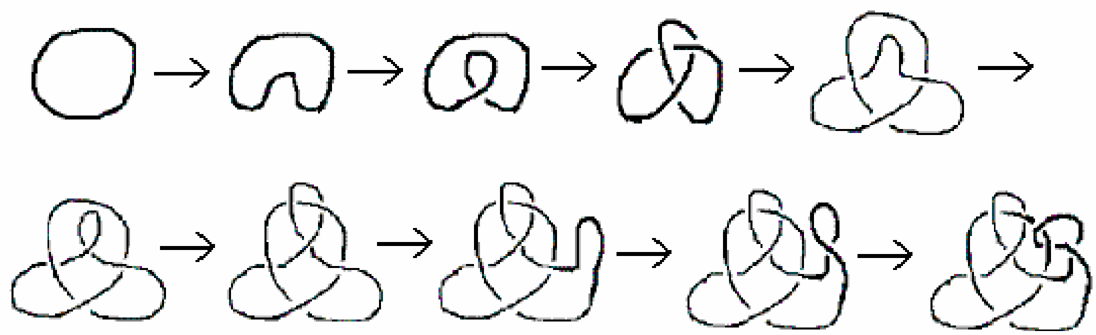
Undergraduate Research Assistantship Final Report

An Exploration of Nontraditional Conceptualizations of Knots:

Minimum Curvature and Knot Construction via Self-Entanglement

By: J.W. (Dub) Justin William Gaberdiel, email: euclidub@hotmail.com



"The Construction of Prime Knot 9.27 via Self-Entanglement"

TABLE OF CONTENTS

1: PREFACE TO THE READER

Herein I assume the reader has at least an introductory level of understanding of knot theory. I will make reference to various basic ideas -- such as Dowker's and Conway's notation -- without explanatory notes. There are a vast number of high quality introductory knot theory books on the market, and there is no need to give another overview here. However, to protect against ambiguity, I will, of course, define all of the concepts that I used in my actual research. If you are unfamiliar with knot theory, I recommend reading "The Knot Book" by Colin C. Adams before proceeding. Also, while reading this report, it would be helpful to have a copy of a prime knot table (located in the back of any knot theory text) on hand.

Throughout this report I use the word "loop" as a shorthand definition of a "closed, nonintersecting path through three-dimensional space" -- the mathematical definition of a knot.

2: PRELIMINARY NOTE

In addition to the time I spent performing my own research, I also spent a good deal of time reading any and all books I could find on knot theory and related topics. In my readings, I learned that most of the current research in the field of knot theory is beyond my present skill level, requiring in-depth knowledge of group theory, topology, quantum gravity, etc. -- all subjects that I have not yet studied and in which I could not possibly become fluent during the course of a one semester research project. Thus, I must emphasize that there is an excellent possibility that my investigations could be complemented or enhanced by other fields of study.

3: ABSTRACT

This report is a summary of my study of and experimentation with knots between June and December of 2002, under the advising of University of Arizona faculty Dr. Maria Robinson. First, I will provide a description of the train of thought I followed that motivated my exploration of nontraditional conceptualizations of knots. Then, I will review my two primary investigations. One investigation was to find a way to calculate the minimum curvature of knots, in hopes that distinct knots would have distinct values of minimum curvature. If this were true, any random projection of a given knot would yield the same value of minimum curvature as any other random projection of the same knot. Such a calculation technique would aid in the task of distinguishing between distinct knots. The other investigation, which was my primary focus, was to devise a new knot construction technique in which knots are conceived of as self-entangled loops, differing only in the degree of complexity of self-entanglement. Thus, rather than constructing knots by connecting "crossings" or "tangles" together, as in Dowker's and Conway's notation, I attempted to define a construction technique in which complex

knots are derived from simpler knots in a recursive process of self-entanglement, using the unknot as the base case.

While I did make some promising findings, much of my research is still open due to the fact that one semester was insufficient time to fully explore all of the avenues I desired to study. Throughout this report, I make note of unanswered questions. If any reader should be so inclined, I would be delighted to receive suggestions or feedback via email.

4: MY INITIAL RESEARCH PROPOSAL: FOUR KNOT CONSTRUCTION TECHNIQUES

Within knot theory, there are plenty of knot invariants that you can calculate in order to distinguish between distinct knots. However, none of these invariants actually define what a knot *is* -- they're simply clever gimmicks that allow you to distinguish certain knots from certain other knots. After all, no one single knot invariant has yet to be defined that can distinguish between all distinct knots. The underlying purpose of my research -- the reason I was motivated to study knot theory in the first place -- was to find a knot invariant that actually described what a knot *is*, providing a description of the *essential property* of a knot, by which distinct knots differ from each other. For instance, I once read "the goal of mathematics is not to deal with all the properties of the various objects, but through a logical subtraction to define and study the properties that set them apart" (Aneziris, pg 18). While this certainly seems to be the way knot theory is traditionally performed, I wanted to actually address the properties of the knot itself, not just the properties that set knots apart. This was my goal.

To obtain this goal, I proposed to study knot construction techniques for my research project. I figured that if I knew how to construct a knot, then I would likely also know what the essential property of a knot is; namely, because I just *constructed* it.

I solve mathematical problems best by studying data and looking for patterns. So, I began my research by simply studying a table of prime knots. Just as links are two or more distinct loops that are entangled with each other in a particular way, when I looked at knots, I saw loops that were *self*-entangled in a particular way. Based upon this conception of knots -- self-entangled loops -- I began devising my own knot construction technique whereby I was looking for a way to construct all prime knots by methodically and algorithmically entangling a loop with itself. (I will discuss this construction technique in detail in section 8 of this report.)

At the same time, I was also studying and experimenting with other knot construction techniques that were similar in kind to the knot construction techniques I had read about in knot theory books. Specifically, I was investigating the following three methods of knot construction:

1) Using graph theory, I planned to encode the details of an 'n' crossing knot into a graph with 'n' vertices. In this way, I hoped to reduce the task of constructing knots to the task of constructing graphs.

2) Using combinatorial theory, I planned to treat knots of 'n' crossings as a special subset of the set of all possible connections between the endpoints of 'n' "X"'s. In this way, I

would simply list all possible knots of 'n' crossings, eliminate the duplicates, and list the remaining distinct knots. This technique is similar to Dowker's notation, except that my "X"'s would be independent, whereas Dowker's notation arranges the "X"'s in a connected row.

3) As a variant of the combinatorial approach to connecting "X"s, I would model knot construction by imagining a group of 'h' humans (facing each other in a circular formation) who hold hands in various combinations. The knot, in this case, is the string of connected arms. (This approach was motivated by a game mentioned on page 278 in Colin C. Adams' The Knot Book.)

These three techniques, in addition to my technique of knot construction via the self-entanglement of loops, were the starting point for my research.

5: DISSATISFACTION WITH THE VISUAL PERSPECTIVE OF KNOTS

However, after performing a literature review on the topic of knot construction, and working with the three more-conventional construction techniques outlined above, I quickly became dissatisfied with the traditional approach to studying knots, i.e.: by observing the properties of the *crossings* of knots in their two dimensional planar diagrams. More specifically, I became dissatisfied with the study of knots from a visual perspective. Given that knots are topological objects that can be deformed and rotated in space, it seemed like a red herring to me to study knots in terms of a property -- crossings -- that changes every time you deform or rotate the knot. Besides: a closed, nonintersecting curve in three-dimensional space -- the true definition of a knot -- does not *technically possess* "crossings"; crossings are simply an accident of the visual perspective.

Thus, I was compelled to take a new approach to studying knots, motivated by the question, "How would a *sightless* race of intelligent beings study knots?" Humans are very visual creatures, and a sightless intelligence would likely approach the study of knots in a completely different way. (Never mind whether or not a race of sightless intelligent beings actually exists in the universe.) This question launched me into an extensive brainstorming process to find new ways to conceptualize knots. Specifically, I wanted to find a holistic conception of knots that was independent of any particular sensory modality, such as vision. I emphasized a holistic approach because any reductionistic approach -- such as studying only the crossing number or only the writhe number of a knot -- focuses on only part of the whole knot, which will likely result in the omission of critical characteristics from analysis.

6: DISSATISFACTION WITH TRADITIONAL CONSTRUCTION TECHNIQUES

Given that traditional knot construction techniques focus on constructing knots out of "crossings" -- a concept I wanted to try to stay away from -- I was also uninterested in pursuing further the three more-conventional construction techniques I initially proposed to study. The following paragraphs explain the new direction I followed in my pursuit of a new knot construction technique.

The basic task of knot theory is to determine if two given knots are topologically different or the same. For example: given knot1 and knot2, if it is not possible to deform either knot1 to look like knot2, or visa versa, then we can safely say that knot1 and knot2 are distinct. Likewise, if you *can* deform knot1 to look like knot2, then they are the same knot. In recent decades, mathematicians have come up with plenty of clever methods to determine if two knots are different or the same, but these methods sidestep a critical question: if two knots are different, *WHAT* about them is different? What is the *essence* of a knot, by which distinct knots differ from each other?

For instance, in the study of two-dimensional surfaces, it is known that every surface is constructed from some connected sum of tori and projective planes. If one were to ask, "what is the essential property of a surface?", I would respond that every surface is a connected sum of tori and projective planes, and that the essence of one particular surface is simply the number of its constituent parts: 2 tori and 1 projective plane, for example. Thus, in the study of surfaces, the construction method is simple and rigorous: simply combine tori and projective planes to construct any surface. In the study of knots, however, no such rigorous construction technique exists. Finding such a rigorous construction technique was one-half of the focus of my research. (And, as it turns out, my method of knot construction via self-entanglement of loops was exactly the construction technique I was looking for.) The other half of my research, which I will discuss first, was finding a knot invariant that is independent of vision, or any other particular sensory modality for that matter.

7: MINIMUM CURVATURE

[Sad Preface: The following is a record of the research I performed regarding finding a way to calculate the minimum curvature of knots. Unfortunately, the results of this particular investigation did not produce the results I was hoping to achieve. Thus, I write this record simply in the spirit of demonstrating what I did, and in the hopes that a reader more knowledgeable than I might email me and give me feedback on how to alter my approach.]

When I first decided to study knots in some way other than looking at them, it seemed an obvious choice that, if you're not going to stand outside of the knot and *look* at it, an alternative would be to travel the path of the knot and *feel* it. For instance, what would you notice if you were traveling along a knotted path through three-dimensional space, much like if you were on a roller coaster ride? If you could travel along the path of one knot, and then travel along the path of a different knot, would you be able to determine that the knots were topologically distinct? Ignoring gravitational forces, you would basically only be able to measure two things while you travel along the path of a knot: how far you traveled until you came back to your starting point, and how much your path curved away from you -- up, down, left, right -- as you traveled. This thought experiment motivated me to study knot curvature.

In order to calculate a meaningful value of knot curvature -- a value that is invariant for any deformation of a knot -- you would need to find a way to calculate curvature such that random deformations of the knot would not affect your curvature measurement. In other words, the calculation of curvature would have to be invariant under Reidemeister moves. Thus, you could not simply measure the *total* curvature of a knot, because total curvature increases indefinitely as you deform the knot more and more. Theoretically, the total curvature of a given knot could be as large as you want. If you did measure total curvature, the only unique value of total curvature for a particular knot would be the theoretical *minimum value* of the total curvature of the knot, which you would only be able to measure if you happened to be privy to the smoothest and least deformed projection of a knot. This privy perspective being unlikely, I began searching for a way to calculate the minimum curvature of a knot given any projection of a knot, and not just its smoothest and least deformed projection.

Even though I had no proof of the fact, I had five reasons for believing that knots would indeed have a specific minimum curvature. First, my intuition told me that to tie a knot out of a straightened piece of rope, you would need to add a certain minimum amount of curvature to the rope in order to tie a knot in the first place. On the other hand, if you try to smooth out a given knot, and take out any excessive deformations, there will come a point when you simply cannot smooth out the knot any further -- otherwise the knot would break apart. This intuition is somewhat related to the minimum stick number of a knot, because if you try to tie a knot out of a straight stick, you would need to *kink* the stick a certain number of times in order to tie the knot. My second reason for believing that knots would have a specific minimum curvature came from reading about research concerning the "ideal shape" of a knot (Gonzalez). The "ideal shape" of a knot is the shape a knot would have if the knot were made out of a rope whose thickness is maximized. For a given projection of a knot, there comes a point when the diameter of the rope simply cannot grow any larger unless the rope begins to intersect with itself. The projection that has the thickest rope is defined as "ideally shaped". Clearly, such an "ideally shaped" knot would not have excessive deformations. Thus, it seemed reasonable to me that the "ideal shape" of a knot is also the shape of a knot whose curvature is minimized. Third, from a text book on differential geometry, I read about Fenchel's theorem, which states that the total curvature of a closed space curve is always greater than or equal to 2*pi (Chern, pg 113). Thus, at least in general, closed space curves have a minimum curvature. Fourth, from the same differential geometry text, I read about Milnor's theorem, which states that the total curvature of a nontrivial knot is always greater than 4*pi (Chern, pg 118). Thus, all nontrivial knots have a minimum curvature of 4*pi. Fifth and finally, I found research on the Internet that posted the value of total curvature for many prime knots (Rawdon, 2001). Of course, these calculations of total curvature were not performed on knots that were "ideally shaped" or smoothed out as much as possible. Each knot was simply drawn in a certain prescribed way, and the calculations were performed on those drawings. Regardless, each knot had a distinct value of total curvature, and the values of total curvature increased roughly as a function of knot complexity, as measured by minimum crossing number.

Each of these five reasons suggested to me that each distinct knot might have a distinct value of minimum curvature. If distinct knots really did have distinct values of minimum curvature, minimum curvature would be the strongest knot invariant possible, able to distinguish between *any* two knots (excluding, perhaps, knots that are mirror-images of each other). Plus, as an added benefit to my own personal desire, the minimum curvature of a knot would actually tell you an *essential* property of a knot: how much a knot is necessarily *curved upon itself*.

