An Exploration of Connect Sums of Knots Using the Trip Matrix

Spring 2023 Senior Thesis

Emerson Worrell

April 19th, 2023

Abstract

We utilize the trip matrix method of calculating the Jones Polynomial to give an alternative proof that the Jones Polynomial is multiplicative under connect sums. We then use the structure of the trip matrix itself as a method to determine if a given knot diagram with minimal crossing number representation is prime or composite. Finally, we briefly explore the effect of Reidemeister moves on the structure of the trip matrix.

1 Introduction and Motivation

The goal of this thesis was to perform an in-depth exploration of the trip matrix method of computing the Jones Polynomial. The eternal goal of knot theory is to determine when two knots are distinct and when they are the same. Knot invariants, such as the Jones Polynomial, are tools we use to answer this question. By learning more about how these invariants behave and by creating new invariants, we can gain a better understanding about the nature of knots.

The trip matrix method was created in 1993 by Louis Zulli [6] and provides a way of computing the Jones Polynomial for a knot that requires only linear algebra. The encapsulation of so much information in a matrix over $\mathbb{Z}_2$ provides an interesting opportunity to see what other tasks the trip matrix can be used to perform. We initially used the trip matrix to provide an alternative proof that the Jones Polynomial is multiplicative over connect sums [2]. This process revealed certain patterns in the structures of these matrices; we explored these patterns to provide a method for determining if and when a knot is composite or prime.

In Section 2 we provide the necessary background information needed to understand our work and then in Section 3, we give results on the structure of trip matrices of composite knots. In Section 4 we provide our proof of the multiplicative nature of the Jones Polynomial. In Section 5 we discuss how the trip matrix can be used to determine when a given knot is prime or composite. Finally in Section 6, we delve into how Reidemeister moves affect the trip matrix.

The work we did was fruitful, but many questions arose that we either lacked the time or the ability to answer, which we discuss in Section 7.

2 Background Information

We begin with an explanation of the necessary terminology, tools, and concepts needed to understand our work. This section consists of a basic introduction to knots, knot invariants, connect sums, and the trip matrix. For more information, see [1].

2.1 Basic Knot Theory

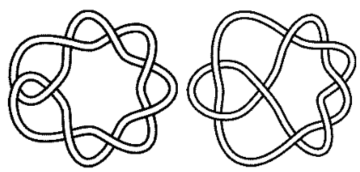
Definition 2.1 (Knot)**.** A *knot* is an embedding of S^1 in $\mathbb{R}^3$. More intuitively, it is some string that has been twisted and wrapped around itself in some manner, and then closed so that there are no loose ends.

Definition 2.2 (Knot Equivalence)**.** Two knots are *equivalent* if one knot can be deformed into the other without cutting the knot or allowing strands to pass through each other. This type of deformation is known as an *ambient isotopy.*

Definition 2.3 (Knot Diagram)**.** A *knot diagram* is a representation of a knot in a 2 dimensional plane. In a knot diagram, there are *crossings* at which exactly two strands of the knot cross over one another. which is represented by a break in the understrand.

A given knot has infinitely many diagrams, although it is common to focus on the simplest of these with the smallest number of crossings possible. See Figure 1a for examples of knot diagrams, and Figure 1b for multiple diagrams for the same knot.

Note 1. There are a number of different ways that knots are classified. In our paper we use the Alexander-Briggs-Rolfsen classification to refer to specific examples of knots (see Appendix A). We also use the colloquial names for certain well-known, simple knots.

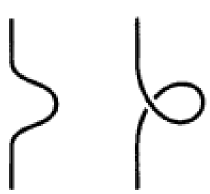


(a) Diagrams for two distinct knots [5]

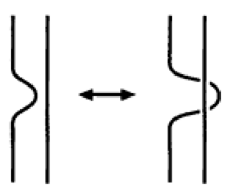
(b) Different diagrams for the same knot [1]

Figure 1: Different knots have different diagrams. But the same knot can have many different diagrams too.

