta_2 \in M_b$ such that $\beta_1\beta_2 = 1$.*

Proof: Suppose there were two non-trivial words β_1 and β_2 such that $\beta_1\beta_2 = 1$. For any two strands b_1, b_2 in the braid $\beta_1\beta_2$, $\mathrm{cr}(b_1, b_2) = 0$. However, in M_b, strands only cross each other from below, so the crossing number between any pair of strands in a non-trivial word in M_b is positive. Thus $\operatorname{cr}(b_1, b_2) > 0$, so $\beta_1\beta_2$ is not the unbraid. $\square$

6.3 Counting Jugglable Braids

Counting braids is a delicate issue. Consider the patterns 42 and 24, which give rise to the braids $\alpha_1\alpha_2$ and $\alpha_2\alpha_1$, respectively. One typically juggles for more than just one cycle, in which case both of these patterns would be $\ldots 42424242424242 \ldots$, and would look exactly the same to an observer. We shall consider two braids β_1 and β_2 the same if β_1 and β_2 can be expressed as

$$\beta_1 = \sigma_{i_1}\sigma_{i_2}\ldots\sigma_{i_n}$$
$$\beta_2 = \sigma_{j_1}\sigma_{j_2}\ldots\sigma_{j_n}$$

such that for some integer k,

$$\beta_2 = \sigma_{i_k}\sigma_{i_{k+1}}\ldots\sigma_{i_n}\sigma_{i_1}\ldots\sigma_{i_{k-1}}$$

This simply means that the word expressing β_1 can be cyclically permuted into the word expressing β_2.

The next goal is to count the number of different braids that can arise from a length-n siteswap pattern with b balls. Recall that a b-ball pattern must have at least one b in its stack-sequence, and that 0's and 1's in the stack sequence have no effect on the braid. Because of this, the braids of the stack sequences 312, 321, and 32 all lead to the braid $\alpha_2\alpha_1$. Since any pattern of length less than n can be lengthened by adding 0's or 1's without changing the braid, we only need to count the number of stack sequences without 0's or 1's, containing at least one b, with period at most n. First of all, we'll consider just one type of throws, namely the α_i's.

Theorem 6.2 *The number of distinct braids in M_b arising from length-n juggling patterns* *is at most*

$$\sum_{k\le n} M'(k,b) = \sum_{k\le n} \frac{1}{k} \sum_{d|k} \mu(k/d) \left((b-1)^d - (b-2)^d\right).$$

Proof: Let $M'(n,b)$ be the number of stack sequences with period n with b balls that contain no 0's or 1's. Because there are $(b-1)^n$ stack sequences with at most b balls and $(b-2)^n$ sequences with at most $b-1$ balls, there are $(b-1)^n - (b-2)^n$ length-n stack sequences with exactly b balls. Each length-n sequence with period d, where d is a divisor of n, is equivalent to d different stack sequences (juggling patterns). Therefore, we can write

$$(b-1)^n - (b-2)^n = \sum_{d|n} dM'(d,b).$$

By Möbius inversion, we can solve for $M'(n,b)$ and get

$$M'(n,b) = \frac{1}{n} \sum_{d|n} \mu(n/d) \left[(b-1)^d - (b-2)^d\right].$$

For any stack sequence of length less than n, we can insert 0's and 1's anywhere in the sequence and not change the braid. Therefore, the number of distinct braids taken over all length-n juggling patterns is at most

$$\sum_{k\le n} M'(k,b) = \sum_{k\le n} \frac{1}{k} \sum_{d|k} \mu(k/d) \left((b-1)^d - (b-2)^d\right)$$

which proves the theorem. □

We stress "at most" in Theorem 6.2 because two braids arising from different juggling patterns may be the same braid. For example, consider the siteswap patterns 33 and 522. The stack sequence of these patterns are 33 and 232, respectively, which means that their braids are $\alpha_2\alpha_2$ and $\alpha_1\alpha_2\alpha_1$. Using the second braid relation we can conclude

$$\alpha_2\alpha_2 = \sigma_1\sigma_2\sigma_1\sigma_2$$

$$\alpha_1\alpha_2\alpha_1 = \sigma_1\sigma_1\sigma_2\sigma_1 = \sigma_2\sigma_1\sigma_2\sigma_1$$

which means that the patterns 33 and 522 have the equivalent braids when juggled in M_b. A juggler might appreciate this fact because when juggled with two hands, 33 (or 3) is called the "cascade," and pattern 522 is called the "slow cascade." Because in practice, most people treat 2's as just holds, 522 is just a higher and slower cascade, and it makes sense that they have the same braid. It is surprising that these two patterns also have the same braid when juggled with one hand. However, it is not true in general that if two siteswap patterns have the same braid with two hands, then they have the same braid with one hand. In fact, it seems that in most cases they do not have the same braid.

A simple corollary of Theorem 6.2 is an upper bound on the number of braids in M_b^+ arising from length-n juggling patterns.

Corollary 6.3 *The number of braids in M_b^+ arising from length-n juggling patterns is at most*

$$\sum_{k\leq n} M'(k,b) = \sum_{k\leq n} \frac{1}{k} \sum_{d|k} \mu(k/d) \left((b-1)^d - (b-2)^d\right) 2^d.$$

Proof: If both front and back throws are allowed as in M_b^+, then for each digit i in the stack sequence, there are two possible throws: α_{i-1} and ω_{i-1}. This means that each length-n pattern in M_b gives rise to at most 2^n possible braids in M_b^+. Thus, the number of braids corresponding with length-n patterns is at most

$$\sum_{k\leq n} M'(k,b) = \sum_{k\leq n} \frac{1}{k} \sum_{d|k} \mu(k/d) \left((b-1)^d - (b-2)^d\right) 2^d.$$

□

Appendix B contains some tables with the values of the upper bounds of the number of braids arising from length-n patterns in the monoids of b-ball juggling braids, M_b and M_b^+, for small values of b and n.