Thus, my goal became discovering a way to quickly and easily calculate the minimum curvature of any given knot. The key to the measurement of minimum curvature, as I saw it, was to measure the resultant amount of curvature experienced by the *traveler* of the path of a knot. For instance, imagine driving along the following two paths (in the direction of the arrows) through two-dimensional space:

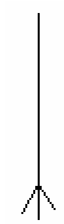


Figure 1

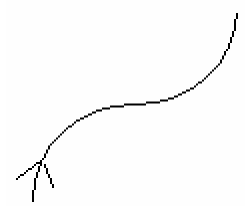


Figure 2

In Figure 1, you clearly experience zero resultant curvature: you turned left just as much as you turned right. In Figure 2, you drive one unit of distance while making a right handed-turn of a certain curvature, and then drive one unit of distance while making a left handed-turn of equal curvature. If you allow equal amounts of curvature in opposite directions (right- and left-handed directions, relative to the traveler) to cancel, then the resultant amount of curvature of the path in Figure 2 is also zero. I define this measurement process as the "resultant directed curvature" (RDC) because you take into account the *resultant* amount of up/down/left/right curvature that the traveler has experienced.

To conceptualize this process, I imagined myself carrying a small measuring device in my lap as I traveled the path of the knot. Over every infinitesimal length of the path, my measuring device would measure how far I had traveled, how much curvature I was experiencing, and in what direction I was experiencing the curvature relative to my own orientation: up, down, left, or right. (Note: the curvature I experienced is always perpendicular, or normal, to my direction of travel.) After traveling a complete circuit along the path of the knot, I would simply add up all of the curvature I experienced, allowing equal amounts of curvature in opposite directions to cancel. The result of this calculation would be a resultant amount of curvature in a particular direction. I assumed that this resultant amount of curvature would reflect the minimum curvature of the knot.

(In a fanciful way, I also imagined that this resultant curvature could also be used to measure the gravity of a given knot, using general relativity to relate the curvature of space to a particular gravitational field. For instance, if I could map the path of a knot onto a straight line, while retaining the information about the curvature and direction of curvature at each point, then, as far as the traveler is concerned, he would be traveling along a straight path that had variable gravitational forces. This was mere fancy, however, and even though I read books on general relativity to supplement this particular investigation, I never actually pursued this line of thinking. I merely mention it now to entertain the reader.)

After conceptualizing what I *wanted* to measure, my next task was to figure out how to mathematically define how to calculate RDC in such a way that the value of RDC is invariant under Reidemeister moves. Take the first Reidemeister move, for example:

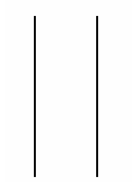


Figure 3

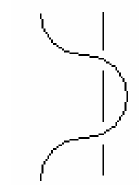


Figure 4

The only deformation that occurs in this picture is the deformation of the left strand from a straight line in Figure 3 to a bell-curved path in Figure 4. For the sake of easy calculation, assume that the curved path in Figure 4 is made up of four arcs, each arc being equal to one-fourth of the circumference of a circle with radius = 1. In Figure 5, I leave a gap between each arc to exaggerate the assumption:

Figure 5

Then, taking the perspective of a traveler moving in the direction of the arrows, you can quickly add up the amount of right hand turns and left hand turns (all of which are of equal distances and curvature) to find that they all cancel out to zero resultant curvature. Thus, at least for this contrived example, RDC is invariant under the first Reidemeister move.

Using this example as a model for how to define RDC for paths through two-dimensional space, I decided that my definition of RDC must include information about the curvature (K) of the path, the length (L) of the path, and the unit vector (N) that describes the direction of the curvature relative to the traveler. In other words, N points in the direction of the curvature (always on the concave side of the curve) and is normal/perpendicular to the tangential movement of the traveler. I found the standard way to calculate the total curvature of a closed space curve in Chern's differential geometry book: "the total curvature of a closed space curve C of length L is defined by the integral over 0 --> L, of the absolute value of K, integrated with respect to arc length" (Chern, pg 112):

Equation 1: $$\int_0^L |K(s)|\, ds$$ where 's' is arc length.

However, I wanted to enhance this calculation by removing the absolute value from K and including information about the direction of the curvature normal to the traveler, N(s). Thus, I altered the above formula like so:

Equation 2:
$$\int_0^L \mathrm{N}(s)\,\mathrm{K}(s)\,ds$$

Furthermore, the only parameterizations I was able to find of knotted paths were defined in terms of a dummy variable 't', rather than arc length 's'. Thus, I further altered the formula like so:

Equation 3:
$$\int_0^L \mathrm{N}(t)\,\mathrm{K}(t)\,\mathrm{L}(t)\,dt$$

where L(t) is a measure of arc length. Equation 3 shows the final form of my definition of RDC. Using the above formula is easy enough to do with simple, contrived examples of paths through two-dimensional space. For instance, take a simple circle of radius 'r':

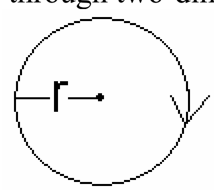


Figure 6

In this case, K(t) = 1/r and N(t) is always only in one direction: either to the left of the traveler or to the right of the traveler, depending on the direction of travel along the path. For this example, assume a left-handed turn is the negative direction, and a right-handed turn is the positive direction, and that the traveler travels around the circle clockwise. Thus, N(t) = 1 always, because the traveler experiences a constant right-handed turn of curvature K(t) = 1/r. Therefore, the resultant directed curvature of a circle with radius 'r' is:

RDC $= \int_0^L \mathrm{N}(t)\,\mathrm{K}(t)\,\mathrm{L}(t)\,dt$

$= \dfrac{\int_0^L \mathrm{L}(t)\,dt}{r}$ (and because the circumference of a circle is 2*pi*r)

= (1/r)*(2*pi*r) = 2*pi.

This result agrees with Fenchel's theorem, which states that the total curvature of a circle is 2*pi (Chern, page 113). Similarly, each of the drawings of non-intersecting closed paths through two-dimensional space in Figure 7 will also have RDC = 2*pi. (As in Figure 5, I made these drawings out of arcs equal to one-fourth of the circumference of a circle with radius = 1, and I left a gap between each arc to make the drawings easy to analyze).

Figure 7

The reader can check that the RDC for each of these drawings will be equal to plus or minus 2*pi, depending on how you travel the path.

After experimenting with this definition of RDC for paths through two-dimensional space, and privately becoming convinced that the RDC of any closed nonintersecting path through two-dimensional space would always be 2*pi, I finally found a proof that confirmed my suspicion. The proof basically stated that if you travel the path of any closed nonintersecting path through two-dimensional space in one direction for a complete circuit of the path, then your tangent vector will rotate a resultant amount of 2*pi. Thus, no matter how much you deform a closed nonintersecting path through two-dimensional space, the RDC of the path will always equal 2*pi.

I did not find a corresponding proof for paths in three-dimensions, but I took the result regarding paths in two-dimensional space as an indication that perhaps the RDC of a given three-dimensional closed nonintersecting path would also be invariant to deformation. If the RDC of a path through three-dimensional space was invariant to deformation, then the only reason that RDC should differ between any two knots is if the knots are topologically distinct.

At this point in the explanation of my approach to calculating the minimum curvature of a knot, I should make a concession. I realize that there are a lot of "IFs" in my approach: 1) IF distinct knots have distinct values of minimum curvature, 2) IF it is possible to quickly and easily measure the value of minimum curvature for a given knot from any projection of that knot, and 3) IF RDC actually measures minimum curvature....THEN this approach will be fruitful. I openly make this concession. However, I justify this approach because, in the spirit of open-ended research, I was searching for something new, and to achieve this end, I was simply following my intuition. Plus, IF everything I conjectured was true, then it seemed reasonable to conjecture further that complex knots would have a larger value of minimum curvature than simpler knots, and that the unknot would have the smallest value of minimum curvature of all knots. Thus, IF....IF....IF...., then minimum curvature would be an exceedingly easy way to organize and tabulate knots. Whether or not my conception of

RDC would help in calculating minimum curvature, I was (and still am) firmly convinced that minimum curvature would be a useful concept in knot theory.

So, feeling enthusiastic about how well my definition of RDC worked with simple examples of paths through two-dimensional space, what I needed to do next was to write a computer program that would calculate the RDC of a knot given the parameterization of a knot: x(t), y(t), z(t).

Equation 4: Parameterization: $r(t) = x(t)i + y(t)j + z(t)k$, where i, j, and k are the standard unit vectors pointing in the x, y, and z direction
Equation 5: Unit Tangent vector: $T(t) = r'(t)/(\|r'(t)\|)$, where $\| \quad \|$ finds the magnitude of a vector.
Equation 6: Unit Normal vector: $n(t) = T'(t)/(\|T'(t)\|)$, not to be confused with N(t), the measure of normal vectors relative to the traveler.
Equation 7: Curvature: $K(t) = (\|T'(t)\|) / (\|r'(t)\|)$

The easy part of this task was finding parameterizations of knots that I could input to the computer program. I found parameterizations for 14 distinct knots on the Internet (Trautwein 1995). The challenging part of this task was figuring out how to measure N(t), the curvature relative to the traveler in three-dimensional space.

In order to define N(t) in three-dimensional space, I first worked with finding a definition of N(t) in two-dimensional space, hoping that I could easily extend the definition to three-dimensions. I knew that I could not simply calculate RDC using the standard definition of the unit normal vector because the normal vector is defined with reference to the origin of the coordinate system, not to the traveler along the path. For instance, if I simply calculated the RDC of a circle (see Figure 8) using the standard definition of the normal vector, every pair of normal vectors from opposite sides of the circle would simply cancel out, yielding RDC = 0.

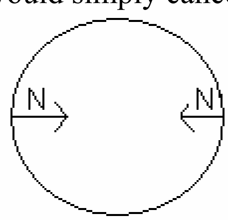


Figure 8

I decided the easiest way to measure the direction of curvature in two-dimensions was to unravel the path, using a rotation function to transform the closed path into a straight line. For instance, Figure 9 shows a circular path that a traveler travels in a clockwise motion, experiencing curvature to *his* right. Figure 10 shows the unraveled picture of the same path, giving a good intuitive idea of what the traveler is experiencing: forward motion, with an experience of curvature to the right.

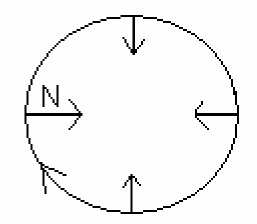


Figure 9

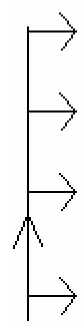


Figure 10

After unraveling the path, it appears as if the traveler travels in a straight line, but the traveler is still able to calculate the original value of curvature of the original path at every point. As another example, consider the middle drawing from Figure 7. Figure 11 shows this drawing again, and Figure 12 shows an approximate picture of what this path would look like unraveled. I marked the starting point of travel with a large dot, and the direction of travel is indicated with an arrow on the path itself.

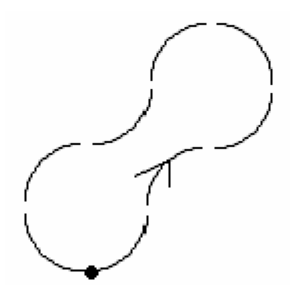


Figure 11

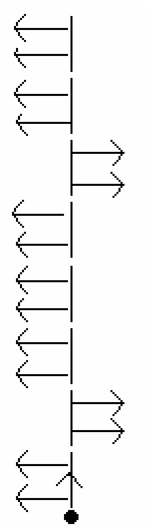


Figure 12

The benefit of this unraveling process is that, whereas in Figure 11 the normal vectors point in any direction in the plane, in Figure 12 the normal vectors only point in the direction of either the positive x direction or the negative x direction (that is, as defined by a standard two-dimensional Cartesian coordinate plane). This unraveling transformation captures the fact that the traveler of the path in two-dimensions would only experience curvature to his left (negative x direction) or to his right (positive x direction).

To accomplish this unraveling of the path, I simply used a rotation function, at each point along the path, to rotate the tangent vector and normal vector equal amounts such that the tangent vector pointed in the positive y direction (relative to a standard Cartesian coordinate plane) In Figure 13, I show a circle with three randomly chosen points shown with their normal and tangent vectors. In Figure 14, I show those normal and tangent vector pairs after rotation. Every other pair of normal and tangent vectors at every other point on the circle would undergo a similar rotation, yielding a final result similar to Figure 12.

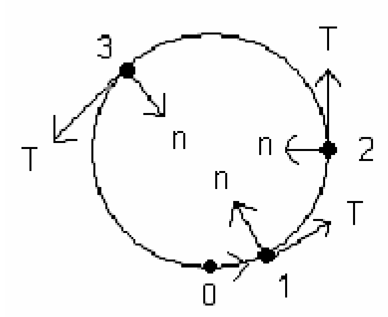


Figure 13

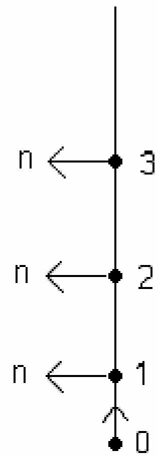


Figure 14

Thus, a rotation function, defined on the tangent and normal vector, results in placing the normal vector in either the positive x or negative x direction. And, because the normal vector is a unit vector, the value of n(t) simply becomes either 1 or -1.