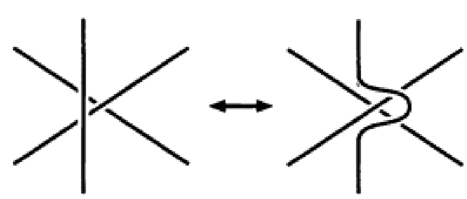
Definition 2.4 (Reidemeister Moves and Planar Isotopies)**.** A *Reidemeister move* is a manipulation of a knot diagram that changes its appearance without cutting or changing the knot itself. There are three types of Reidemeister moves, shown in Figures 2a, 2b, and 2c. Additionally, there exist *planar isotopies*, which are deformations of the projection plane itself, which in turns morphs the knot diagram but does not involve changing crossings or sliding strands over or under others. Figure 3 shows a planar isotopy of a knot diagram.



(a) Type 1 Reidemeister Move



(b) Type 2 Reidemeister Move



(c) Type 3 Reidemeister Move

Figure 2: The three Reidemeister moves [1]

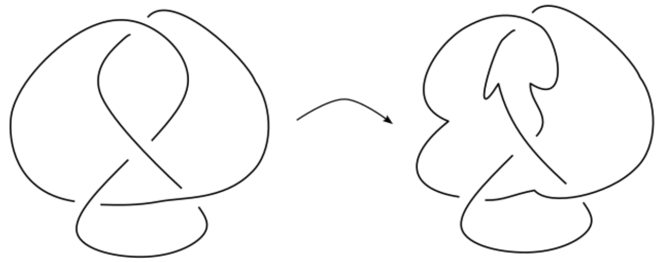


Figure 3: An Example of Planar Isotopy. Notice how the crossings are unchanged, but the strands have been warped and curved.

Theorem 2.5 (Reidemeister [4])**.** The following are both true:

1. If there are two different diagram representations of the same knot, then there exists a sequence of Reidemeister moves and planar isotopies to take one diagram to the other.

2. Two knots are equivalent in 3-space if and only if their knot diagrams are equivalent under Reidemeister moves and planar isotopies.

This theorem is very powerful and will be important later on. Most notably it means that if we can deform one knot in some manner in 3-space, then there exists a sequence of moves in the plane that also show the diagrams are equivalent.

Definition 2.6 (Crossing Number)**.** The *crossing number* of a knot K, denoted $c(K)$, is the smallest number of crossings with which a knot can be represented in a knot diagram.

Every knot has a crossing number and this can be used as a knot invariant (see Definition 2.7). However, finding cross number for knots is in general a difficult problem.

Definition 2.7 (Knot Invariants)**.** A *knot invariant* is a mathematical object that remains unchanged by Reidemeister moves and planar isotopies.

If two knot diagrams are different representations of the same knot, then their invariants will be the same. Distinct knots on the other hand often yield different values for a given invariant, but sometimes yield the same value. Therefore, we use invariants to determine when two knot diagrams represent distinct knots.

Our paper focuses on an operation on knots called the connect sum. To form the connect sum, take two knots, K_1 and K_2. Cut each of these knots in one location away from a crossing, leaving 4 loose ends. Attach these together in pairs such that each loose end from K_1 is now connected to an end from K_2 and no new crossings are introduced. The resulting knot is denoted $K_1 \# K_2$, and is known as a *composite knot*.

All knots are either

- Composite: the result of some connect sum of multiple non-trivial knots.
- Prime: cannot be broken down into the connect sum of two non-trivial knots.
- The unknot, also known as the trivial knot. This is simply a circle in the plane.

Figure 4 below is an example of how two knots may be connect-summed together:

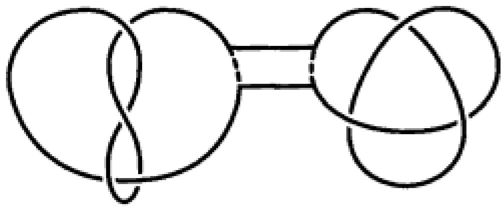


Figure 4: Connect-summing two knots [1]