6.4 Determining Unbraids

A non-trivial unbraid is a word of at least one generator that is equivalent to the unbraid. A natural question that arises about the monoids M_b and M_b^+ is whether or not they contain any non-trivial unbraids. It is not difficult to show that M_b does not contain any non-trivial unbraids. Every element except the identity in M_b has at least one pair of strands (i, j) such that $\mathrm{cr}(i, j) > 0$. And it is impossible to get any pair of strands to have a negative crossing number. In order to get a non-trivial unbraid, the sum of the crossing numbers of every pair of stands must be zero. This is impossible using just words in the α_i's.

However, this argument does not work for M_b^+. Right away we see that $\alpha_1\omega_1$ is an unbraid, and we can concatenate this to itself to get an infinite family of unbraids. In fact, these are not the only unbraids in M_b^+. One such example, the braid $\omega_1\alpha_2\alpha_2\omega_2\omega_1$ in M_3^+, is shown in Figure 6.7.

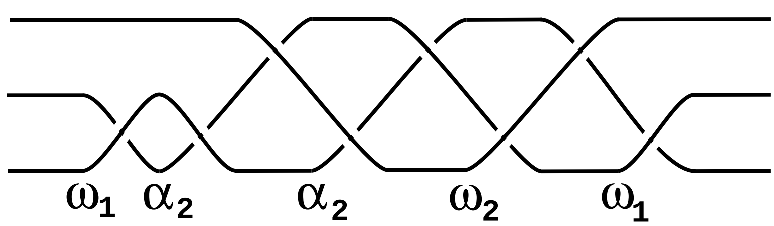
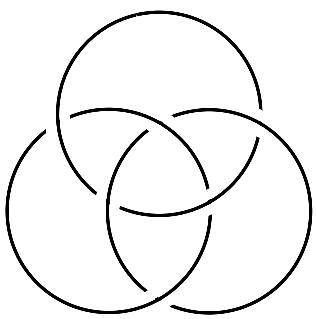


Figure 6.7: A non-trivial unbraided juggling pattern: $\omega_1\alpha_2\alpha_2\omega_2\omega_1$.

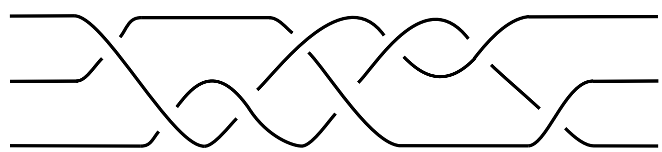
We wish to classify all such unbraids. We shall start by looking at M_3^+, three-ball patterns allowing both front and back throws. In the remainder of the chapter, a juggling pattern will be assumed to be with three balls unless otherwise stated. Since the crossing number of a pair of strings is a pure braid invariant, for any distinct pairs of strings i and j of an unbraid, $\mathrm{cr}(i, j) = 0$. However, there are braids that are not unbraids that have this property.

In knot theory, a *Brunnian link* is a collection of linked rings, called unknots, with the property that if one component is removed, the rest become unlinked. The most common example of a Brunnian link is called the *Borromean rings*, which is a link of three components. The Borromean rings are pictured in Figure 6.8 (a).

The ends of any braid can be identified to form a knot or a link (a knot with several components). The knot or link formed from identifying the ends of a braid is called the *closure* of that braid. If a Brunnian link is cut in the right place and stretched out, the resulting braid will have the property that removing any one string will leaving the remaining braid unbraided. Figure 6.8 (b) is a braid whose closure is the Borromean rings.



(a)



(b)

Figure 6.8: The Borromean rings, and a braid whose closure is the Borromean rings.

If a braid is unbraided, then for all distinct pairs of strands i and j, $\mathrm{cr}(i,j) = \mathrm{cr}(j,i) = 0$. However, the converse is not true. If $\mathrm{cr}(i,j) = \mathrm{cr}(j,i) = 0$ for every pair of strands i and j of a braid, then the closure of the braid might be a Brunnian link, or something more complicated. Upon inspection, it looks doubtful that there exists a 3-braid in M_3^+ whose closure is the Borromean rings. So we shall proceed with caution to find all three-ball jugglable unbraids by determining all braids in M_3^+ that have all six crossing numbers equal to zero.

Every 3-braid in M_3^+ is a product of α_i's and ω_i's, and each α_i or ω_i will change exactly i crossing numbers by ± 1. Suppose the balls are numbered #1,#2, and #3. An α_i crosses under the first i strings from below, so this increments each $\mathrm{cr}(1, j)$ by 1 for all $j \leq i$, assuming that the bottom ball is labeled ball #1. An ω_i crosses over the first i strings, so each of these strings crosses under the bottom string from above. Thus all crossing numbers $\mathrm{cr}(j, 1)$ are decremented by 1.

(a) Increments $\mathrm{cr}(1, 2)$ and $\mathrm{cr}(1, 3)$

(b) Decrements $\mathrm{cr}(2, 1)$ and $\mathrm{cr}(3, 1)$

Figure 6.9: How α_2 and ω_2 change the crossing numbers.

The crossing numbers that get changed are dependent not only on the type of throw, but also on the current permutation of the braid. An example is given in Figure 6.9. If the permutation of the balls from bottom to top is 123, and the next throw is an α_2, then ball 1 crosses behind the paths of ball #2 and ball #3. This increments $\mathrm{cr}(1, 2)$ and $\mathrm{cr}(1, 3)$. However, if the permutation of the balls had been 213, then ball #2 would have crossed behind the paths of ball #1 and ball #3. A subsequent α_2 would have instead incremented $\mathrm{cr}(2, 1)$ and $\mathrm{cr}(2, 3)$.

