ON THE EXISTENCE OF A WINNING STRATEGY IN THE $T(4,3)$ KNOTTER VS. UNKNOTTER GAME

By

Timothy Chang

A Thesis submitted to the Duke University
Mathematics Department in partial fulfillment of the
requirements for the degree of

Bachelor of Arts with Honors
in
Mathematics

Date: April 2013.
APPROVAL

As members of the Undergraduate Thesis Evaluation Committee, we certify that we have read the thesis entitled


and recommend that it be accepted as fulfilling the thesis requirement for the degree of Bachelor of Arts in Mathematics with Honors.

________________________________________ Date: April, 2013.
David P. Herzog, Chair

________________________________________ Date: April, 2013.
Christopher Cornwell, Committee Member
Abstract

Here we prove that the “Knotter” in the $T(4,3)$ “Knotter vs. Unknotter Game” has a winning strategy provided she goes second.
**Acknowledgments**

I would like to express my sincere gratitude to James Morrow and the REU program at the University of Washington, sponsored by the National Science Foundation, for giving me the opportunity to pursue my research interests in knot theory. In addition, I would like to thank Allison Henrich of Seattle University for laying the foundational work of this project and for her continued support and assistance throughout. I also thank Christopher Cornwell of Duke University for his expertise in knot theory and constant availability throughout this project. Finally, I would like to thank my research advisor, David Herzog of Duke University, for constantly pushing and challenging me to produce the best work possible.
Mathematically a knot is an embedding of a circle in $\mathbb{R}^3$. However, one should think of a knot as nothing more than a smooth closed curve in $\mathbb{R}^3$ which does not intersect itself. See Figure 1.1 for a few examples of knots visualized using knot diagrams.

There are many interesting questions one can ask about knots. See, for example, the book [Ada09] for an outline of topics. Here, however, the focus is on the more recent development of knot games, a venture begun in 2009 as part of an REU (Research Experience for Undergraduates) program at Williams College. Knot games are as they sound; that is, games played on knots, and naturally one could dream of many ways of playing games on knots. We will be, however, exclusively concerned with one game: The so-called Knotter vs. Unknotter Game.

1.1. **Knotter vs. Unknotter Game.** First recall (see the caption under Figure 1.1), for a fixed knot diagram, all crossing information is given. Suppose now that we do not have any crossing information; that is, we do not know at each crossing which strand goes over or under. The Knotter vs. Unknotter game is played on such a configuration, called a shadow, in which two players, the “Knotter” and the “Unknotter”, take turns resolving each crossing. Like the names suggest, the Knotter’s goal is to produce a genuine knot (i.e. something that cannot be completely untied to a circle) whereas the Unknotter’s object is to yield a knot that can be completely untied.
There are many configurations on which a pair could play the Knotter vs. Unknotter Game. It is thus natural to wonder if either player has a winning strategy for a given shadow. The earliest results to this effect are contained in the work \cite{HMN+11} where shadows are assumed to be those of simple twist knots. Here we will consider a geometry which is a bit more complex. Specifically, we study the Knotter vs. Unknotter Game played on the shadow of the torus knot $T(4,3)$ (see Figure 1.2). We will show, provided the knotter goes second, she has a tractable winning strategy.

The outline of the paper is as follows. In Section 2, we give the necessary mathematical background so that we may proceed with greater ease in Section 3 where we prove the main result. Section 4 provides the reader with some concluding remarks as well as suggestions for further research.

2. NOTATION AND TERMINOLOGY

Here we will be more precise than in the introduction. Recall that we discussed the notion of a knot shadow. Suppose now, for a given shadow, some or none of the crossing information is available. We call such a diagram a pseudoknot. Hence a shadow is a
Pseudoknot but a pseudoknot may not be a shadow. Throughout, we will often refer to various pseudoknots of \( T(4, 3) \). We thus require a clear, consistent method of doing so. We label the crossings of \( T(4, 3) \) as in Figure 2.3 using \( i_k, o_k \) for \( k = 1, 2, 3, 4 \) to denote, respectively, the “inside” and “outside” crossings. To distinguish the way in which each crossing is resolved, i.e. which strand goes over the other, we give it a sign, either + or −, to be explained below using Figure 2.3.