Armed with a rotation function that worked conceptually in the two-dimensional case, I attempted to extrapolate my rotation function to three-dimensions. I reasoned as follows. As the traveler moves along the knotted path through three-dimensional space, he moves in the direction of the tangent vector, and experiences curvature in the direction of his normal vector: either up, down, left, or right. Just as I rotated the normal vectors of the traveler in two-dimensional space onto a one-dimensional line (to represent the right- and left-handed turns of the traveler), I wanted to rotate the normal vectors of the traveler in three-dimensional space onto a two-dimensional plane (to represent the right, left, upward, and downward turns of the traveler), specifically: the xy plane. Thus, with the help of my faculty advisor Dr. Robinson, I defined a rotation function that rotates the tangent vector into the z direction in a prescribed way, and rotates the normal vector in the same prescribed way, which results in placing the normal vector in the xy plane, while still retaining the original relationship between n(t) and T(t). I assumed that the rotation function would simply translate the experienced direction of curvature of the traveler into one plane, which would make analysis simple. Once the normal vectors are rotated into the xy plane, I can simply calculate the RDC for each component -- x and y -- , and then combine the component RDCs into a final RDC value.

I decided to use the polar coordinate system as a background for my three-dimensional rotation function. In the polar coordinate system, every point in space is defined by an angle 'phi', which denotes the angular distance from the z axis to the z-component of the point, and by an angle 'theta', which denotes the angular distance from the x axis to the x-component of the point. Thus, for every tangent vector on the path of the knot, I first found the value of phi and theta. Then, I rotated the tangent vector by -phi and -theta, such that the tangent vector pointed in the z direction, and rotated the normal vector by the same -phi and -theta, such that the normal vector lay entirely in the xy plane. Thus, the rotation function transformed n(t) into N(t), which, according to my assumption, should represent the direction of the curvature as experienced by the traveler. The rotation function itself is the following:

Equation 8:

Given the unit Tangent vector, T = (a, b, c), such that either a > 0 or b > 0, then
Theta = arccos(a/[(a^2 + b^2)^(1/2)])
Phi = arcos (c)
Given the unit normal vector, n = (d, e, f), then the rotated normal vector, N, is:
N = (-fsin(phi) + (dcos(theta) + ysin(theta))cos(phi) ,
-dsin(theta) + ecos(theta),
fcos(phi) + (dcos(theta) + esin(theta))sin(phi))

Otherwise, if a = b = 0, then the Tangent vector is already pointing in the z direction, and no rotation function is needed.

Note: This rotation function rotates n(t) by -theta and -phi. Thus, N(t) should only have two components: one component in the x direction and one component in the y direction.

At this point in my discussion, I am ready to describe the algorithm of my computer program that calculates RDC, and to show the computer program itself. The algorithm is as follows:

1) Define the parameterization of the knot: x(t), y(t), z(t)
2) Define the equations for the Unit Tangent Vector, T(t), the Unit Normal Vector, n(t), and the Curvature, K(t). (Defined above, in Equations 4, 5, 6, and 7).
3) Use the rotation equation (Equation 8) to transform n(t) into N(t).
4) For each component of N(t), use Equation 3 to find the RDC of each component. Label these two results I and J.
5) Calculate the magnitude of I and J, which should equal RCP of the knot.

I used the mathematics computer program Maple 7 to run the above algorithm. Here is the program, along with the results the program produced while calculating the RDC of a unit circle that lies entirely in the xy plane. This demonstration calculates the RDC of a unit circle because the values of K(t), ||n(t)||, and ||T(t)|| are whole numbers, making the steps of the program easier to follow.

RDC Computer Program

Comment: Define the parameterization of the knot -- x(t), y(t), z(t) -- to be used in r(t) = x(t)i + y(t)j + z(t)k. In this case, the knot is the unknot, as represented by a unit circle in the xy plane..

```
x := t -> cos(t);
y := t -> sin(t);
z := t -> 0;
```

$x := \cos \quad y := \sin \quad z := 0$

Comment: Define the derivative of each component of the parameterization, to be used in r'(t). Use 'p' (for prime) to denote the derivative.

```
xp := unapply( diff(x(t),t), t);
yp := unapply( diff(y(t),t), t);
zp := unapply( diff(z(t),t), t);
```

$xp := t \rightarrow -\sin(t)$ $yp := \cos$ $zp := 0$

Comment: Check that the magnitude of r'(t) is equal to 1.

```
rpmag := unapply( simplify( sqrt((xp(t))^2 + (yp(t))^2 + (zp(t))^2) ), t);
```

$rpmag := 1$

Comment: Define each component of the Unit Tangent Vector, T(t).
Use standard i,j,k vector variables to denote the x, y, and z components.

```
Tx := unapply(  (xp(t))/(rpmag(t)), t);
Ty := unapply(  (yp(t))/(rpmag(t)), t);
Tz := unapply(  (zp(t))/(rpmag(t)), t);
```

$Tx := t \rightarrow -\sin(t)$ $Ty := \cos$ $Tz := 0$

Comment: Define the derivative of each component of the Unit Tangent Vector

```
Txp := unapply(  diff(Tx(t),t), t);
Typ := unapply(  diff(Ty(t),t), t);
Tzp := unapply(  diff(Tz(t),t), t);
```

$Txp := t \rightarrow -\cos(t)$ $Typ := t \rightarrow -\sin(t)$ $Tzp := 0$

Comment: Check that the magnitude of T'(t) is equal to 1.

```
Tpmag := unapply( simplify(  sqrt((Txp(t))^2 + (Typ(t))^2 + (Tzp(t))^2) ), t);
```

$Tpmag := 1$

Comment: Define each component of the Unit Normal Vector, n(t).

```
nx := unapply( (Txp(t))/(Tpmag(t)), t);
ny := unapply( (Typ(t))/(Tpmag(t)), t);
nz := unapply( (Tzp(t))/(Tpmag(t)), t);
```

$nx := t \rightarrow -\cos(t)$ $ny := t \rightarrow -\sin(t)$ $nz := 0$

Comment: Check that the unit normal vector is indeed perpendicular to the unit tangent vector by verifying that the dot product of T and n equal zero.

```
simplify((Tx(t))*(nx(t)) + (Ty(t))*(ny(t)) + (Tz(t))*(nz(t)));
```

0

Comment: Check that the curvature at each point is equal to 1.

```
K := unapply( simplify(  (Tpmag(t))/(rpmag(t)) ), t);
```

$K := 1$

Comment: Define the value of theta in two ways, in order to make calculations simplify nicely. Define "thetac" for use with cosine functions; define "thetas" for use with sine functions.

```
thetac := unapply(  piecewise( Tx(t)^2 + Ty(t)^2 = 0, 0, simplify( arccos(  (Tx(t)) /
(sqrt(Tx(t)^2 + Ty(t)^2)))   )   ), t);
thetas := unapply(  piecewise( Tx(t)^2 + Ty(t)^2 = 0, 0, simplify( arcsin(  (Ty(t)) /
(sqrt(Tx(t)^2 + Ty(t)^2)))   )   ), t);
```

$$thetac := t \to \text{piecewise}\left(\sin(t)^2 + \cos(t)^2 = 0,\ 0,\ \frac{1}{2}\pi + \arcsin(\sin(t))\right)$$

$$thetas := t \to \text{piecewise}\left(\sin(t)^2 + \cos(t)^2 = 0,\ 0,\ \frac{1}{2}\pi - \arccos(\cos(t))\right)$$

Comment: Define the value of phi in two ways, in order to make calculations simplify nicely. Define "phic" for use with cosine functions; define "phis" for use with sine functions.

phic := unapply(arccos(Tz(t)), t);
phis := unapply(piecewise(Tx(t)^2 + Ty(t)^2 = 0, 0, simplify(arcsin(sqrt(Tx(t)^2 + Ty(t)^2)))), t);

$$phic := t \to \frac{1}{2}\pi \quad phis := t \to \text{piecewise}\left(\sin(t)^2 + \cos(t)^2 = 0,\ 0,\ \frac{1}{2}\pi\right)$$

Comment: Define each component of the function N(t), which (assumedly) represents the direction of curvature relative to the traveler.

Nx := unapply(simplify(-(nz(t))*(sin(phis(t))) + ((nx(t))*cos(thetac(t)) + (ny(t))*sin(thetas(t)))*cos(phic(t))), t);

Ny := unapply(simplify(-(nx(t))*sin(thetas(t)) + (ny(t))*cos(thetac(t))), t);

Nz := unapply(simplify((nz(t))*(cos(phic(t))) + ((nx(t))*cos(thetac(t)) + (ny(t))*sin(thetas(t)))*sin(phis(t))), t);

$$Nx := 0 \quad Ny := 1 \quad Nz := 0$$

Comment: Define the arclength formula, L(t).

arclength := unapply(sqrt(xp(t)^2 + yp(t)^2 + zp(t)^2), t);

$$arclength := t \to \sqrt{\sin(t)^2 + \cos(t)^2}$$

Comment: Define the RDC of the x component of the traveler's curvature, and the y component of the traveler's curvature.

xcomponent := Int((Nx(t))*(K(t))*(arclength(t)) , t = 0..2*Pi);
ycomponent := Int((Ny(t))*(K(t))*(arclength(t)) , t = 0..2*Pi);

$$xcomponent := \int_0^{2\pi} 0\, dt \quad ycomponent := \int_0^{2\pi} \sqrt{\sin(t)^2 + \cos(t)^2}\, dt$$

Comment: Simplify the values of preceding components.

Finalx := evalf(xcomponent);
Finaly := evalf(ycomponent);

$$Finalx := 0. \quad Finaly := 6.283185307$$

Comment: Calculate the final value of RDC.

RDC := sqrt((Finalx)^2 + (Finaly)^2);

$$RDC := 6.283185307$$

This computer program produced the results I expected for paths that lay exclusively in the xy plane. Inputs of both circles and ellipses in the xy plane produced a value of RDC = 2*pi. However, the above program did not produce similar results for circles that extended in all three dimensions. After analyzing my algorithm, I found the error to lie in my conceptualization of what is necessary to mathematically define N(t) relative to the traveler. My rotation function did not smoothly map the normal vectors as I thought it would. At this point, I am not sure how to modify the definition of N(t) in order to reflect the curvature experienced by the traveler.

While this particular approach failed, I am still hopeful that it is possible to find a way to measure the minimum curvature of a knot. I simply have a powerful hunch that distinct knots are fundamentally different with respect to their minimum curvature. For instance, assume you have a long bar of rubber whose tendency is to be as straight as possible. Also, assume you are able to measure the amount of stress in the bar: zero when the bar is straight, and a positive value when the bar is bent/deformed, such that the value of stress in the bar increases as you continue to deform the bar more and more. Using this bar, you could tie a knot and measure how much stress is in the bar. It seems clear enough that each distinct knot would have a different experimental value of stress. Of course, the stress you are measuring is actually just a measure of how much the bar is curved. This is simply a thought experiment, but it reflects a deeply rooted hunch within me that an essential difference between knots is their minimal curvature.

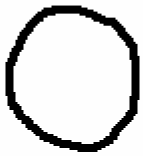
8: PRIME KNOT CONSTRUCTION VIA SELF-ENTANGLEMENT

I will now describe the prime knot construction technique that I created during the course of my research. This construction technique was the focus of my research, and is my one original contribution to the field of knot theory. While this construction technique still requires a good deal of work and refinement, I can give a basic outline of how it works and what I eventually hope to achieve.

As stated previously, when I first started studying prime knots, I conceived of knots as self-entangled loops (not to be confused with the 'tangles' used in Conway's notation). For instance, consider the right-handed trefoil:

Figure 15

Using my method of self-entanglement, I construct the right-handed trefoil from a loop according to the following method:



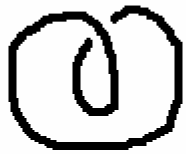



Figure 16 Figure 17 Figure 18 Figure 19

Starting with an unknotted loop (Figure 16), I put a dimple in the top of the loop (Figure 17). Then, I put a right-handed twist in dimpled part of the loop (Figure 18). Finally, I clasp the twisted segment onto the bottom of the loop (Figure 19). Thus, the "clasping" operation is not a deformation of the loop; rather, it is a temporary breaking of the loop, serving the purpose of allowing one segment of the loop to pass through another segment of the loop.

I define the twist in Figure 18 as a right-handed twist because if you were to use your right hand to make the twist in the dimpled segment of the loop, as in Figure 20:

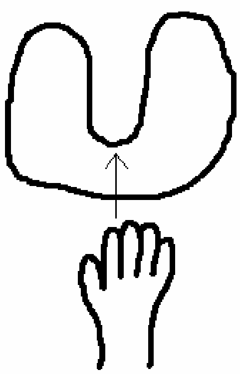


Figure 20

your hand would rotate to the right, according to the "righty-tighty, lefty-loosy" rule. (This terminology agrees with the standard knot theory definition of "positive/right-handed" and "negative/left-handed" crossings.) Furthermore, just as you can twist a segment of the loop to the right or left:

Figure 21: Left-handed twist (TL)

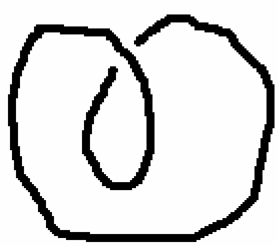


Figure 22: Right-handed twist (TR)

you can clasp a segment of the loop to the right or left:

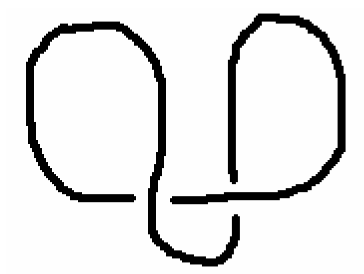


Figure 23: Left-handed clasp (CL)   Figure 24: Right-handed clasp (CR)

The direction of the clasp is defined by the side of the downwardly-directed (relative to you, the viewer) dimpled part of the loop (see Figure 25) that is passed *under* the part of the loop onto which it is being clasped.