Table 6.1 shows how α_2 and ω_2 throws affect the crossing numbers of the braid given its current permutation. A "+" in an entry means that the crossing number in that column is incremented by one if the braid permutation is one of the two in that row. Likewise, the "–" means the crossing number is decremented by one. The α_1 and ω_1 throws are much simpler. Since such a throw simply switches the

Permutations	Throw	$\mathrm{cr}(1,2)$	$\mathrm{cr}(2,1)$	$\mathrm{cr}(1,3)$	$\mathrm{cr}(3,1)$	$\mathrm{cr}(2,3)$	$\mathrm{cr}(3,2)$
123, 132	α_2	+		+			
213, 231	α_2		+			+	
312, 321	α_2				+		+
123, 132	ω_2		–		–		
213, 231	ω_2	–					–
312, 321	ω_2			–		–	

Table 6.1: How α_2's and ω_2's affect crossing numbers.

bottom two balls, only two crossing numbers can be affected. If the braid permutation is ijk, then an α_1 will increment $\mathrm{cr}(i, j)$ and an ω_1 will decrement $\mathrm{cr}(j, i)$. In both cases, the resulting permutation is jik. In conclusion, there are exactly three pairs of crossing numbers that can be incremented by a single throw, and three pairs of crossing numbers that can be decremented by a single throw. Also, any of the crossing numbers can be decremented independently of the others given the correct type of throw and braid permutation.

The information in the table above can be encoded in a graph called a *stack graph*. The stack graph of a b-ball juggling pattern has $b!$ vertices – one for each braid permutation. There is a directed path from a vertex v_i to v_j if and only if it is possible to get from the permutation of v_i to the permutation of v_j by throwing an α_k or ω_k where $k < b$. Algebraically, this means that there is an element s of the symmetric group S_b of the form $(k\ \ k{-}1\ \ \ldots\ \ 2\ \ 1)$ such that $s : v_i \to v_j$. (In this chapter, we resort to the standard definition of the symmetric group S_n, the group of permutations of the set $\{1, \ldots, n\}$). For example, referring back to Figure 6.9, after ball #1 is thrown, the order of the balls changes from 123 to 231, and the permutation that does this is $(3\ \ 2\ \ 1)$. The stack graph of all three-ball patterns is shown in Figure 6.10

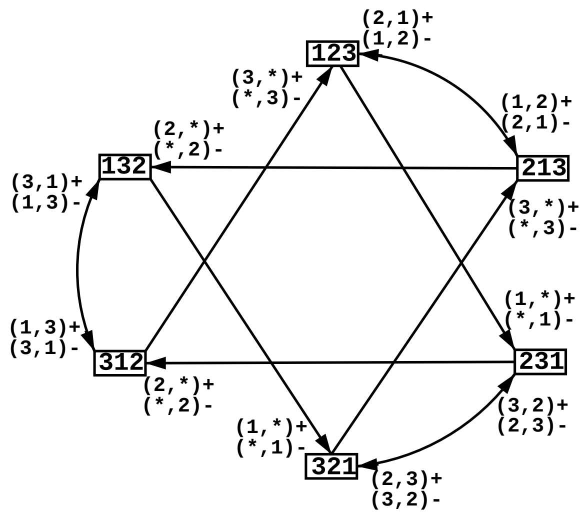


Figure 6.10: The stack graph for three-ball juggling patterns.

Each path in the stack graph has two labels which describe how the crossing numbers can change with each throw, as described in the table in Figure 6.1. For every edge traversed, we must choose whether the throw will be an α or an ω. For example, starting from the 123 vertex, there are two ways to get to 231: either throw an α_2 or an ω_2. The α_2 is denoted by $\mathrm{cr}(1,*)+$, which means that we increment $\mathrm{cr}(1,2)$ and $\mathrm{cr}(1,3)$. The "*" is a wild-card. Likewise, the ω_2 is denoted by $\mathrm{cr}(*,1)-$, which means that $\mathrm{cr}(2,1)$ and $\mathrm{cr}(3,1)$ are decremented.

Any one-handed three-ball juggling pattern can be represented as a walk on the stack graph. Moreover, pure braids have the nice property that they must be a cycle on the stack graph. This makes the task of classifying all unbraids easier. Readers familiar with the mathematics of juggling might notice a resemblance between a stack graph and the state graph, which describes when two siteswap patterns can be concatenated to form a new pattern. In both graphs, vertices represent some kind of state, and edges represent throws. Siteswap patterns correspond to closed loops on the state graph, whereas any path on the stack graph corresponds with a siteswap pattern. However, state graphs and stack graphs describe two completely different aspects of siteswap patterns. For a brief summary of state graphs, see Appendix A. A great source for learning all about state graphs is [9].

The stack graph displays a good deal of symmetry. There are two types of edges: each vertex has one "long" edge, corresponding with an α_2 or ω_2, going into it and one going out of it. Also, each vertex has one "short" edge, corresponding with an α_1 or ω_1, going into it and one short edge leaving. Next we will present several ways to set up a system of equations whose solutions will describe all unbraids.

6.4.1 *Setting the crossing numbers to zero.*

Without loss of generality, assume that any three ball juggling pattern begins with the permutation 123. If we keep a running total of the sum of all six crossing numbers, then unbraids will be cycles such that all six crossing numbers are zero. There are six pairs of crossing numbers that can be changed with a single throw, as well as all six individual crossing numbers that can be changed independently. Thus there are twelve possible non-empty subsets of

$$\{\,\mathrm{cr}(1,2),\ \mathrm{cr}(2,1),\ \mathrm{cr}(1,3),\ \mathrm{cr}(3,1),\ \mathrm{cr}(2,3),\ \mathrm{cr}(3,2)\}$$

that can be changed by a single throw. An unbraid has the restriction that each of the crossing numbers is zero. This gives us a system of six equations on twelve variables, which we can represent by the following matrix:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

Each row of the matrix represents a crossing number, and each column in the matrix represents a way to change the crossing numbers. Observe that the first six columns in (6.1) are the six rows in Table 6.1. Elements in the nullspace of C describe ways to traverse edges in the stack graph so that the sum of each crossing number is zero. However, it is important to notice that such an element might not necessarily be a closed path, which means that it physically cannot be juggled. The nullspace of C is six-dimensional, with basis $\{X_1, X_2, X_3, X_4, X_5, X_6\} =$

$$\left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{6.2}$$

Pictorially, these six elements may be realized on the stack graph as in Figure 6.11.