First, start at the base point \( \cdot \) which lies on the strand between \( o_1 \) and \( o_4 \) as in 2.3 and trace along the knot in the direction of the arrow. Note that during a complete trace of the knot, each crossing will be passed twice. If, during the first pass of a given crossing, the strand that is being traced passes over the other strand, then that crossing is deemed positive by putting a + symbol in front of the name of the crossing (e.g. \(+o_1\)). Otherwise, we deem the crossing negative by placing a − symbol in front of the name of the crossing. Consult Figure 2.4.

Using this convention, we shall denote a given pseudodiagram of the knot \( T(4, 3) \) as follows:

\[ [\text{sgn}(i_{j_1}), \text{sgn}(o_{k_1}), \ldots, \text{sgn}(i_{j_l}), \text{sgn}(o_{k_m})] \]

where \( 1 \leq l, m \leq 4, \ j_k, k_n \in \{1, 2, 3, 4\} \) and \( \text{sgn} \) means either + or − depending on how the crossing is resolved. If, for a fixed pseudoknot \( K \) of \( T(4, 3) \), we resolve all remaining crossings, the result of this procedure is a knot diagram which we call a resolution of \( K \). The finite set of all resolutions of \( K \) is denoted by \( \text{res}(K) \).
An essential component of the definition of the Knotter vs. Unknotter game is that of *equivalence of knots*. Thus, we can make precise the phrase “completely untied”. Two arbitrary knots $K_1$ and $K_2$ are called *equivalent* if the knot diagram of $K_1$ can be obtained from the knot diagram of $K_2$ using the three *Reidemeister moves* [Tra83]. We need not dive into the specifics of these moves, but one should think of them as what one could do with the knot physically in hand. A knot is called *trivial* if it is equivalent to the *unknot*, the knot whose knot diagram is a circle. A knot is called *nontrivial* otherwise. Thus the object of the Knotter (respectively, Unknotter) is to pick crossings in such a way so that the resulting knot is non-trivial (respectively, trivial).

Although equivalence is important in determining the triviality of a knot, we will also employ another useful fact; that is, a knot $K$ is trivial if and only if its *mirror image* (i.e. has the same knot diagram except every crossing is resolved in the opposite way) $m(K)$ is trivial. To see this, consider an arbitrary trivial knot $K$ and its mirror image $m(K)$. By taking any sequence of Reidemeister moves that unknots $K$ and performing the mirror image sequence on $m(K)$, we can unknot $m(K)$ as well. A similar argument yields the converse statement.

We now have sufficient notation and terminology, and thus we turn to the proof of the main result.
3. Proof of Main Result

Recall that the goal is to show that the Knotter, if she moves second, has a winning strategy in the $T(4, 3)$ Knotter vs. Unknotter Game. The proof of this splits into two parts. First, we find a sufficient number of complete choices of crossings whose associated knots are non-trivial. Lastly, we then use this, along with a convenient identification of the game with “tic-tac-toe” on the octagon, to show that, regardless of the choices of the Unknotter, the Knotter can always pick one of these non-trivial knots.

3.1. Pseudoknot generation. Consider the pseudoknot

$$K_1 = [-i_1, +o_1, -i_3, +o_4].$$

We show later that every $K_1 \in \text{res}(K_1)$ is non-trivial. First, though, using permissible operations on $K_1$, we generate a list of pseudoknots such that each of its resolutions is non-trivial if and only if every $K_1 \in \text{res}(K_1)$ is non-trivial. Certainly, ambient rotations are permissible. Moreover, applying the argument given in Section 2, we may take mirror images. We, however, still will require a less obvious manipulation, which can be thought of intuitively as turning $K_1 \in \text{res}(K_1)$ inside out. To make this precise, however, we must apply some well-known facts from the intersection of knot and graph theory.
Consider a fixed projection $K$ of an arbitrary knot in the plane. We may associate a graph $g(K)$ to the knot $K$ as follows: Note that the compliment $\mathbb{R}^2 \setminus K$ contains a finite number of disjoint regions, all homeomorphic to a disk, which can be divided into two classes $B$ and $W$ such that an arc emanating from a crossing is a common boundary of a region in $B$ and a region in $W$. We assume that the point at infinity belongs to a region of $W$, and this now completely determines all members of each class $B$ and $W$. Mark the regions belonging to $B$ as $B_1, B_2, \ldots, B_k$. Each region $B_j$ corresponds to a vertex on $g(K)$ and we attach an edge connecting two vertices $P_i, P_j \in g(K)$ if and only if their associated regions have a common crossing on their boundaries. We also include crossing information by placing a $+$ and $-$ on an edge according to convention in Figure 3.5.