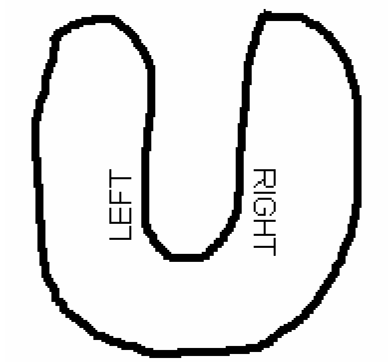


Figure 25

Thus, similar to the construction of the right-handed trefoil, I construct the left-handed trefoil by putting a left-handed twist in the top of the loop, and then using a left-handed clasp to clasp the twisted segment onto the bottom of the loop:

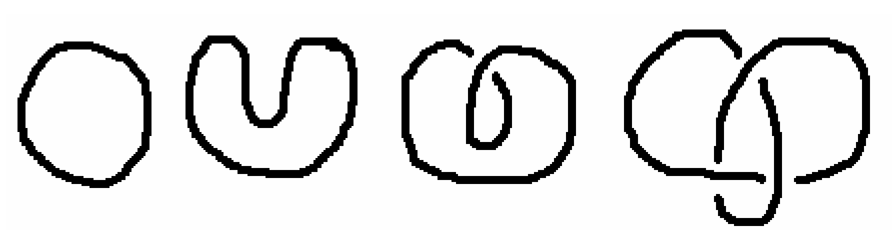


Figure 26

According to this construction technique, the self-entanglement of the right-handed trefoil can be symbolized by (TRCR), because, operating upon the unknot, you put a right-handed twist in the top of the loop, TR, and then clasp this twisted piece of loop onto the bottom of the loop with a right-handed clasp, CR. Similarly, the self-entanglement of the left-handed trefoil can by symbolized by (TLCL).

The previous examples of the trefoils capture the essence of the self-entanglement construction technique. The definition of this technique is simple:

Self-entanglement: the process whereby one continuous segment of a loop is (T)wisted, either right-handed (TR) or left-handed (TL), 'x' number of times (where x = 0, 1, 2, 3,...), and then (C)lasped, either right-handed (CR) or left-handed (CL), onto another continuous segment of the loop.

I call each instance of self-entanglement a "tentacle", simply because when I entangle two segments of a loop (Figure 27), it looks like a tentacle is released from one part of the loop (Figure 28) and clasped onto the other part of the loop (Figure 29):

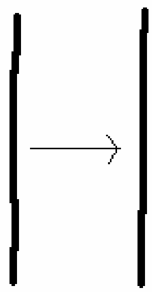
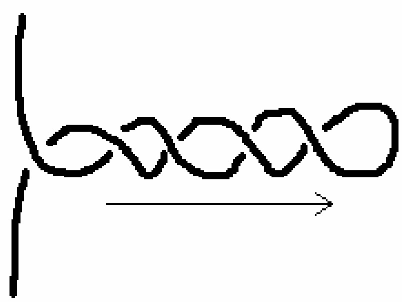
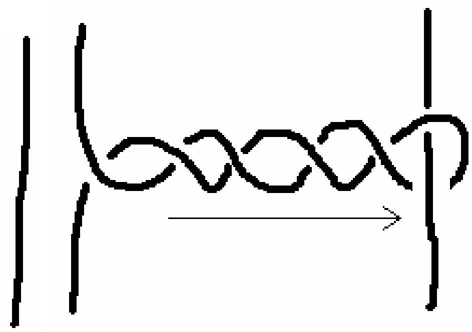
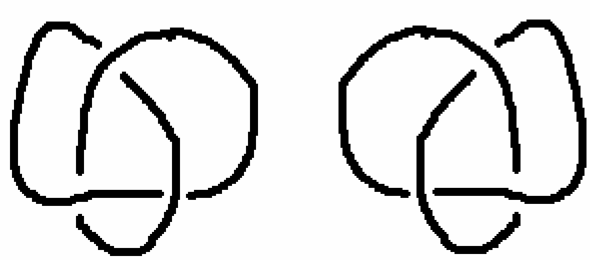

,

,


Figure 27 Figure 28 Figure 29

Thus, every tentacle can be symbolized by some number of 'T's, followed by a single 'C', denoting the fact that every tentacle has 'x' number of twists before being clasped onto another segment of the loop. A single tentacle is represented by the notation (T….TC), where the only variables are the number of T's and the handedness (right or left) of the twisting and clasping. Using this notation, *every* prime knot can be labeled in terms of its component tentacles. (I will support this claim later in this report).

As an example of this labeling system, consider the following prime knots in Figures 30 - 37. Below each knot is listed first its tabulation number (as given in any standard prime knot table), then a letter in parenthesis to indicate if the knot is (L)eft-handed, (R)ight-handed, or (A)mphicheiral (meaning the left- and right-handed version of a given knot are topologically identical), and then the symbols representing the type of tentacles in the knot. I put the symbols for each tentacle within parentheses, and I put all of the tentacle labels for a given knot within brackets.

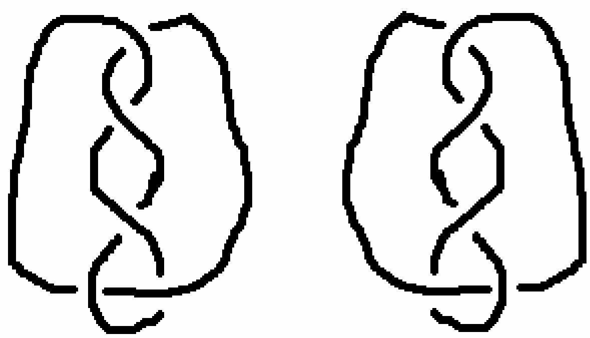


3.1(L) {(TLCL)} 3.1(R) {(TRCR)}

Figure 30

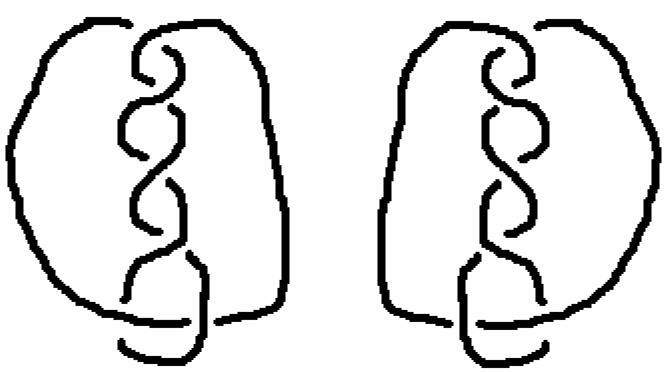
4.1(A) {(TRTRCR)} or {(TLTLCL)}

Figure 31



5.2(R) {(TRTRTRCR)} 5.2(L) {(TLTLTLCL)}

Figure 32



6.1(R) {(TRTRTRTRCR)} 6.1(L) {(TLTLTLTLCL)}

Figure 33

Figures 30 - 33 all contain prime knots that consist of a single tentacle. The only difference between these knots is the number of twists in the tentacle and the handedness of the twists and clasps.

Now consider the knots in Figure 34:

5.1(R) {(TRCR),(CR)}   5.1(L) {(TLCL),(CL)}

Figure 34

Consider knot 5.1(R), which consists of two tentacles. The first tentacle is (TRCR), which is the same self-entanglement that transforms an unknotted loop into the right-handed trefoil. The second tentacle in knot 5.1(R) is (CR), which is simply a right-handed clasp. The construction of 5.1(R) from the right-handed trefoil is shown in Figure 35:

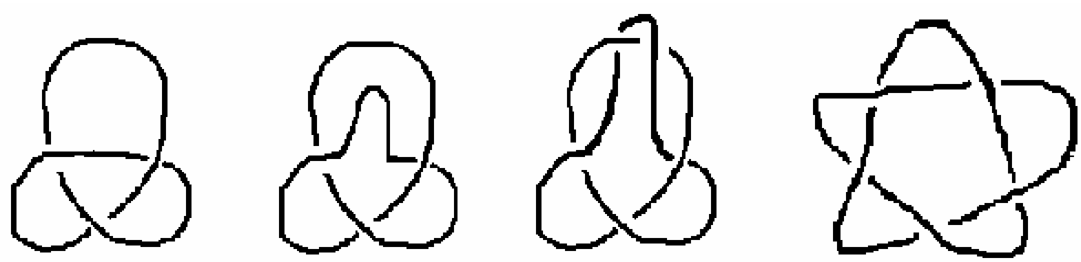
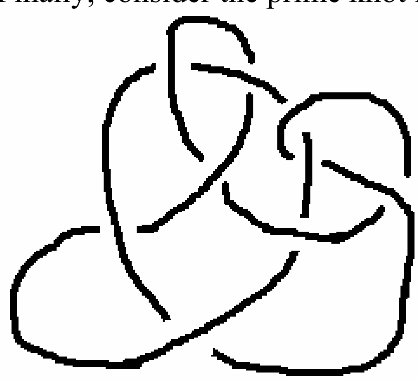


Figure 35

Finally, consider the prime knot in Figure 36:



9.27(L) {(TLCL)(TLCL)(TRCR)}

Figure 36

Knot 9.27(L) consists of three tentacles. A detailed account of the construction of Knot 9.27(L) is given in Figure 37. Each row of steps demonstrates the construction of one tentacle.

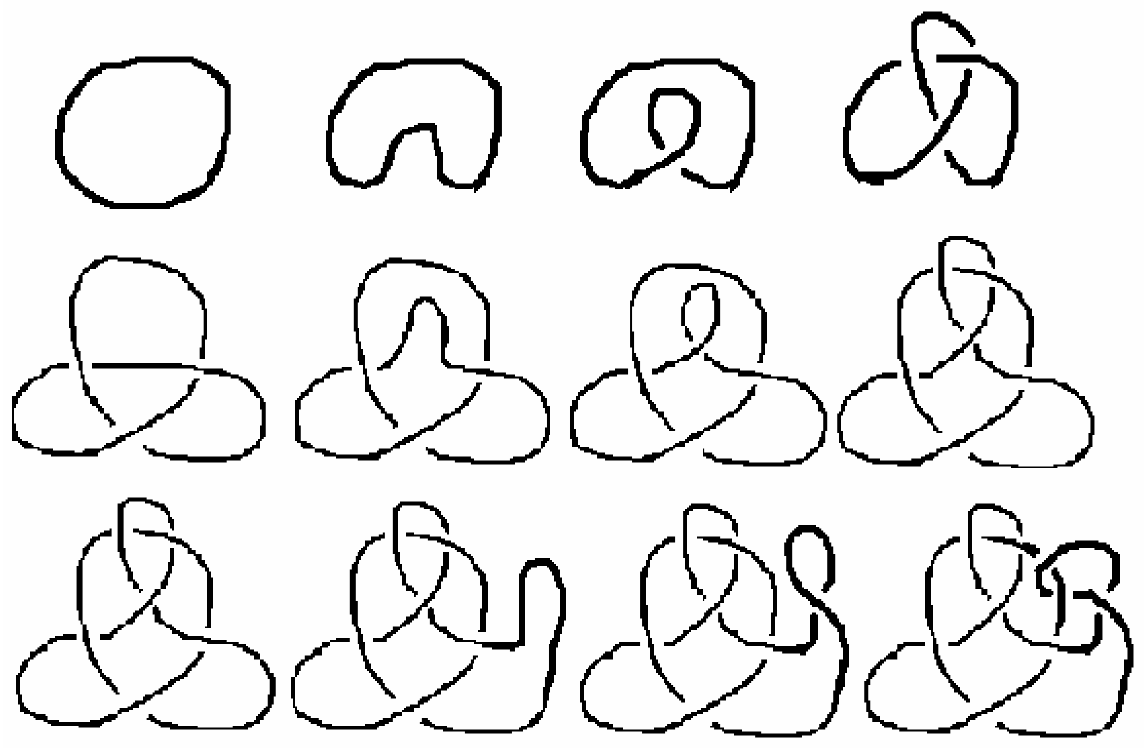


Figure 37

The first row shows the construction of the first tentacle: (TLCL). The second row shows the construction of the second tentacle: (TLCL). The third row shows the construction of the third tentacle: (TRCR). Thus, I label knot 9.27(L) with {(TLCL)(TLCL)(TRCR)}. Note that the order of construction of the second and third tentacle could have been reversed without altering the resultant knot. I will discuss the implications of the variability of tentacle placement later in this report.

The previous examples help to demonstrate two important facts about this construction technique. The first fact is that knot construction via self-entanglement is a recursive process that constructs complex knots from simpler knots, using the unknot as the base case. At each point in the construction process, I recognize a "parent" knot, upon which a single tentacle is constructed, and a child knot, which is the result of that particular tentacle construction. Thus, according to this construction technique, *all prime knots are related to each other according to parent/child relationships*. The implication of this fact is that prime knots can be organized in a definitive way within an evolutionary tree with the unknot as the root.

The second, and most important fact is that according to this construction technique, knots can be conceived of in a way that clearly demonstrates what it actually means for a loop to be *knotted*. For instance, just as every two-dimensional surface can be simply and rigorously constructed out of some number of tori and projective planes, so, too, can any prime knot be simply and rigorously constructed out of some number of tentacles. In other words, I am claiming that the essential and actual difference between knots -- the essential property I originally set out to find in my research -- is the complexity of self-entanglement, as measured by tentacles. Indeed, self-entanglement and *knotted-ness* are synonymous. Furthermore, tentacle construction can be said to be the one and only Anti-Reidemeister move, because it is the only way to change the topology of a knot.

Thus far, I have introduced the idea of self-entanglement and provided examples of how knots are composed of what I call tentacles. I have much yet to discuss, so I would like to offer a brief outline of what is still to come in this discussion of self-entanglement.

First, I will present a proof that every prime knot can be constructed via self-entanglement.

Second, I will discuss a few rules I have defined that streamline the construction process and eliminate the construction of duplicate knots.

Third, I will outline some of the parent/child relationships I have found among the prime knots I have studied, and present a currently incomplete evolutionary tree that clearly demonstrates those parent/child relationships.

Fourth, I will demonstrate how the concept of self-entanglement can be applied equally as well to links.

Fifth, I will discuss some of the currently unresolved difficulties of this construction technique, and what still must be done before I can write a construction algorithm that can be implemented by a computer.

Sixth, I will offer some conjectures regarding this construction technique.

Seventh, I will conclude my discussion of knot construction via self-entanglement with a concise summary.