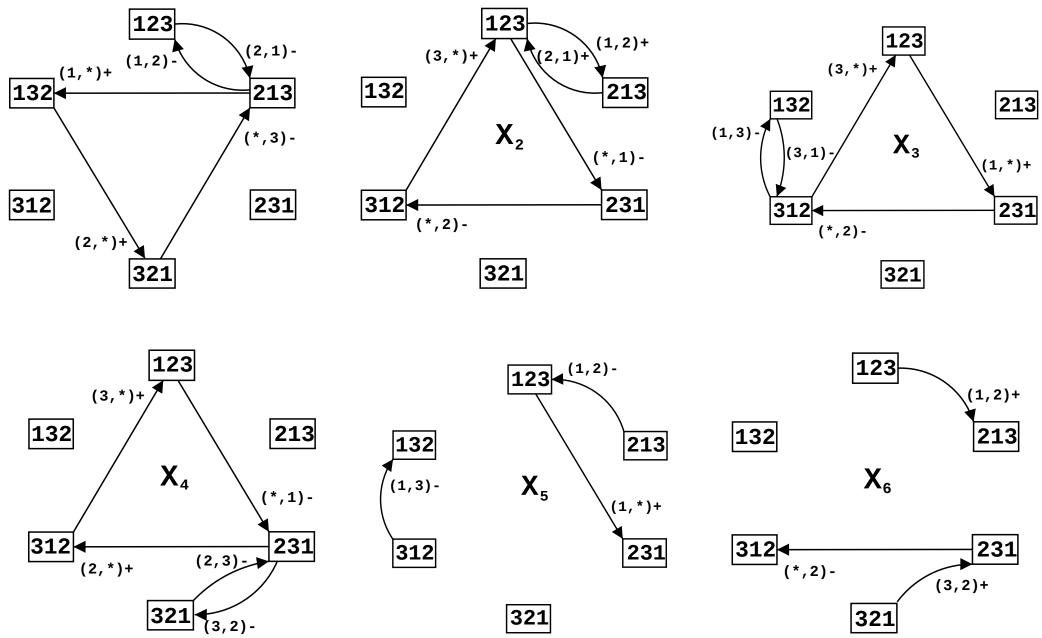


Figure 6.11: Realizations of the basis for all unbraids on the stack graph.

However, it is important to understand that each basis element has several valid realizations on the stack graph. Notice that as the stack graph is drawn in Figure 6.10, the parallel lines have the same effect on the crossing numbers. There is a subtle difference between unbraids, walks on the stack graph, and elements in the nullspace of C.

Definition 6.1 *An unbraid class is twelve-dimensional vector $\mathcal{U}$ that can be expressed as a linear combination of X_i's such that the first six entries of $\mathcal{U}$ are non-negative.*

Every unbraid class is an element of the nullspace of C. Unbraid classes correspond to ways of selecting weighted edges from the stack graph so that all six crossing numbers sum to zero. One question that arises is whether or not any unbraid classes contain a braid whose closure is a Brunnian link, or some other non-trivial braid. In other words, do all closed paths on the stack graph correspond to unbraids? Upon inspection, it looks like this statement is likely true, because the braid in 6.8(b) does not appear to be jugglable, and because X_1, X_2, X_3, and X_4 are all unbraids. However, we cannot rule out the possibility.

It is important to understand that unbraid classes do not specify the order that the edges are traversed, so an unbraid class can correspond to many unbraids (or possibly, braids with Brunnian link closures). For example, suppose that $\mathcal{U} = X_1 + X_2 + X_3$. Starting at vertex 123 in the stack graph, one possible unbraid of $\mathcal{U}$ is to traverse X_1, X_2, and X_3 in that order as shown in Figure 6.11. Another possibility is X_1, X_3, X_2. Still, there are more complicated ways. Notice that there are two potential starting directions for X_2 as it is depicted in Figure 6.11 when starting at vertex 123. It is even possible to insert one of the X_i's before finishing traversing another. For example, traverse X_2 and upon reaching the 312 vertex, before completing the cycle, start traversing X_3, but upon reaching the 123 vertex, traverse X_1. Then finish X_3, and then finish X_2. These are all realizations of the unbraid class $X_1 + X_2 + X_3$. There are also realizations of unbraid classes that do not correspond with paths on the stack graph.

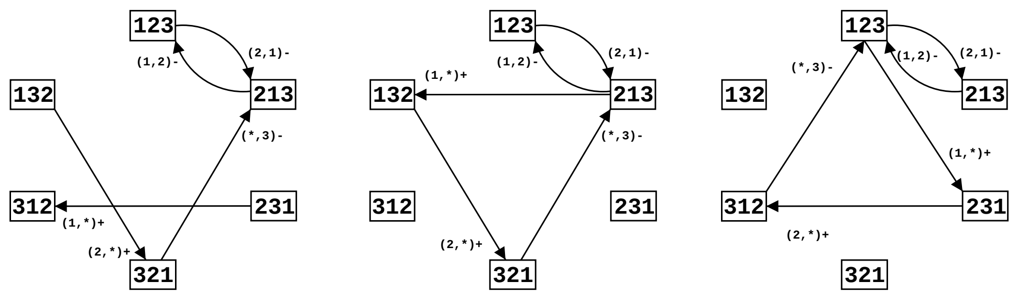
Definition 6.1 *A walk of an unbraid class $\mathcal{U}$ is a path on the stack graph that is a realization of $\mathcal{U}$.*

A walk of an unbraid class is a cycle on the stack graph. Every walk gives rise to precisely one braid.

Definition 6.2 *A fragment of an unbraid class $\mathcal{U}$ is a realization of $\mathcal{U}$ that is not a walk.*

A walk is not a fragment and a fragment is not a walk. Moreover, every realization of an unbraid class on the stack graph is either a walk or a fragment. A walk can be juggled but a fragment cannot. Figure 6.12 shows X_1 realized three different ways. The first one is a fragment and the last two are walks.