In a similar fashion, we can also introduce another graph $g'(K)$, called the dual, which is obtained by instead assuming that the point at infinity now belongs to $B$. It is not hard to see that $g'(K)$ is the same as the graph theoretical dual of $g(K)$.

Using this construction, one can translate Reidemeister moves into so-called fundamental operations on the graphs and show that (see [YK57]).
**Theorem 3.1** (Kinoshita, Yajima ’57). *Every* \( g(K) \) *of a knot* \( K \) *is equivalent to its dual graph* \( g'(K) \)* *by the fundamental operations.*

Therefore, one can establish the equivalence of two knots, \( K \) and \( J \), by showing that \( g(J) \) is equivalent to either \( g(K) \) or \( g'(K) \) by the fundamental operations. We will not need any specific fundamental operation here, as is the reason we do not introduce them. However, we shall use Theorem 3.1 to generate a pseudoknot \( K_2 \) such that every \( K_2 \in \text{res}(K_2) \) is equivalent to some \( K_1 \in \text{res}(K_1) \). Note that

\[
K_2 = [+i_1, -o_1, +i_2, -o_3]
\]

does the trick since, for every \( K_2 \in \text{res}(K_2) \), there exists \( K_1 \in \text{res}(K_1) \) satisfying \( g(K_2) = g'(K_1) \) (see Figure 3.6).

Recalling the initial goal of this subsection, one can apply rotations of \( K_1 \) to see that

\[
K_3 = [+o_1, -i_2, -o_2, +i_4], \quad K_4 = [+i_1, -o_2, +i_3, -o_3], \quad K_5 = [+i_2, -o_3, -i_4, +o_4]
\]

are such that every resolution \( K_i \in \text{res}(K_i), i = 3, 4, 5 \), is non-trivial if and only if every resolution \( K_1 \in \text{res}(K_1) \) is non-trivial. Rotating \( K_2 \), we see that

\[
K_6 = [+i_2, +o_2, -i_3, +o_4], \quad K_7 = [+o_1, -i_3, +o_3, +i_4], \quad K_8 = [+i_1, -o_2, +i_4, -o_4]
\]

and, making use of the mirror image,

\[
K_9 = [+i_1, -o_1, +i_3, -o_4], \quad K_{10} = [-i_1, +o_1, -i_2, +o_3], \quad K_{11} = [-o_1, +i_2, +o_2, -i_4],
\]

\[
K_{12} = [-i_1, +o_2, -i_3, +o_3], \quad K_{13} = [-i_2, +o_3, +i_4, -o_4], \quad K_{14} = [-i_2, -o_2, +i_3, -o_4],
\]

\[
K_{15} = [-o_1, +i_3, -o_3, -i_4], \quad K_{16} = [-i_1, +o_2, -i_4, +o_4]
\]

all share the same property.

We now turn to showing that every \( K_1 \in \text{res}(K_1) \) is non-trivial. To do this, we shall make use of the computer, and we can do this in two ways. The first of which is to
take an arbitrary $K_1 \in \text{res}(K_1)$ and rewrite it in Dowker-Thistlethwaite notation, or DT notation (for reference see [http://en.wikipedia.org/wiki/Dowker_notation](http://en.wikipedia.org/wiki/Dowker_notation)), to obtain the following sequence: $12 \ -14 \ 16 \ \text{sgn}(2) \ \text{sgn}(4) \ 6 \ \text{sgn}(8) \ \text{sgn}(10)$, where $\text{sgn}(i) = i$ or $-i$, depending on how the crossing $i$ is resolved. Using what is known as a knot invariant calculator (see [Indiana University's "KnotFinder"](http://knotfinder.indiana.edu/)), which takes an arbitrary knot $K$ and returns its minimal knot, i.e. $K$ in its most reduced form, we obtain the minimal knot of our arbitrary $K_1 \in \text{res}(K_1)$. Since the minimal knot does not correspond to the unknot, which we can verify using any knot atlas (e.g. "The Rolfsen Knot Table"), our $K_1 \in \text{res}(K_1)$ is in fact knotted. By repeating this process for every $K \in \text{res}(K_1)$, we see that every $K_1 \in \text{res}(K_1)$ is non-trivial. Note that this now allows to conclude:

**Lemma 3.2.** For all $i = 1, 2, \ldots, 16$, every $K_i \in \text{res}(K_i)$ is non-trivial.

If one does not like the “computer” way, one can explicitly compute the Alexander polynomial of each $K_1 \in \text{res}(K_i)$ and see that all are not equal to 1, also giving Lemma 3.2.

We now know that there are sixteen pseudoknots $K_1, K_2, \ldots, K_{16}$ with the property that each of its resolutions is a non-trivial knot. In the context of the game, this means that it is enough for the Knotter to resolve four crossings as prescribed in the pseudoknots $K_1, K_2, \ldots, K_{16}$ to win. The next part of the proof will demonstrate, provided the Knotter moves second, picking the crossings in this way is always possible.

### 3.2. Map to the Octagon

In order to paint a more intuitive picture of the game, we map the $T(4,3)$ knot shadow to a regular, edgeless octagon. This allows us to visualize the game as it is played more clearly and avoid the unnecessary complexities of the various psuedoknots. Each vertex of the octagon represents an unresolved crossing of the shadow with the N, E, S, W vertices representing the outside crossings, $o_1, o_2, o_3, o_4$ respectively, and the NE, NW, SW, SE vertices representing the inside crossings, $i_1, i_2, i_3, i_4$ respectively. Such a mapping should look like the rightmost picture in Figure 3.7.

As crossings are resolved with each turn, we mark the vertex corresponding to the crossing that is moved upon using the following convention: "O" if + and "X" if −. When
translating to the octagon, the sixteen pseudoknots $K_1, \ldots, K_{16}$ are as in Figure 3.8 and Figure 3.9.

The goal of the Knotter is to thus produce, in the process of the game, one of the sixteen patterns regardless of the sequential choices of the Unknotter. Let us now turn to proving that, provided the Knotter moves second, this is always possible.

3.3. Base Pairs. To simplify our discussion, we introduce the notion of a base pair. By observation of the sixteen pseudoknots above, we see that each contains two vertices opposite each other on the octagon. It is this pair of vertices of the pseudoknot on the octagon that we refer to as the base pair. Note that a base pair is not any arbitrary resolution of two vertices opposite each other. These vertices must be resolved in the correct manner so as to partially complete a pseudoknot. Furthermore, we also observe that any base pair is shared by two distinct pseudoknots from $K_1, \ldots, K_{16}$. Consult Figure 3.10.

Now superimpose two pseudoknots with the same base pair onto the octagon and notice that the four non-base pair vertices form two new base pairs. Each of these new base pairs are, by definition, a part of two different pseudoknots. The crucial observation is that by superimposing these four new pseudoknots onto the original two, the six base pair vertices are all resolved in the same way for all six pseudoknots. It is not difficult to see that this property holds given any base pair. Consult Figures 3.11, 3.12.
Hence, given a base pair, there exists a “pre-determined” way of resolving the four adjacent vertices to obtain two new base pairs, resulting in three adjacent base pairs. In addition, the six pseudoknots of which the three base pairs are a part, all share the same crossing information with respect to the base pair vertices. The significance of this property will become apparent in the following lemma:
Lemma 3.3. The completion of two adjacent base pairs within seven moves guarantees the completion of one of $K_1, \ldots, K_{16}$, provided the Knotter moves second.