Proof that every prime knot can be constructed via self-entanglement

The proof that every prime knot can be constructed via self-entanglement is simple and intuitive. Given any prime knot, it is possible to deconstruct this knot by removing tentacles. That is: unclasp a given part of the knot, and unravel any twists in that unclasped part. Excluding wild knots (knots with an infinite number of crossings), a given prime knot will only have a finite number of tentacles to deconstruct. Once all of the tentacles are removed, the result will be the unknot. Thus, to construct the original prime knot, simply construct each tentacle in the reverse order by which they were removed. Therefore, every prime knot can be constructed via self-entanglement. (Indeed, this proof applies equally well to knots and links, except that links are two or more loops that are *entangled*, rather than a single loop that is *self-entangled*. In either case, the principle of tentacle construction is the same. More on links later.)

Rules that reduce the redundancy of the construction process

The above proof guarantees that there is always at least one way to construct a given prime knot via self-entanglement. Unfortunately, there is no guarantee that there is *only* one way to construct a knot via self-entanglement. In fact, there is often more than one way to construct a given knot via self-entanglement. For instance, I have already discussed the fact that in the case of knot 9.27(L) in Figure 33, the order of construction of the second and third tentacle could have been reversed. Redundancy of construction seems to be the widespread bane of knot construction techniques, for all construction techniques produce duplicate knots. However, I have been able to define a few rules that effectively eliminate *some* of the redundancy of knot construction via self-entanglement. I am hopeful that I will eventually be able to define sufficiently many rules to govern this construction technique such that it will produce prime knots without redundancy. At this point in my research, I have three rules.

Rule #1: When constructing a new tentacle on a nontrivial prime knot, do not clasp a tentacle directly onto the same strand of the loop from which the tentacle was made, for this results in a composite knot.

A strand is traditionally defined as a continuous piece of the two-dimensional projection of the loop that extends from one under crossing to another under crossing. For example, consider the trefoil, which has three strands (numbered for convenience):

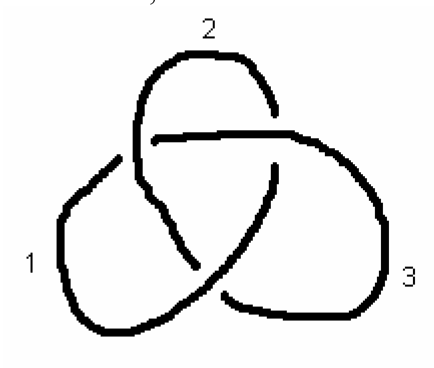


Figure 38

If I were to construct a new knot from this trefoil by constructing a tentacle that originates from strand 1 and clasping directly back onto strand 1, the result would be a composite knot (indicated by the dotted lines in the right-most drawing of Figure 39):

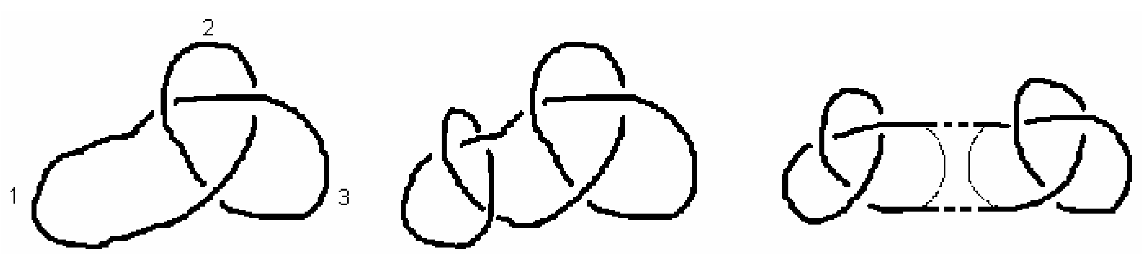


Figure 39

However, when constructing a tentacle on the unknot, there is *only* one strand. That is why Rule #1 only applies to nontrivial knots. This rule effectively limits construction via self-entanglement to the construction of prime knots, rather than prime *and* composite knots. There is simply no need to use self-entanglement to construct composite knots, because there already exist rules that govern the process of taking the composition of two knots.

Rule #2: Every tentacle is either right-handed or left-handed.

As previously stated, every tentacle consists of some number of twists followed by a clasp: (T...TC). This rule states that within a given tentacle, the handedness of each T and C must be the same. The reasons for this are obvious. First, every twist within the tentacle must be in the same direction, otherwise a TL and a TR would cancel to zero twists (see Figure 40):

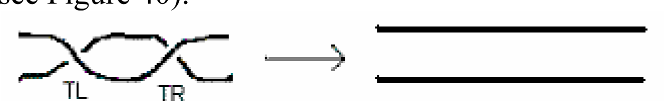


Figure 40

Second, the clasp must be in the same direction as the twists, otherwise one of the twists would cancel. For example, Figure 41 demonstrates that a tentacle consisting of (TRTRTRCL) simplifies to (TRTRCR):

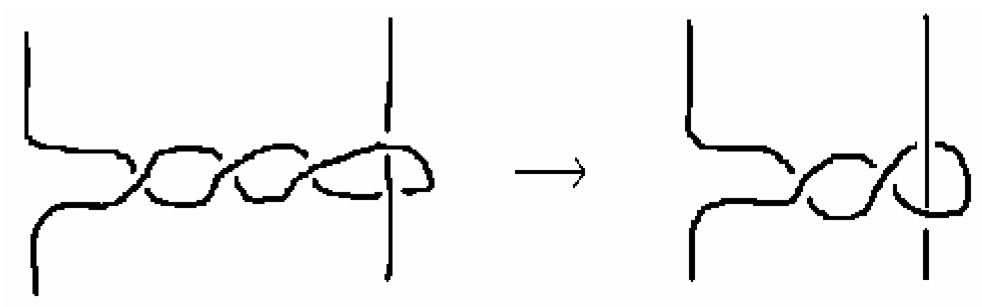


Figure 41

This rule effectively reduces the set of all possible tentacles to right-handed tentacles, { CR, TRCR, TRTRCR, TRTRTRCR, ...}, and left-handed tentacles, {CL, TLCL, TLTLCL, TLTLTLCL, ...}.

Rule #3: Do not construct a tentacle consisting of a single clasp -- either CR or CL -- in order to connect two strands that were twisted together in a previous tentacle, for this alters that previous tentacle, resulting in redundancy. In other words, "Do not clasp in the direction of the twist".

For example, consider knot 5.2(L) in Figure 42. The arrows indicate adjacent strands that are twisted together in a tentacle.

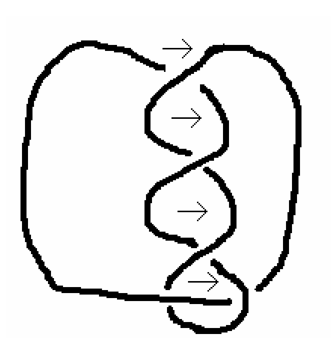


Figure 42

If you were to construct a tentacle consisting of a single *right*-handed clasp at any of the locations marked with an arrow (that is, originating the tentacle at the base of the arrow and clasping the tentacle onto the target of the arrow) the result would be knot 7.2(L), demonstrated in Figure 43:

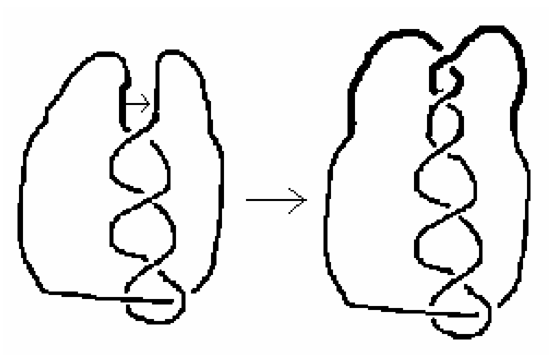


Figure 43

In this example, constructing the tentacle (CR) at the location of the arrow is redundant because the result is a knot that you would have constructed anyway by constructing the tentacle (TLTLTLTLTLCL) on the unknot.

Still referring to Figure 42: if you were to construct a tentacle consisting of a single *left*-handed clasp at any of the locations marked with an arrow, the result would be knot 3.1(L), demonstrated in Figure 44:

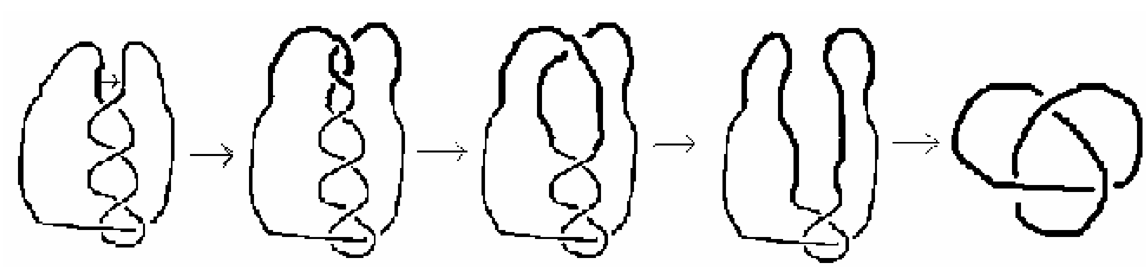


Figure 44

Thus, in this example, constructing the tentacle (CL) at the locations of the arrow is redundant because the result simplifies to a knot that you would have constructed anyway by constructing the tentacle (TLCL) on the unknot.

The significance of the locations marked with arrows is that these strands are part of a previously existing tentacle. Thus, by connecting these strands by a single clasp, you are simply lengthening or reducing that previously existing tentacle by two twists. This rule effectively reduces *some* of the redundancy of this construction technique.

Even with these three rules, there is still a great deal of redundancy is the construction of prime knots via self-entanglement. However, like I said, I am hopeful to define more rules that limit redundancy.

An outline of parent/child relationships among prime knots, and a first attempt at an evolutionary tree that demonstrates those relationships.

During the course of my research project, I had at my disposal 60 distinct prime knots to analyze according to the principles of self-entanglement. Specifically, I analyzed all prime knots with minimum crossing number 8 or less, and half of the 9 crossing knots. At present, I have been able to organize most of these knots into categories -- "generations" -- according to the total number of tentacles within each knot. Within each category, I have further organized the knots to reflect parent/child relationships. Namely, every knot in the (x+1)st generation is a child knot of a particular parent knot in the xth generation. Particular parent/child relationships are not always clear, so I present here in this report only the best examples of the relationships I have found. My ultimate goal is to create a complete evolutionary tree of all prime knots (that is, all prime knots listed in pictorial tables) that demonstrates exactly how complex knots are derived from simple knots in a recursive process of self-entanglement. Even though the following tabulation is incomplete, I hope it is complete enough to be suggestive of the value of organizing knots according to self-entanglement.

For the sake of simplicity, in the following figures, I will only provide the right-handed variety of each knot. Also, below each knot, I will include the traditional labeling number used in prime knot tables.

Zeroth Generation of Prime Knots

The zeroth generation of prime knots contains those knots that have zero tentacles. Clearly, there is only one distinct knot in this generation: the unknot.

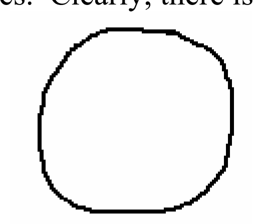


Figure 45

First Generation of Prime Knots

The first generation of prime knots contains those knots that have one tentacle. Clearly, the only way to obtain a knot with a single tentacle is to construct a single tentacle upon the unknot. For instance, Figure 46 shows all of the prime knots that I have found that contain only one tentacle.

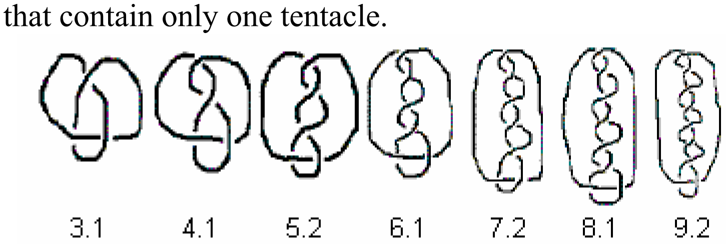


Figure 46

Each of the knots in Figure 46 was constructed by simply clasping the tentacle *directly* onto an adjacent strand. I assume that prime knots within the first generation (or any generation, for that matter) can also be constructed by wrapping the tentacles *around* other strands, or even themselves, before clasping onto their target strands, resulting in a satellite knot. I have no such knots to include in this report at this time.

Second Generation of Prime Knots

The second generation of prime knots contains those knots that have two tentacles. In order to demonstrate some of the better examples of parent/child relationships I have found within the second generation, I will present one particular parent/child relationship in each of the following figures. In each figure, I will present a row of knots. The first knot in each row will be the parent knot from the first generation, with an arrow indicating the origin and target of the tentacle. The remaining knots in each row will be the children knots of that parent, differing only in the length of their second tentacle.

In Figure 45, the parent is a trefoil, upon which tentacles are constructed internally:

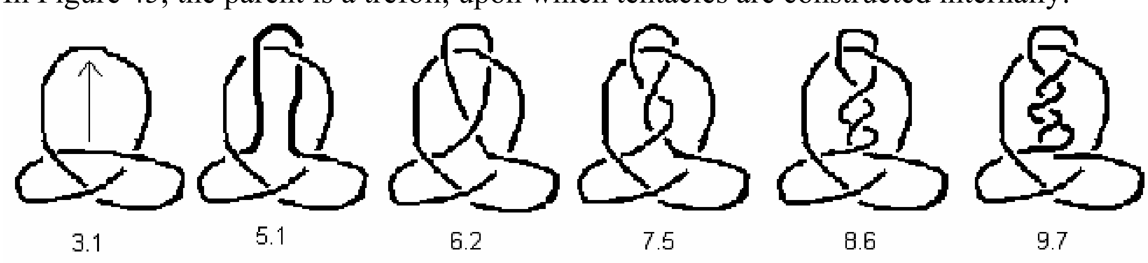


Figure 47

In Figure 48, the parent is once again a trefoil, upon which tentacles are constructed externally. Note that I did not construct a tentacle consisting of a single clasp along the direction of the arrow, because that would violate my **Rule #2**.