The basis element X_1 can be realized as the braid $\omega_1\alpha_2\alpha_2\omega_2\omega_1$, which is the unbraid shown in Figure 6.7. Every realization of X_5 and X_6 are fragments because



(a) Fragment (b) Walk (c) Another walk

Figure 6.12: Different realizations of the basis element X_1.

neither can be traversed as a connected path on the stack graph. However, it is not clear if there are any complicated linear combinations of fragments that can be put together with X_5 or X_6 to form a cycle. This systems of twelve equations on six variables is not the best way to determine all unbraids. Solutions to the system, such as X_5 and X_6, may not be able to be realized as a walk, and thus cannot be juggled. Any unbraid on the stack graph must have the property that the in-degree of any vertex equals its out-degree. Equation (6.1) does not guarantee this. The problem lies in the fact that for each possible pair of crossing numbers that can be incremented or decremented together, there are two different edges that can do this. In addition, each edge can correspond to two different variables in (6.1), because each edge has a positive and negative label that correspond with α and ω throws, respectively. Each of the twelve edges has two different labels, so a throw in a juggling pattern can correspond to moving on the stack graph one of twenty-four possible ways.

6.4.2 *The complete system of equations*

Suppose we label the edges of the stack graph as shown in Figure 6.13. The short double edges are actually two edges, and are labeled as such. For example, e_1 is the edge from vertex 123 to vertex 213, while e_2 is the edge from vertex 213 to vertex 123.

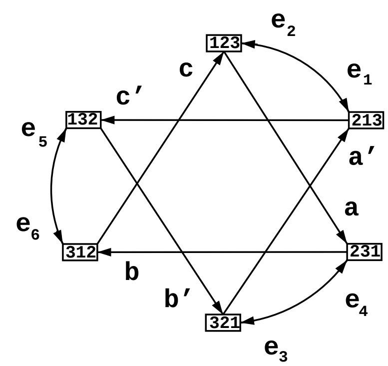


Figure 6.13: Labeling the edges of the stack graph.

However, we want to be able to distinguish between α and ω throws. For each edge labeled i, let i_+ represent traversing that edge by an α and i_- represent traversing that edge by an ω. For example, the a_+ edge is the edge from vertex 123 to vertex 231 where $\mathrm{cr}(1,2)$ and $\mathrm{cr}(1,3)$ are incremented. On the other hand, a_- is that same edge, only $\mathrm{cr}(2,1)$ and $\mathrm{cr}(3,1)$ are decremented. Recall that the twelve columns in the matrix in (6.2) represented the twelve ways two change the crossing numbers. Using this same notation, a braid represented by a path on the stack graph will have all six crossing numbers zero only if it satisfies (6.3).

$$
\begin{aligned}
&(e_{1+} - e_{2-})\begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix} - (e_{2+} - e_{1-})\begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix} + (e_{3+} - e_{4-})\begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix} - (e_{4+} - e_{3-})\begin{bmatrix}0\\0\\0\\1\\0\\0\end{bmatrix} + (e_{5+} - e_{6-})\begin{bmatrix}0\\0\\0\\0\\1\\0\end{bmatrix} - (e_{6+} - e_{5-})\begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix} \\
&+ (a_+ + b'_+)\begin{bmatrix}1\\0\\1\\0\\0\\0\end{bmatrix} - (a_- + b'_-)\begin{bmatrix}0\\1\\0\\1\\0\\0\end{bmatrix} + (b_+ + c'_+)\begin{bmatrix}0\\1\\0\\0\\1\\0\end{bmatrix} - (b_- + c'_-)\begin{bmatrix}1\\0\\0\\0\\0\\1\end{bmatrix} + (c_+ + a'_+)\begin{bmatrix}0\\0\\0\\1\\0\\1\end{bmatrix} - (c_- + a'_-)\begin{bmatrix}0\\0\\1\\0\\1\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\\0\\0\\0\end{bmatrix}
\end{aligned}
\tag{6.3}
$$

In addition to setting the crossing numbers equal to zero like we did in (6.1), we can eliminate a lot of solutions that do not correspond to juggling patterns by ensuring that the out-degree of each vertex equals its in-degree. This yields the following six equations:

$$
\begin{aligned}
\mathbf{123}: &\quad (e_{1+} + e_{1-}) + (a_+ + a_-) = (e_{2+} + e_{2-}) + (c_+ + c_-) \\
\mathbf{213}: &\quad (e_{2+} + e_{2-}) + (c'_+ + c'_-) = (e_{1+} + e_{1-}) + (a'_+ + a'_-) \\
\mathbf{231}: &\quad (e_{3+} + e_{3-}) + (b_+ + b_-) = (e_{4+} + e_{4-}) + (a_+ + a_-) \\
\mathbf{321}: &\quad (e_{4+} + e_{4-}) + (a'_+ + a'_-) = (e_{3+} + e_{3-}) + (b'_+ + b'_-) \\
\mathbf{132}: &\quad (e_{5+} + e_{5-}) + (b_+ + b_-) = (e_{6+} + e_{6-}) + (c'_+ + c'_-) \\
\mathbf{312}: &\quad (e_{6+} + e_{6-}) + (c_+ + c_-) = (e_{5+} + e_{5-}) + (b_+ + b_-)
\end{aligned}
\tag{6.4}
$$

Together, (6.3) with (6.4) gives us the following 12×24 matrix:

$$
\begin{bmatrix}
1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
0 & -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1
\end{bmatrix}
\tag{6.5}
$$

The nullspace of (6.5) is thirteen-dimensional, and vectors in the nullspace describe every possible unbraid that is a cycle. However, since traversing an edge a negative number of times has no physical meaning, a vector in the nullspace can only be physically realized as an unbraid if all of its entries are non-negative. It is inconvenient to have a basis of unbraids consisting of thirteen 24×1 vectors, most of which are not even realizable juggling patterns.