Proof. Suppose we have two adjacent base pairs. Recall by definition that two distinct pseudoknots $K_1, \ldots, K_{16}$ share the same base pair, which implies that there are currently four partially complete pseudoknots. From our discussion above, we also know that
these four pseudoknots share the same crossing information with respect to all vertices upon which two or more pseudoknots overlap, i.e. any vertex involved in two or more pseudoknots is resolved in the same way in all of the involved pseudoknots. It is not difficult to see that each of these four partially complete pseudoknots require only one more vertex be resolved correctly for completion. In addition, this final vertex is unique to each of the four pseudoknots. Let $m$ represent the number of moves that have occurred:

- If $m = 4$, then no other vertices have been moved on apart from the two adjacent base pairs. This implies that four vertices remain unresolved, i.e. there are four different partially complete pseudoknots that are one move away from completion. Since the Unknotter cannot move on all four crossings in one turn, she can only negate one of the possible four. This leaves the Knotter to complete one of three partially complete pseudoknots.

- If $m = 5$, then only one vertex has been moved on apart from the two adjacent base pairs, leaving three vertices unresolved. Since it is the Knotter’s turn, she is left to complete one of three remaining partially complete pseudoknots.
- If $m = 6$, then there still remain two unresolved vertices, and thus, two partially complete pseudoknots remain. Again, the Unknotter can only negate one of these, leaving the Knotter to complete the other.
• If \( m = 7 \), then it is the Knotter’s turn, on which she is free to complete the last remaining partially complete pseudoknot.

Hence, we see that provided the Knotter moves second, the completion of two adjacent base pairs results in the eventual completion of one of \( K_1, \ldots, K_{16} \). □

3.4. The Knotter Wins. We have effectively reduced the task of completing a pseudoknot to obtaining two adjacent base pairs. Let us demonstrate via a generalized walkthrough of the game that this is always possible, provided the Knotter moves second:

**Move 1:** The Unknotter moves on any vertex.

**Move 2:** Given the Unknotter’s move, the Knotter completes the base pair.

At this point, the game can proceed in one of two ways, depending on whether or not the Unknotter moves on a vertex adjacent to the base pair.

- Suppose the Unknotter moves on a vertex adjacent to the base pair:
  - **Move 3:** The Unknotter moves on any one of four vertices adjacent to the base pair.
  - **Move 4:** The Knotter moves on the vertex that completes the trapezoid (exclusive definition). (*)

- Suppose the Unknotter moves on a vertex not adjacent to the base pair:
  - **Move 3:** The Unknotter moves on one of two vertices not adjacent to the base pair.
  - **Move 4:** The Knotter moves on any vertex adjacent to the base pair.

- **Move 5:** The Unknotter is forced to move on the vertex that would complete the pseudoknot produced by moves 1, 2, and 4.
- **Move 6:** The Knotter completes the second base pair (from move 4). This results in two adjacent base pairs.

(*) Note: There are two pairs of vertices opposite each other on the octagon that are adjacent to the base pair. Assuming that the Unknotter moved in a way that would not result in her immediate loss, the Knotter should not move on the same pair that the Unknotter
moved on. Thus, the Knotter should move on the other pair of opposite vertices, and has one of two choices. By moving on the vertex that completes the trapezoid, the Knotter has obtained a partially complete pseudoknot that is only one vertex away from completion. Thus, the Unknotter’s next move is forced: she must prevent that pseudoknot from being completed. In doing so however, she leaves the Knotter free to complete the base pair on her next turn. If the Knotter had chosen not to move on the vertex that completes the pseudoknot, it is easy to see that the completion of two adjacent base pairs is not guaranteed.

Hence, if the Knotter moves second, she is always able to complete two adjacent base pairs in six or fewer moves. By Lemma 2.3, this results in the completion of one of the sixteen pseudoknots generated above. Since by Lemma 2.2, every \( K_i \in \text{res}(K_i) \) is knotted, the resultant knot will be knotted by the end of the game. Therefore, the Knotter wins.

4. The Strategy Demonstrated On the Knot

Here, we play the Knotter vs. Unknotter Game for fun, demonstrating our winning strategy. We will take on the role of Knotter moving second and make the moves necessary to win. Keep in mind the sixteen pseudoknots we have generated, as they are essential in determining how to resolve the necessary crossings.

Move 1: The Unknotter moves on any crossing.