Figure 48

In Figure 49, the parent knot is prime knot 5.2, upon which tentacles are constructed externally.

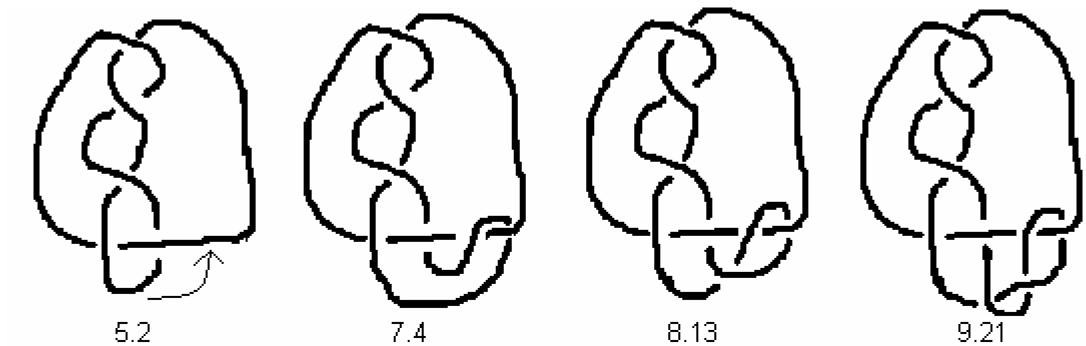


Figure 49

In Figure 50, the parent knot is once again prime knot 5.2, but the tentacles (each with one twist) are attached at various different locations. The purpose of the drawings in Figure 48 is to demonstrate how many distinct locations there are on one parent knot at which to construct a tentacle. The arrows on the parent knot indicate three possible locations for a tentacle. (There are more than three locations for this particular knot, these three are sufficient to make a point.)

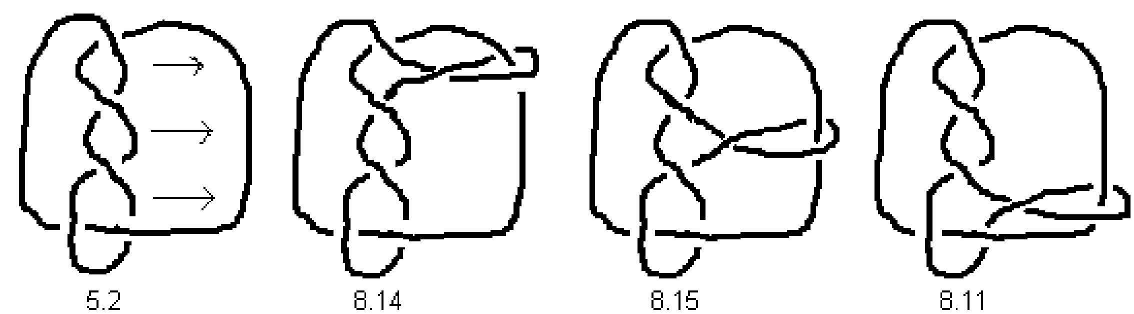


Figure 50

Each of Figures 47 - 50 demonstrate some of the best examples I know of prime knots with two tentacles that can be easily organized according to parent/child relationships. Other examples I could have shared are similar to the above examples, except with different parent knots.

Third Generation of Prime Knots

The third generation of prime knots contains those prime knots that have three tentacles. Again, every knot in the third generation is the child knot of some parent knot in the second generation. I will present the knots in this generation in the same way I presented the second generation.

The most instructive examples of prime knots with three tentacles are those constructed from the parent prime knot 5.1. In Figure 51, the parent is prime knot 5.1, upon which tentacles are constructed internally. Note the similarity between the knots in Figure 51 and in Figure 47.

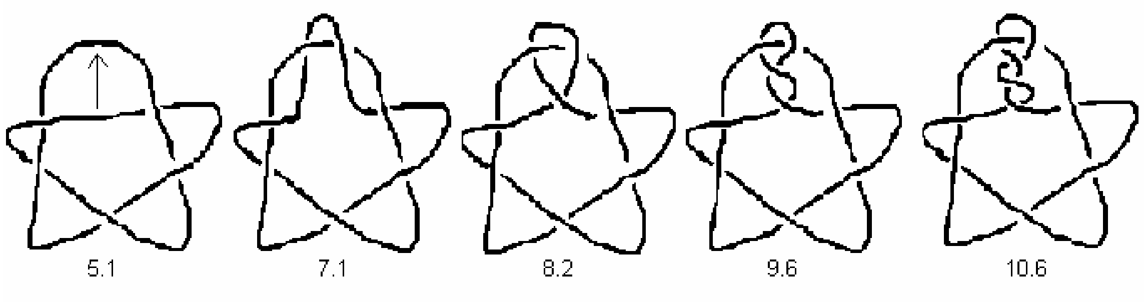


Figure 51

In Figure 52, the parent is once again prime knot 5.1, upon which tentacles are constructed externally. Note the similarity between Figure 52 and Figure 48. Also note that, once again, I did not construct a tentacle consisting of a single clasp along the direction of the arrow in Figure 52, because that would violate my **Rule #2**.

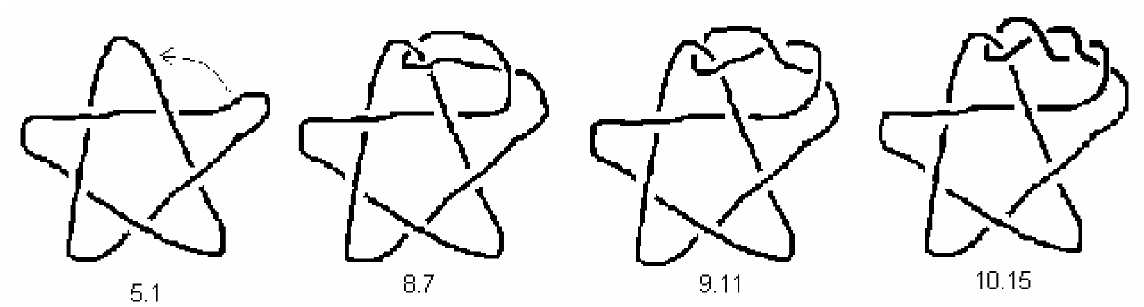


Figure 52

In Figure 53, I present the prime knot 9.27. This three-tentacled knot has an interesting relationship to the knots in Figures 47 and 48 because it is a trefoil with one tentacle constructed internally and one tentacle constructed externally (each tentacle consisting of one twist). (Reference Figure 37 to see the complete construction of this knot.)

Figure 53

In Figure 54, the parent knot is prime knot 6.2. I drew two arrows on the parent knot to show the origin and target of two possible tentacles.

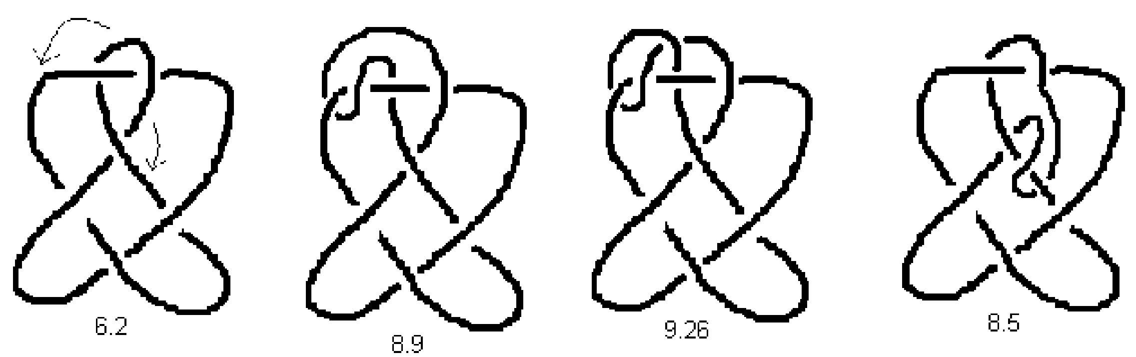


Figure 54

Fourth Generation of Prime Knots

The fourth generation of prime knots contains those prime knots that have four tentacles.

Of all the prime knots that I have currently analyzed, I have only found two prime knots that have four tentacles: knot 9.1(R) and knot 9.1(L). Consider knot 9.1(R), which is the child of the parent knot 7.1(R). The construction of knot 9.1(R) from knot 7.1(R) is shown in Figure 55:

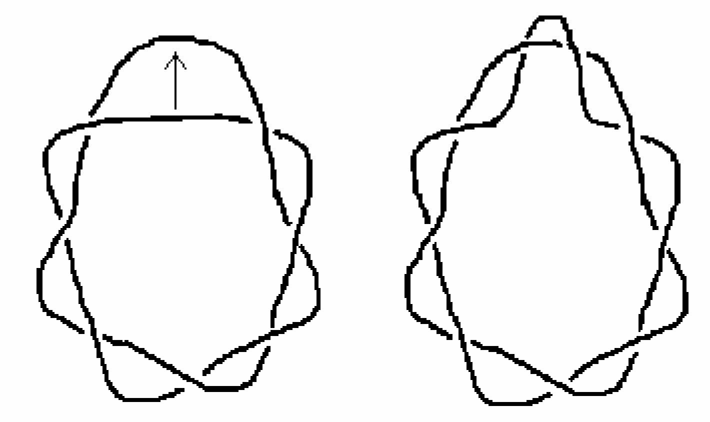


Figure 55

There may be other four-tentacled knots within the remainder of the 9 crossing knots that I have not yet analyzed. And most certainly, when I begin analyzing the 10 crossing knots, I will find more four-tentacled knots.

Fifth Generation of Prime Knots

At present, I have no knots to present in the fifth generation of prime knots, nor should I expect to find any such knots among the prime knots with minimum crossing number of 10 or less. The first fifth generation knot will occur among the prime knots with minimum crossing number 11. My argument for this claim is as follows:

A given tentacle will introduce a certain number of "crossings" into the projection of a given knot. Every twist will add a single crossing, and every clasp will add two crossings. The smallest tentacle that can be added to the unknot is (TRCR) or (TLCL), because an unknot with a single tentacle of either (CR) or (CL) is still the unknot. The smallest tentacle that can be added to any nontrivial knot is (CR) or (CL). Thus, the knot with the five smallest tentacles -- which must be notated by either (TRCR)(C?)(C?)(C?)(C?? or (TLCL)(C?)(C?)(C?)(C?), where the question marks denote uncertainty as to whether the clasps are right or left handed -- will have the least number of crossings; namely, eleven.

Thus, whenever a pictorial table of prime knots with 11 crossing is published, I can continue my analysis of prime knots into the fifth generation.

An Evolutionary Tree Demonstrating the Parent/Child Relationships for All Prime Knots with Minimum Crossing Number 7 and Less.

Figures 45-55 contained some of the best examples I currently have of parent/child relationships among prime knots. Other examples that I could have shown include different parent knots, but similar patterns of tentacle construction. For the time being -- because I have not yet finished analyzing the 8 and 9 crossing knots in terms of parent/child relationships -- I will present here an evolutionary tree that contains all of the prime knots with 7 crossings or less (see Figure 56). The following evolutionary tree is merely a beginning, as I have yet to incorporate the 8, 9, and 10 crossing prime knots. However, the following tree is indicative of the parent/child relationships I expect to find among all prime knots.

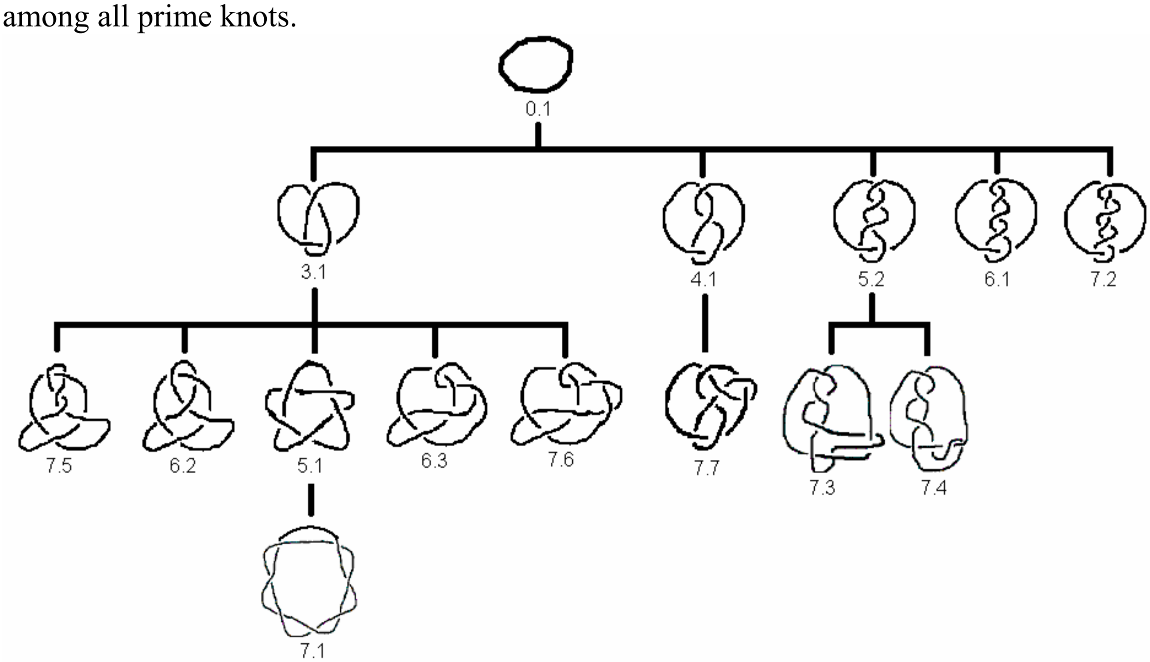


Figure 56

The xth row in this tree contains the prime knots from the xth generation of prime knots. For example, the zeroth row contains the unknot from the zeroth generation of prime knots. An evolutionary tree with 250 entries (the number of prime knots with minimum crossing number 10 and less) is indeed a monumental task, but such a completed tree would demonstrate the principle of self-entanglement perfectly.