6.4.3 *Simplifying the equations*

Notice that in all six equations in (6.3) as well as in (6.4), e_{1+} appears on one side of the equation if and only if e_{2-} is on the other side. Likewise, e_{2+} and e_{1-} also come in pairs. In fact, each of the e_{i+} has a corresponding e_{j-}. The following six variables can be substituted into (6.3) and (6.4):

$$\begin{aligned}
E_1 &= e_{1+} - e_{2-} \\
E_2 &= e_{2+} - e_{1-} \\
E_3 &= e_{3+} - e_{4-} \\
E_4 &= e_{4+} - e_{3-} \\
E_5 &= e_{5+} - e_{6-} \\
E_6 &= e_{6+} - e_{5-}
\end{aligned} \tag{6.6}$$

This simplification leads to a 12×18 matrix, where the eighteen columns represent the following eighteen variables, respectively:

$$a_+, a_-, b_+, b_-, c_+, c_-, a'_+, a'_-, b'_+, b'_-, c'_+, c'_-, E_1, E_2, E_3, E_4, E_5, E_6.$$

This 12×18 matrix is given in (6.7).

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{6.7}$$

Even though (6.7) has six fewer variables than (6.5), the dimension of the nullspace is only reduced by three. However, the nullspace has a basis with the nice property that the last six non-zero entries of the vectors can be either positive or negative as still be realized on the stack graph. A basis for the nullspace is given in (6.8).

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\} \tag{6.8}$$

The basis in (6.8) has several advantages over the basis in (6.2). First of all, each basis vector in (6.8) is a cycle on the stack graph. We no longer have to worry about fragments because each entry in the vectors in (6.8) refers to a specific way to traverse a specific edge on the stack graph. This new basis also shows that it is indeed possible to have a linear combination of the two fragments, X_5 and X_6 from (6.8), with the other X_i's, and get a juggable pattern. The unbraid in Figure 6.14 is an example.

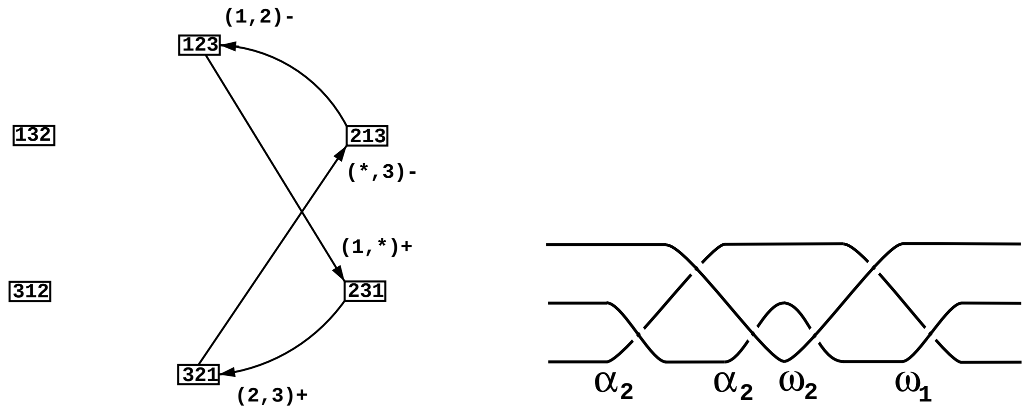


Figure 6.14: A length-four unbraid.

Any linear combination of basis vectors in (6.8) can be juggled if the first twelve entries are non-negative. However, even though there is only one realization for each linear combination of basis elements on the stack graph, there may be several different possible orders to traverse the edges. In the original basis, two of the vectors were fragments and could not be concatenated with other vectors at will. However, in both bases, there are unbraids that cannot be represented as positive linear combinations of the basis vectors. It would be nice to find a smallest set of vectors, if such a finite set exists, in both (6.2) and (6.8) such that any unbraid can be represented as a positive linear combination of basis vectors. The existence of such a set would immediately answer whether or not there exists any non-trivial jugglable braids with all crossing numbers equal to zero.

Another natural step is to examine not just unbraids, but also look into when two three-ball juggling patterns yield the same braid, and when two paths on the stack graph correspond with the same braid. Eventually, it would be interesting to look at stack graphs of patterns of more than three-balls and see if similar results hold. Proof techniques must be generalized, because the size of the stack graphs grow large very quickly. There are $b!$ vertices in the b-ball stack graph.

6.5 Adding More Balls

Thus far, we have not examined the stack graphs of patterns with more than three balls because the size of the graph grows very quickly. A good way to understand a larger stack graph is to collapse it into a smaller graph. For any stack graph, identify two vertices if they have the same top ball in their permutation, and remove all singleton edges. This new graph is called the *condensed graph*.

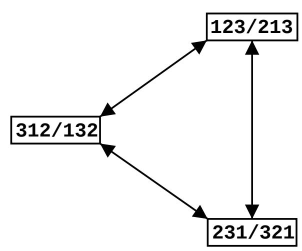
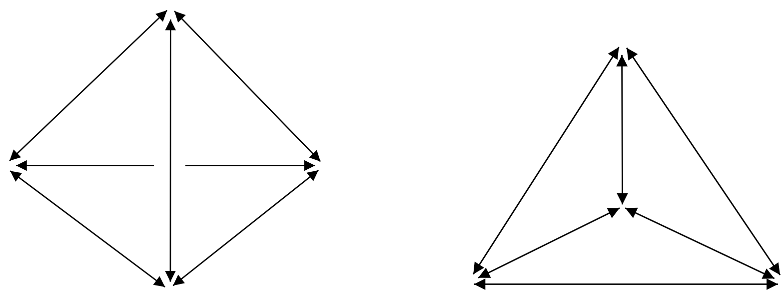


Figure 6.15: The condensed three-ball stack graph.