Move 2: Move on the crossing opposite the Unknotter’s crossing. Two crossings are opposite one other if they are either both inside crossings or outside crossings and are not adjacent to each other. Resolve this crossing in the way that contributes to the completion of one or more of the sixteen pseudodiagrams. For example, if the Unknotter moved \(+o_1\), we would move \(+o_3\) since \( K_7 = [+o_1, -i_3, +o_3, +i_4] \) and \( K_{10} = [-i_1, +o_1, -i_2, +o_3] \).

Move 3: The Unknotter moves on any crossing.

Move 4:

- Suppose move 1 was on an outside crossing.
  - If move 3 was also on an outside crossing, move on any inside crossing.
– If move 3 was on an inside crossing, move on the inside crossing such that if the knot shadow were split along the line intersecting the crossings of moves 1 and 2, is on the same side as move 3.

• Suppose move 1 was on an inside crossing.

  – If move 3 was also on an inside crossing, move on any outside crossing.
  
  – If move 3 was on an outside crossing, move on the outside crossing such that if the knot shadow were split along the line intersecting the crossings of moves 1 and 2, is on the same side as move 3.

Upon selecting the proper crossing, resolve it such that it contributes to the completion of one of the pseudodiagrams involving the crossings of moves 1 and 2, but not move 3. Continuing in the above example, suppose the Unknotter moved $+i_1$. Split the knot shadow along the line intersecting the crossings of moves 1 and 2, or $o_1$ and $o_3$, respectively. Given that move 1, $+o_1$, is on an outside crossing and move 3, $+i_1$, is on an inside crossing, we would move on the inside crossing on the same side of the line as move 3. That crossing is $i_4$. The way in which we resolve crossing $i_4$ is such that it contributes to the completion of one of the pseudodiagrams containing $o_1$ and $o_3$, but not $i_1$. $K_7$ is such a pseudodiagram, and it tells us to move $+i_4$.

**Move 5**: The Unknotter is forced to move on a particular crossing as to prevent the completion of one of the pseudodiagrams involving the crossings of moves 1, 2, and 4.

**Move 6**: Move on the crossing opposite move 4. Resolve this crossing in the way that contributes to the completion of one of the pseudodiagrams involving the crossings of moves 1 and 4. Back to our example, since $K_7$ was one crossing away from completion, the Unknotter would be forced to move $+i_3$. Given that move 4 was $+i_4$, we would move on the opposite crossing, namely $i_2$. Since moves 1 and 4 were $+o_1$ and $+i_4$ respectively, and $+o_1, +i_4 \in K_3 = [+o_1, -i_2, -o_2, +i_4]$, we would move $-i_2$ as present in $K_3$.

**Move 7**: The Unknotter moves on any crossing.

**Move 8**: Move on the last crossing in such a way that completes any of the remaining sixteen pseudodiagrams, which we know exists by our proof and will involve the crossings
of moves 2, 4, and 6. Since all resolutions of the sixteen pseudodiagrams are nontrivial, the result must be knotted. Completing our example, suppose the Unknotted moved \( +o_2 \) as to prevent the completion of \( K_3 = [+o_1, -i_2, -o_2, +i_4] \). We would then move on crossing \( o_4 \) according to one of the sixteen pseudodiagrams involving \( +o_3, +i_4, -i_2 \). Since such a pseudodiagram is \( K_{13} = [-i_2, +o_3, +i_4, -o_4] \), we would move \( -o_4 \), completing \( K_{13} \) and effectively producing a genuine knot. We win!

5. Conclusion

In summary, we have shown that provided the Knotter moves second, she has a winning strategy for the Knotting/Unknotting game when played on the \( T(4, 3) \) knot shadow. All that is required is the successful completion of one of sixteen pseudoknots shown above, each of which is composed of only four resolved crossings. The resolution of the shadow in a way that produces one of these pseudoknots always results in a nontrivial knot, regardless of how the remaining precrossings are resolved. Further, such a resolution is possible if the Knotter moves second. The natural question that arises now is whether a winning strategy exists for other torus knots, especially the family of \( T(n, 3) \) knots for \( n \in \mathbb{N} \).

References