<u>The applicability of the concept of self-entanglement to the construction of links</u>

Amazingly enough, I did not consider the application of the concept of self-entanglement to the construction of links until I began writing this final report. (Of course, links are *entangled* loops, not *self-entangled* loops, so I will begin using the term "entanglement" when referring to links.) Much to my surprise and pleasure, entanglement applies equally well to the construction of two-component links. (I have yet to study three-component links, but I expect to find similar results.) I had at my disposal a total of 31 distinct links (located in the appendix of Colin C. Adams' The Knot Book), spanning all tabulated two-component links with minimum crossing number 8 and less. 21 of these links are two unknots entangled together. The remaining 10 links are actually a prime knot entangled with the unknot. I am not quite sure at this moment in time how to incorporate the 10 unknot/knot links with the 21 unknot/unknot links in a fluid relationship, nor am I quite sure about how those 10 unknot/knot links are related to each other, so I will not include them in this report. However, I will present the parent/child relationships among the 21 unknot/unknot links in the same way I presented the parent/child relationships among prime knots. I have divided these links into generations that reflect the total number of tentacles within the link, and in every row of links I present within a given Figure#, the first link is the parent, and the remaining links are the children. After presenting each specific parent/child relationship, I will present the evolutionary tree of the 21 unknot/unknot links that reflects the parent/child relationships between them.

I have found that each of the 21 unknot/unknot links can be drawn in a way that suggests that one of the components is a fixed and immobile unit circle, while the other component is a fluid and dynamic loop, which is the source of the tentacles. I maintain this standard in all of the following drawings. Also, below each link, I label the link with the number used by standard link tables.

Zeroth Generation Of Links

The zeroth generation of links contains those links that have zero tentacles. Clearly, there is only one distinct link in this generation : the unlink.

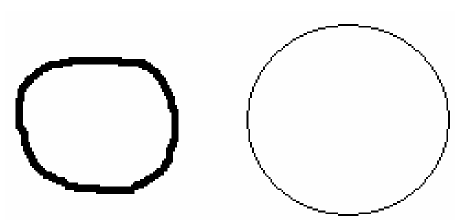


Figure 57

First Generation Of Links

The first generation of links contains those links that have one tentacle. Again, there is only one such distinct link in this generation , link 2.1:

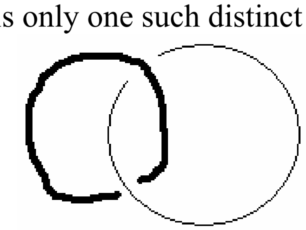


Figure 58

Second Generation Of Links

The second generation of links contains those links that have two tentacles. In Figure 59, the parent is link 2.1, upon which tentacles are constructed internally:

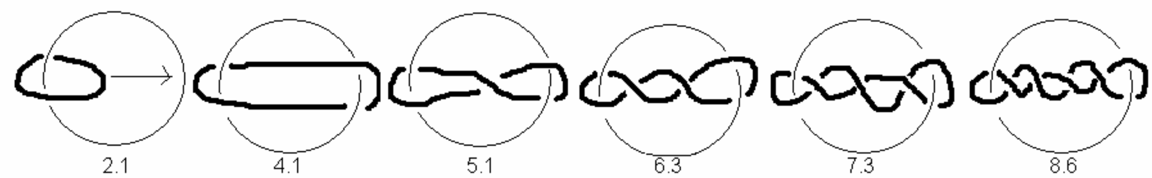


Figure 59

Third Generation Of Links

The third generation of links contains those links that have three tentacles. Every link in the third generation is the child link of some parent link in the second generation . In Figure 60, the parent is link 4.1, upon which tentacles are constructed internally:

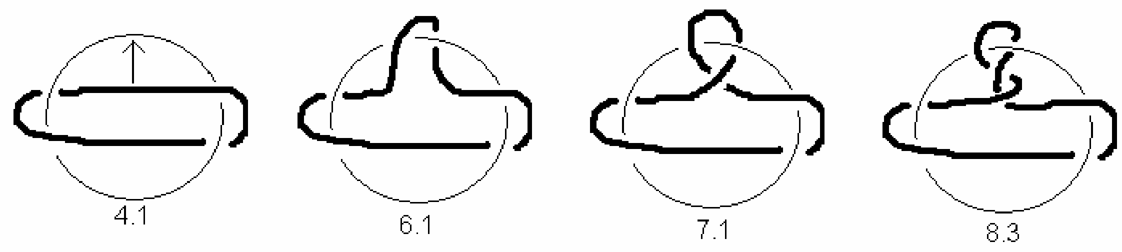


Figure 60

In Figure 61, the parent is again link 4.1, but the tentacles are constructed externally:

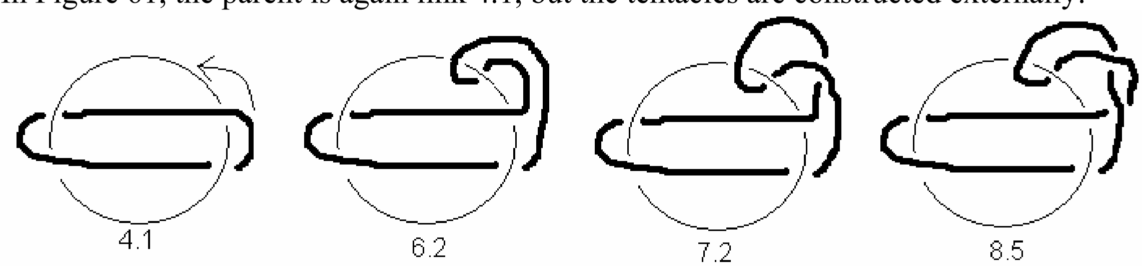


Figure 61

In Figure 62, the parent is link 5.1. The arrows on the parent link show the two locations for tentacles.

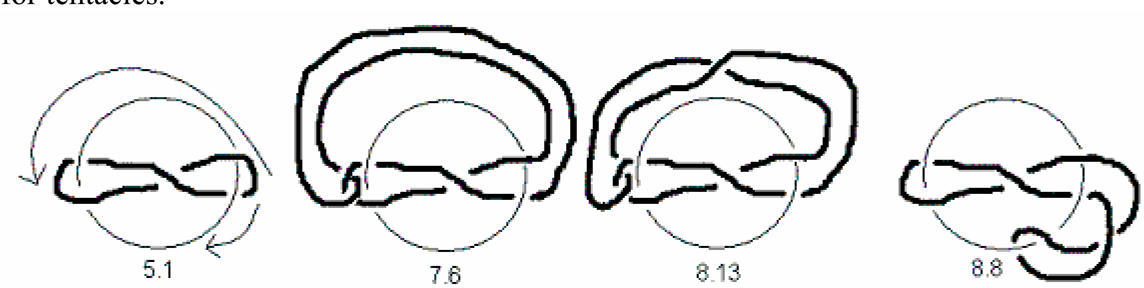


Figure 62

In figure 63, the parent is link 6.3. Again, the arrows on the parent link show the two locations for tentacles.

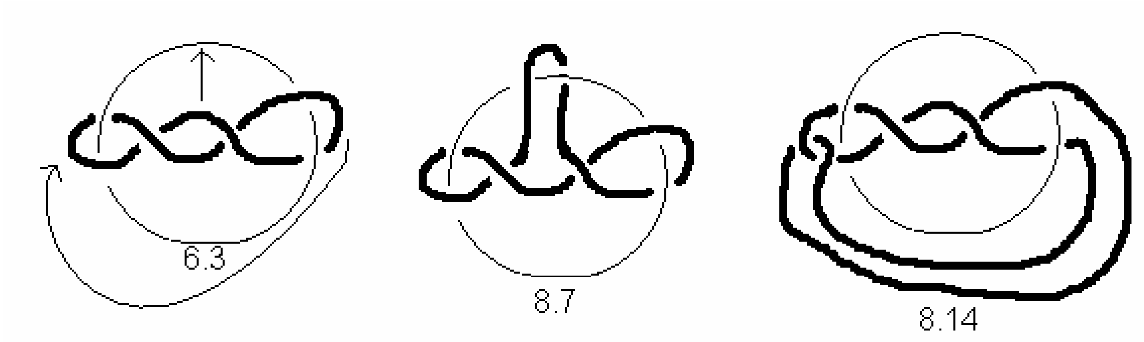


Figure 63

Fourth Generation Of Links

The fourth generation of links contains those links that have four tentacles. Of the 21 unknot/unknot links, there are only three four-tentacled links, all of which have the same parent: link 6.1 (See Figure 64).

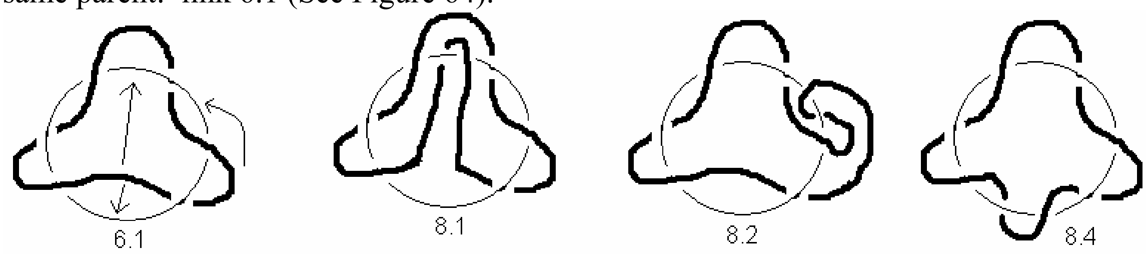


Figure 64

Fifth Generation

The fifth generation of links contains those links that have five tentacles. I found no examples of a five-tentacled link among the links with minimum crossing number 8 or less, nor did I expect to find any. Because every tentacle contains a clasp, and every clasp introduces 2 crossings into the projection of a link, the first five-tentacled link would occur among the links with minimum crossing number 10. Thus, I currently have no links to present in this generation.

The Evolutionary Tree for All 21 Unknot/unknot Links with Minimum Crossing Number 8 or Less

Figures 57 - 64 contain all of the 21 unknot/unknot links with minimum crossing number 8 or less. Now, I will present the evolutionary tree that clearly demonstrates all of these parent/child relationships at once:

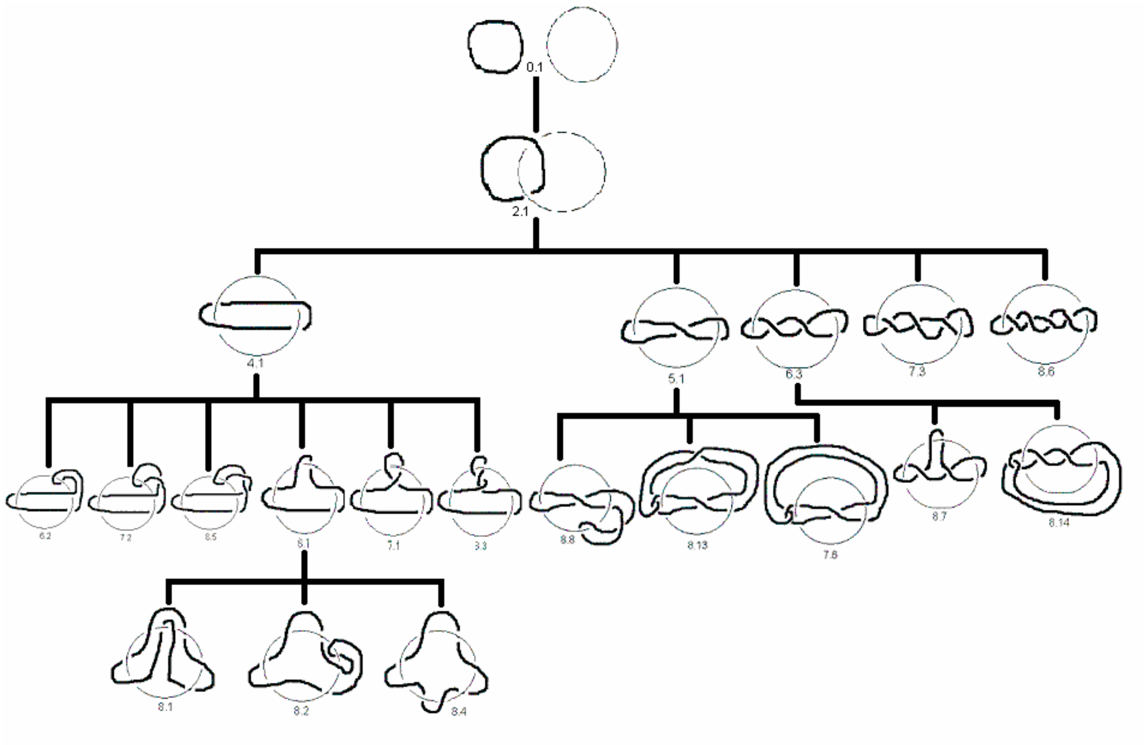


Figure 65

The xth row in this tree contains the links from the xth generation of links. For example, the zeroth row contains the unlink from the zeroth generation. This type of evolutionary tree, which so beautifully sums up the relationship between all 21 unknot/unknot links with minimum crossing number 8 or less, is exactly the structure into which I would like to place all prime knots. I consider the applicability of the concept of construction via self-entanglement to both knots and links as a confirmation of the value of the technique.

Three unresolved challenges regarding knot construction via self-entanglement

How to select the origin and target of a given tentacle?