As an example, consider the three-ball stack graph. Vertices 123 and 213 become one vertex, 231 and 321 become another, and 132 and 312 become the third. One way to think about the three-ball condensed graph is as the three-ball stack graph modulo the two-ball stack graph, which is just a double-edge between two vertices. Figure 6.15 shows the three-ball condensed graph. The four-ball condensed graph has just four vertices is shown in Figure 6.16.

One can think of the four-ball condensed graph as a 3-simplex, or as the complete graph on four vertices, K_4. The four-ball stack graph can be very messy when drawn in the plane. The four-ball stack graph has four three-ball stack graphs as subgraphs, one at each vertex in the condensed graph. The four-ball stack graph is shown in Figure 6.17.

In the three-ball stack graph, there were two types of edges. The short edges corresponded with α_1 and ω_1 throws, and the long edges corresponded with α_2



(a) 3-simplex

(b) K_4

Figure 6.16: Two ways to view the four-ball condensed graph.

and ω_2 throws. In the four-ball stack graph, there is a third type of edge, which corresponds with α_3's and ω_3's. Each vertex has one of each type of edge going in and one type of edge leaving.

6.6 Interesting Questions

1. For a given unbraid class $\mathcal{U}$, how many different braids can be realized as walks of $\mathcal{U}$?

2. For a given linear combination of vectors in (6.8), how many different braids can be realized as walks? In other words, how many different ways are there to traverse those edges that lead to different braids?

3. Recall that every braid can be closed into a knot or link (a knot with multiple components). What can be said about the closure of the braids in M_b and M_b^+? Which knots are achievable? It is known that any link can be achieved by the closure of some braid.

4. Does there exist a jugglable braid whose closure is a Brunnian link?

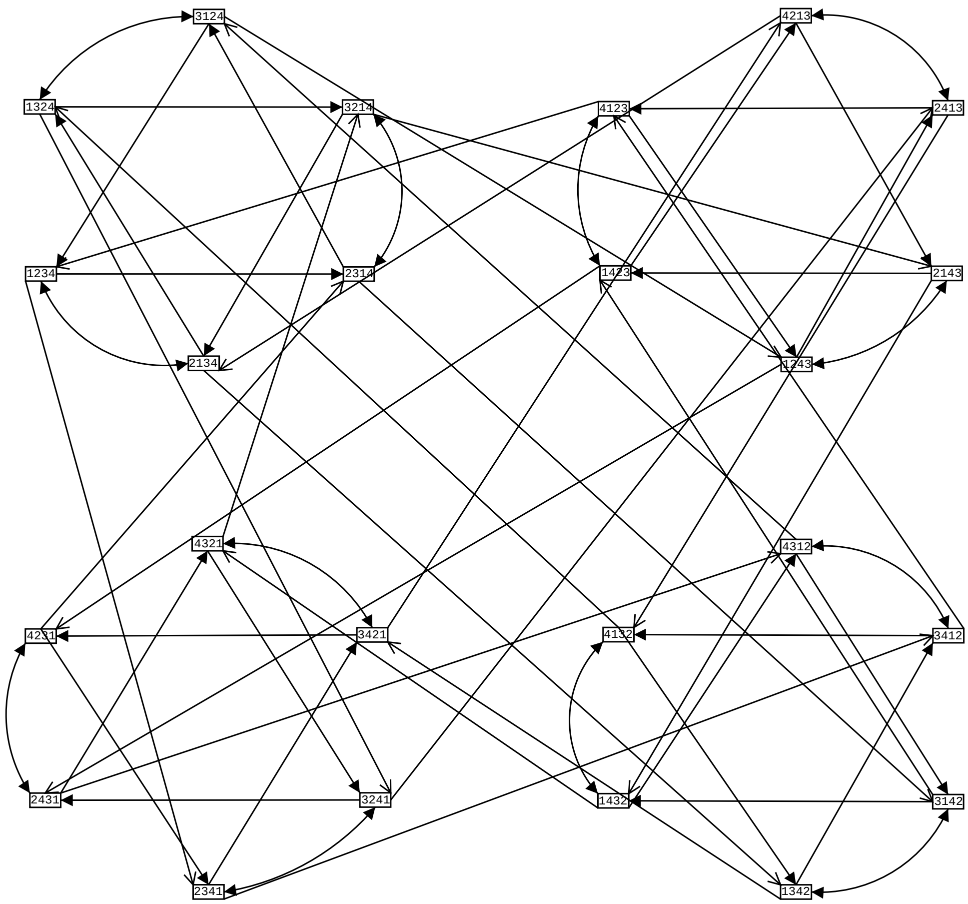


Figure 6.17: The stack graph of four-ball juggling patterns.

5. We can generate more braids if we allow bounced throws, which look like regular parabolas in the profile braids. What braids can be generated with bounced throws?

6. Generalize stack notation and the algebras M_b and M_b^+ to multiple-hand juggling where each hand has its own stack.

7. Which elements of $\mathbf{B}_n$ are realizable two-handed siteswap patterns? Can we determine all possible braids if we fix the length of the pattern? Can some of the results about M_b^+ and stack graphs be generalized to bounced juggling

patterns?

8. We call two n-ball juggling patterns *homotopic* if they yield the same braid in $\mathbf{B}_n$. For a given n, how many distinct juggling patterns are there up to homotopy? Recall that we have an upper bound for this number, not an exact value.

9. Define the *writhe* of a braid to be the sum of the exponents of the σ_i's. The writhe basically measures how much the braid is twisting. If S is a two-handed siteswap pattern of odd period, then $wr(\beta(S)) = 0$. If $wr(\beta(S)) = 0$ for some two-handed siteswap pattern S, then must S have odd period?

Appendix A

Appendix

A.1 State Graphs

Suppose we are juggling a siteswap pattern, and want to know which throws, no higher than a certain digit, can be thrown next beat. State graphs can answer this. Set a maximum throw height, m. At any point while juggling, we look at the next m beats and write a 0 if no ball will land and a 1 if a ball will land. For example, suppose one wants to know what can be thrown after the 1 in the four-ball pattern 561, and because of a low ceiling, the highest throw must be no more than a 7. In this case, consider the next seven beats.

$$\ldots 561561561561\;\underline{x}\,\underline{x}\,\underline{x}\,\underline{\ }\,\underline{x}\,\underline{\ }\,\underline{\ }$$

The x's denote beats when the four balls will land. We write this as

$$1110100$$

The first digit of this sequence represents the next throw. Because two balls can't land on the same beat, the next throw cannot be a 1, 2, or 4. However, a 3, 5, 6, or 7 will work. Suppose we throw a 3. Then our pattern becomes

$$\ldots 5615615615613\;\underline{x}\,\underline{x}\,\underline{x}\,\underline{x}\,\underline{\ }\,\underline{\ }\,\underline{\ }.$$

which gives rise to the binary string

$$1111000$$

In the example with four balls and maximum height 7, there are 35 possible binary strings. We can construct a graph where each vertex represents a legal binary string, and there is a directed edge from vertex v_i to v_j if and only if it is possible to go from the state v_i to v_j by a single throw. That edge is labeled with the height of the throw required to go from v_i to v_j. In the above example, there would be a directed edge labeled with a 6, from the 1110100 vertex to the 1111000 vertex. Such a graph is called a *state graph*. State graphs are discussed extensively in [9]. They have the nice property that any possible siteswap pattern given constraints of the number of balls and the maximum throw height, corresponds to some path on the state graph. Conversely, for any path of the state graph, the string the digits of the edges is a siteswap pattern. To give an example, the state graph of two balls with a maximum height 4, denoted $S(2,4)$, is shown below.

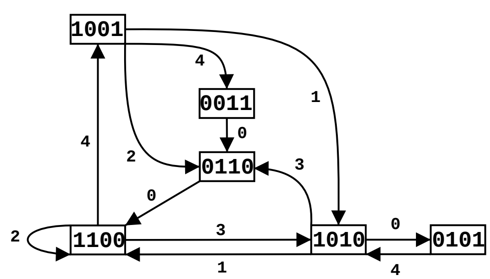


Figure A.1: The state graph $S(2,4)$.

State graphs resemble stack graphs because in both graphs, the vertices represent certain "states" and directed edges represent all possible throws. However, as shown, the two are very different and describe completely different aspects of siteswap patterns.

(b,n)	1	2	3	4	5	6	7	8	9	10
2	1	1	1	1	1	1	1	1	1	1
3	1	2	4	7	13	22	40	70	126	225
4	1	3	9	24	66	173	467	1247	3375	9156
5	1	4	16	58	214	768	2796	10146	37082	135956
6	1	5	25	115	535	2445	11265	51855	239735	1111229

Table A.1: Equation (A.1) evaluated for small b and n

A.2 Tables of Sequences

As described in Chapter 6, an upperbound for the number of different braids that arise from b-ball juggling patterns of length n, where throws are only made from one side of the pattern (ω's are not allowed) is given by the formula

$$\sum_{k=1}^{n} \frac{1}{k} \sum_{d|k} \mu(k/d) \left((b-1)^d - (b-2)^d\right) \tag{A.1}$$

Table A.1 shows (A.1) evaluated for small of b and n. One might notice that when $n = 3$, then this formula seems to equal $(b-1)^2$. In fact, it does, but not because of a neat identity. Recall that $\mu(1) = 1$ and $\mu(2) = \mu(3) = -1$. When $n = 3$, (A.1) evaluated at $b-1$ becomes

$$[b-(b-1)]+\frac{1}{2}\left[-(b-(b-1))+(b^2-(b-1)^2)\right]+\frac{1}{3}\left[-(b-(b-1))+(b^3-(b-1)^3)\right]$$

and this is just b^2 when simplifed.

If throws can be made from both sides of the pattern (α's and ω's), then an upper bound for the number of braids is

(b,n)	1	2	3	4	5	6	7	8	9
2	2	3	5	8	14	23	41	71	127
3	2	7	25	82	280	941	3263	11393	40457
4	2	11	61	316	1666	8731	46381	248011	1338611
5	2	15	113	806	5804	41665	301267	2188117	15981421

Table A.2: Equation (A.2) evaluated for small b and n

$$\sum_{k=1}^{n} \frac{1}{k} \sum_{d|k} \mu(k/d) \left((b-1)^d - (b-2)^d\right) 2^d \tag{A.2}$$

Table A.2 shows (A.1) evaluated for small values of b and n.

Bibliography

[1] C. Adams. *The Knot Book*. W.H.Freeman, 1994.

[2] B. Tiemann B. Magnusson. The physics of juggling. *Physics Teacher*, 27:584–589, 1989.

[3] J. Buhler, D. Eisenbud, R. Graham, and C. Wright. Juggling drops and descents. *American Mathematical Monthly*, 101(6):507–519, 1994.

[4] The juggling information service. Available online at `http://www.juggling.org`.

[5] B.I. Kurpita K. Murasugi. *A Study of Braids*. Kluwer Academic Publishers, 1999.

[6] J. Kalvin. *Optimal juggling*. Published online at `http://www.juggling.org/papers`, 1996.

[7] J. Kalvin. *The human limits*. Published online at `http://www.juggling.org/papers`, 1997.

[8] D. Margalit, N. Picciotto, and S. Babineau J. Llobrera. *Topology and juggling*. Published online at `http://www.brown.edu/Students/OHJC/topology/index.html`, 2002.

[9] B. Polster. *The Mathematics of Juggling*. Springer-Verlag, 2003.

[10] M.A. Readdy R. Ehrenborg. Juggling and applications to q-analogues. *Algebraic Combinatorics in Discrete Mathematics*, 157:107–125, 1996.

[11] `rec.juggling` newsgroup. The archives of the newsgroup can be found at `http://www.juggling.org`.

[12] N.J.A. Sloane. The on-line encyclopedia of integer sequences. Available online at `http://www.research.att.com/njas/sequences/`, 2002.

[13] R.P. Stanley. *Enumerative Combinatorics*. Cambridge University Press, 2001.