As previously stated, the premise of this construction technique is that you can construct complex knots from simple knots in a recursive process, using the unknot as the base case. Thus, every knot is a "parent" knot, upon which you can construct new tentacles, resulting in "child" knots. Every time you want to construct a child knot from a parent knot, you have three decisions to make. Decision 1 is selecting a strand from the parent knot, upon which to build the tentacle. I call this the "origin" of the tentacle. Decision 2 is selecting the handedness and length (how many twists) of the tentacle. Decision 3 is deciding *where* to clasp that tentacle. I call this the "target" of the tentacle.

Thus, the three decisions of this construction technique concern the origin, length and handedness, and target of the tentacle. If you wanted to be thorough, you could simply build every possible of tentacle at every possible origin, and clasp it onto every possible target. However, this process would certainly be quite redundant. At this point, I am hoping that as I continue to analyze the origin and target sites in known prime knots, I will find a pattern that I can generalize into simple rules that govern the construction process.

Another difficulty in selecting the origin and target of a tentacle is encoding these decisions within an algorithm that can be implemented by a computer program. As a human, it is easy enough to find origins and targets, but how would a computer make these decisions? This is currently an open question.

How to define an algebra that governs this construction process?

At this point in my research, the concept of construction via self-entanglement is an interesting idea with definite potential in its ability to construct and tabulate knots in a meaningful way. However, I am still lacking a rigorous mathematical structure that governs this construction process. My ultimate goal is to encode the information of this construction process within a mathematical structure, such that any construction process that produces a given knot -- knotX -- will yield the same value.

Specifically, I am hoping to define a value -- for the sake of discussion, let us simply call it the "Self-entanglement number" -- that labels prime knots in the same way that Euler's number labels two-dimensional surfaces. For example, when you partition a given two-dimensional surface with vertices and edges, you can calculate the Euler number for that surface. Because every distinct surface has a distinct Euler number, the partitioning of a surface is a quick and easy way to determine what type of surface you are observing. Likewise, my hope is to define a mathematical structure for the self-entanglement construction technique such that every construction scheme of the same knot will produce the same Self-entanglement number. Just as vertices, edges, and faces are used to calculate Euler's number, I would like to use the origin, length and handedness, and target of each tentacle within a given knot to calculate the Self-entanglement number.

While this is a lofty goal, I believe it is on target with what I need to do to make this construction technique rigorous. I know that many prime knots have more than one construction scheme that produces them. If two given construction scheme yield the same knot, it seems reasonable that an algebra structure that defines those schemes should also yield the same value. If such a mathematical structure exists, then any given projection of any given knot can be analyzed according to the origin, length and handedness, and target of each tentacle within it, and the Self-entanglement number could be easily calculated.

How do I include a prime knot within an evolutionary tree when it has two parents?

Any two distinct knots can converge to a common child. For instance, consider prime knot 9.27 (Refer to Figure 33). It is a trefoil with two single-twist tentacles, one of which is clasped internally, and one of which is clasped externally. Thus, knot 9.27 can be interpreted as either the child of prime knot 6.2 or the child of prime knot 6.3, depending on the order of which the internal and external tentacles of knot 9.27 were constructed.

How should this fact be recorded by the evolutionary tree? Should knot 9.27 be the child of knot 6.2, of knot 6.3, of both, or of the trefoil itself? It seems arbitrary to have the parent of knot 9.27 be only either knot 6.2 or knot 6.3. Also, it would ruin the point of an evolutionary *tree* to allow a single child to have two parents. Finally, it would ruin the assumption that "a parent and child are separated by a *single* tentacle" if I allowed knot 9.27 to be the child of the trefoil.

Furthermore, according to this construction technique, any two unique parent knots will eventually share a common knot in their separate progenies. Much like finding a common denominator for two fractions, given any two distinct parent knots, it is possible to construct tentacles on each such that the resultant knots are identical. Thus, this problem of multi-parenting presents a real challenge. Perhaps I shall have to find a structure other than an evolutionary tree in which to organize prime knots. This problem of multi-parented knots is still an open question.

Four conjectures regarding knot construction via self-entanglement

Herein, I will briefly present four conjectures I have made regarding knot construction via self-entanglement.

Conjecture 1: Given a prime knot with a particular tentacle of length 'x', the lengthening of that particular tentacle to any length greater than 'x' will result in another distinct prime knot.

For example, in many of the preceding Figures (such as Figures 44, 45, 46, 49, and 50), I have presented a row of child knots where the only difference between them is the length of the newly constructed tentacle. I am compelled to assume that I can increase the length of any tentacle indefinitely and still obtain another distinct prime knot.

Conjecture 2: An amphicheiral prime knot is a knot that has a single projection, in which can be seen two distinct ways to construct that knot via self-entanglement, each way containing the same number of tentacles, of equal length, but opposite handedness.

For example, consider the following projection of the amphicheiral prime knot 4.1, shown in Figure 66:

Figure 66

Within this single projection, you can see two distinct ways to construct knot 4.1 via self-entanglement. The first way is shown in Figure 67:

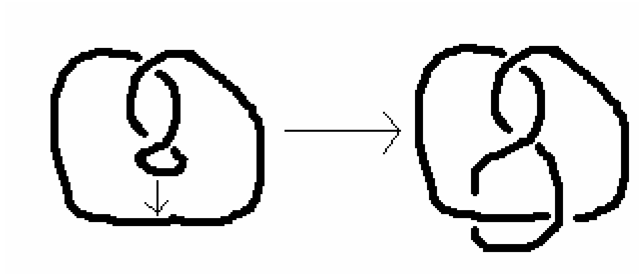


Figure 67

Thus, in Figure 67, prime knot 4.1 is composed on one tentacle: (TLTLCL)

The second way is shown in Figure 68:

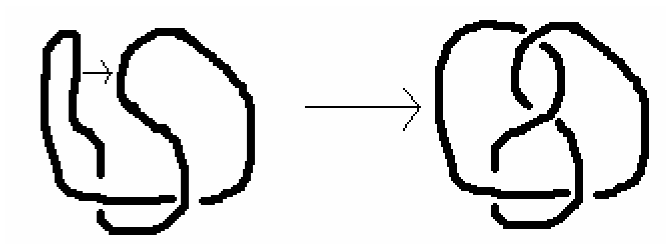


Figure 68

Thus, in Figure 68, prime knot 4.1 is composed of one tentacle: (TRTRCR).

The two construction schemes shown in Figures 67 and 68 utilize tentacles of equal length but opposite handedness. I conjecture that the ability to construct a knot in two such ways -- utilizing tentacles of equal length but opposite handedness -- is unique to amphicheiral knots alone. For example, consider a nonamphicheiral knot, such as the right-handed trefoil in Figure 69:

Figure 69

Viewing this projection, there is only one possible way to construct this knot using self-entanglement, and that construction schema involves the tentacle: (TRCR). There is no way to construct this knot using the tentacle (TLCL).

At present, I have found two amphicheiral knots -- knot 4.1 and knot 8.3 -- that satisfy this conjecture, and no nonamphicheiral knots that satisfy it.

Note: this conjecture requires that a single projection of a knot be reinterpreted in two distinct ways. Perhaps it is the ability to interpret a CL as a TRTR --or, visa versa, a CR as a TLTL -- that permits this reinterpretation of a single projection. For instance, the CL in Figure 67 is the TRTR in Figure 68. As another example, consider the amphicheiral prime knot 8.3:

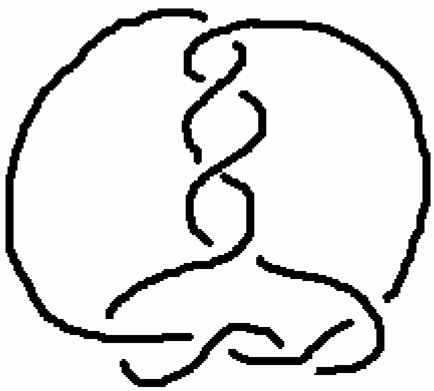

8.3

Figure 70

On the one hand, you can interpret the central vertical tentacle as having 4 left-handed twists and as being clasped onto the bottom of the loop with two left-handed clasps. Or, you can interpret that bottom horizontal tentacle as having 4 right-handed twists and as being turned upward to clasp onto the top of the loop with two right-handed clasps. Thus, knot 8.3 also satisfies this conjecture.

Conjecture #3: There is a way to test whether or not a given knot is invertible by observing particular patterns in self-entanglement.

Though I have no evidence to support this conjecture as of yet, I am hopeful that self-entanglement will give insight into both amphicheirality and invertibility.

Conjecture #4: There is a relationship between the way links are entangled and the way knots are self-entangled.

For instance, compare the first generation knots in Figure 46 with the second generation links in Figure 59. Note the predominant twisted tentacle. Also, compare the second generation knots in Figure 47 with the third generation links in Figure 60. Note the similarity in the internally constructed tentacles. Finally, compare the second generation knots in Figure 48 with the third generation links in Figure 61. Note the similarity in the externally constructed tentacles. While I recognize that these comparisons may seem rather arbitrary, I have a hunch that these similarities are more than just an accident. After all, the concept of entanglement is used in the production of both knots and links. One would expect entanglement to function similarly upon a single loop, two loops, three loops, etc.

A summary of knot construction via self-entanglement

My original goal was to find a way to conceptualize knots in a holistic way that is independent of any particular sensory modality, such as vision. I believe that the concept of self-entanglement satisfies both of these requirements. It is holistic because it takes into account how the loop is entangled with itself, via tentacles, rather than simply keeping count of how many "crossings" there are in a given knot. It is independent of any particular sensory modality because it focuses on the construction of tentacles at unique origins and targets, rather than paying undue attention to the "crossings" of a knot, which are merely an accidental visual perspective of the tentacles. Indeed, once I define an algebra system for this construction technique, there will be no need to make reference to the drawings of knots whatsoever.

For the time being -- with so much work left to do -- I will end this discussion of knot construction via self-entanglement with a concise summary.

1 All prime knots can be constructed in a recursive process of self-entanglement.
2 The unknot is the base case of this recursive process.
3 Every prime knot is a parent knot, upon which a tentacle is constructed, resulting in a child knot.
4 Therefore, all prime knots are related to each other via parent/child relationships.
5 All prime knots can be arranged in an evolutionary tree that reflects these parent/child relationships.
6 Each of the above five statements applies to links as well.

9: DUB'S CONJECTURE REGARDING SELF-ENTANGLEMENT AND MINIMUM CURVATURE

At some point during my research project, I came to realize a clear relationship between knot construction via self-entanglement and minimum curvature. On the one hand, I felt convinced that each distinct knot would have a distinct value of minimum curvature (if only I could calculate it!), with the unknot possessing the least of all such values. On the other hand, I was experimenting with a technique that constructs knots by entangling a loop with itself in various ways. The relationship between the two concepts is this: minimum curvature is simply a *numerical measurement* of the complexity of the self-entanglement of a loop. In other words, my effort to calculate the minimum curvature of a knot was, in a sense, equivalent to my effort to determine the degree of complexity of the self-entanglement of a knot. That said, even though I did not form any definitive conclusions about either knot construction via self-entanglement or minimum curvature, I make the following conjecture as a summary of what I expect to find as I continue my research in the future:

Dub's Conjecture Regarding Self-Entanglement and Minimum Curvature:
Both knots and tentacles have minimum curvature. If a method is ever devised to measure minimum curvature, it will be found that the value of minimum curvature of a given knot is simply the sum of the minimum curvature of the unknot and the minimum curvature of the tentacles contained within the knot.

10: CONCLUDING REMARKS

I began this research project with the goal of figuring out the *essence* of a knot: that property that defines a knot, *and* sets it apart from other distinct knots. By the end of my research project, my conceptions of self-entanglement and minimum curvature (and of knots, in general) were unified into a single structure. Whether I am right or wrong, the following is a description of how I currently conceptualize knots:

Traditionally, a knot is defined as a closed, nonintersecting curve in three-dimensional space. While this definition is technically true -- obviously, all knots are made out of closed, nonintersecting curves in three-dimensional space -- this definition does not actually say anything about what makes a closed, nonintersecting curve in three-dimensional space *knotted*. Defining a knot in this traditional way is like defining an automobile engine as "metal". Certainly, an engine is made of metal, but such a definition says nothing about what makes that "metal" an *engine*. For instance, I use the word "loop" as a shorthand definition of "closed, nonintersecting curved in three-dimensional space". I do not think it is sufficient to say, "a knot is a loop". It would be more accurate to say, "a knot is a *knotted* loop". Thus, I propose that the traditional definition of a knot should be expanded to include some specification about what makes a loop *knotted*. Specifically, making reference to my technique of knot construction via self-entanglement, I now define a knot as a "*self-entangled*, closed, nonintersecting curve in three-dimensional space". I feel this definition more accurately describes what a knot *is*, and it gives an indication of how knots can be studied and tabulated: according to the complexity of their self-entanglement.

REFERENCES

Adams, Colin C. The Knot Book: An Elementary Introduction to the Mathematical Theory of Knots. New York, NY: W. H. Freeman and Company, 2001.

Aneziris, Charilaos N. The Mystery of Knots: Computer Programming for Knot Tabulation. River Edge, NJ: World Scientific, 1999.

Chern, Shiing-Shen. Global Differential Geometry. Washington, D.C.: Mathematical Association of America, 1989.

Gonzalez, Oscar. Global curvature, ideal knots and models of DNA packing. Retrieved September 20, 2002 from the World Wide Web: http://lcvmwww.epfl.ch/curvature.html.

Rawdon, Eric J. & Millett, Kenneth C. (2001). Energy, ropelength, and other physical aspects of equilateral knots. Retrieved September 20, 2002 from the World Wide Web: http://www.mathcs.duq.edu/~rawdon/knotdata.html.

Rolfsen, Dale. Knots and Links. Berkeley, CA: Publish or Perish, 1976.

Trautwein, A. (1995). Harmonic Knots. Retrieved October 3, 2002 from the World Wide Web: http://www2.carthage.edu/~trautwn/.