2.2 The Trip Matrix and The Jones Polynomial

This paper focuses on a knot invariant known as the Jones Polynomial. In particular, we explore a specific method of calculating the Jones Polynomial using a tool known as the trip matrix. As previously mentioned, this method was first introduced by Zulli [6]. The trip matrix is constructed from a knot diagram as follows:

1. Label the n crossings of the knot diagram 1 through n in any order.
2. At each overcrossing, choose a direction along the knot at random and draw an arrow pointing in that direction. We refer to the overcrossing arrow at crossing i by i^+.
3. At each undercrossing, draw an arrow such that the undercrossing arrow is pointing counterclockwise from the corresponding overcrossing arrow. We refer to the undercrossing arrow at crossing i by i^-.
4. Construct an $n \times n$ matrix with entries in $\mathbb{Z}_2$ in the following manner:
    (a) For entry (1,1), follow the overcrossing arrow from crossing 1 until you return to crossing 1 at the undercrossing arrow. If the undercrossing arrow "leads you on", the entry in $(1, 1)$ is a zero. If it "pushes you back", the entry is a one. Repeat this process at each crossing $i \in \{2, 3, \ldots, n\}$ for diagonal entries (i, i).
    (b) The matrix is inherently symmetric, so $(j, k) = (k, j)$.

(c) For the entries off diagonals (entries (j,k) where $j \neq k$), we similarly begin at the overcrossing arrow of crossing j. We then follow the path from here until we reach the undercrossing arrow of crossing j. The number of times modulo 2 that this path goes through crossing i is the value placed in (j,k). Note that because the matrix is symmetric, we could start at crossing i and count the number of times j is crossed to yield the same result.

We refer to the trip matrix for a given knot K by T_K. The resulting matrix can be used to calculate the Jones Polynomial.

Example 1. Trip Matrix
Let's walk through an example of this process, since words on a page can only explain so much. Consider Figure 5, which is the figure 8 knot after numbering the crossings and adding the overcrossing arrows in any order. We choose the arrows of the overcrossing arrows at random and the undercrossing arrows are all pointed counterclockwise from their overcrossing counterparts. The overcrossing arrows are in red with the undercrossing arrows in blue, which we maintain as the standard throughout this paper. Now, lets compute the trip matrix T_K.

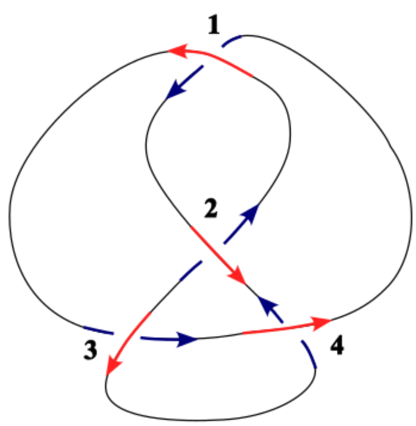


Figure 5: The Figure 8 Knot.

T_K will be a 4×4 matrix. Let's compute the diagonal entries first. Starting at crossing 1, we follow the overcrossing arrow, and when we return to the undercrossing arrow 1^-, it leads in the direction we are currently travelling, so entry $(1,1)$ is a zero. The same is true for crossing 2, so $(2,2)$ is a zero as well. For crossings 3 and 4, however, the undercrossing arrows lead back the way we came, so these entries are both ones.

Now for the other entries. The path from crossing 1 to itself goes through crossing 3, crossing 4, and then arrives back at 1. So entries $(1,2)$, $(1,3)$, and $(1,4)$ are zero, one,

one respectively. Since this matrix is symmetric, $(2,1)$, $(3,1)$, and $(4,1)$ can also be filled in with these same values. For crossing 2, we can ignore any interactions with crossing 1 since both $(1,2)$ and $(2,1)$ have been filled. Our path takes us through crossing 4 and 3 once each, so $(2,3) = (3,2)$ and $(2,4) = (4,2)$ are all ones. Finally, we start at crossing 3 and only consider how it interacts with crossing 4. This path takes it through 4 twice, and since we are operating modulo 2, entry $(3,4) = (4,3)$ is a zero. Our resulting complete trip matrix is seen below:

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Definition 2.8 (Trip Matrix State)**.** A knot with n crossings will have 2^n states. Each state is a string of A's and B's, each with A or B corresponding to one of two possible states at each crossing. We denote a specific state of a knot K as S, and the set of all possible sets for K as $\mathcal{S}(K)$. The trip matrix generated by the method above corresponds to the state containing all A's. In order to get a different state, change the entry along the diagonal in row i from a 0 to a 1 (or vice versa) if the corresponding letter in that position of the string is a B, which we refer to as toggling between states. The matrix for K when toggled to a state S is denoted by T_{K_S}. The number of A's and B's in a given state can not be determined exclusively from the trip matrix itself. However, they are used in the calculation of the Jones Polynomial and can be kept track of separately. These are denoted $A(S)$ and $B(S)$ respectively for a given state S.

Definition 2.9 (Writhe)**.** We define the *writhe* of a knot K (denoted $w(K)$) as the total number of ones along the diagonal minus the total number of zeroes along the diagonal in the trip matrix T_K

Now that we have defined these terms, we can finally use them to define the Jones Polynomial itself:

Definition 2.10 (Jones Polynomial [6])**.** The *Jones Polynomial* for a knot K is defined as the following:

$$V_K = (-t^{\frac{3}{4}})^{w(K)} \sum_{S \in \mathcal{S}(K)} t^{-\frac{1}{4}^{A(S)}} t^{\frac{1}{4}^{B(S)}} (-t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{nul(T_{K_S})}$$

Theorem 2.11 (Jones [2])**.** The Jones Polynomial is a knot invariant.

3 The Structure of Trip Matrices of Composite Knots

We now explore how the trip matrix behaves for composite knots. For the remainder of this section, as well as Sections 4 and 5, we assume that any knot diagrams are representations of knots with their minimal crossing number. That is to say, it is impossible to reduce the number of crossings any further with Reidemeister moves. We make this assumption since it will not change the Jones Polynomial, it will make the trip matrix smaller and therefore easier to deal with, and it is possible because every knot has a representation with its minimal crossing number. Using the smallest diagram and its corresponding trip matrix gives us the clearest picture of the behavior of the knot without any extraneous crossings. Section 6 discusses some of the effects Reidemeister moves and extraneous crossings have on the trip matrix.

Before we continue, there is also one important fact we must address. There is an important open question regarding the crossing number of connect sums. Specifically, it is unknown whether crossing number is preserved under connect sums, i.e., does $c(K_1 \# K_2) = c(K_1) + c(K_2)$ for all possible knots? This conjecture has been proven true for certain classes of knots, namely, torus and alternating knots, conjectured to be true for all knots, and a counterexample has yet to be found [3]. However, this has yet to be proven true for all knots and it is essential to our proofs that it is true. Therefore we assume from this point forward that every claim made only holds for those knots for which crossing number is preserved under connect sums. If at some point in the future this conjecture is proven true, then the following will be true for all knots. If it is proven false, then hopefully it becomes clear for which knots it is and isn't true, and we can restrict our results to those classes of knots for which it is true.

3.1 A Composite Knot Yields a Block Matrix Structure

In this paper, we define a block matrix as an $n \times n$ matrix that is formed by placing smaller $m \times m$ matrices along the diagonal. The remaining entries not in these submatrices are all zeroes. Below is an example of what a block matrix may look like.

$$\begin{bmatrix} x & x & x & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ x & x & x & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ x & x & x & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & x & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & x & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & x & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & x & x & x \end{bmatrix}$$

Note that these blocks can be any size (although we show later that trip matrices of knots with minimal crossing representation must have block size at least 3). Additionally, each x represents either one or zero.

Theorem 3.1. If a given knot is composite, then its trip matrix must have a block matrix structure.

Proof. Let K be a composite knot formed by the connect sum of prime knots K_1 through K_n with crossing numbers $m_1, m_2, \dots, m_n$. Without loss of generality, the connect sum K must have been constructed by first taking K_1 and connect summing K_2 to it in some manner. Then K_3 would be added to this new composite knot $K_1 \# K_2$, continuing all the way through K_n. Now, because we can label the crossings of K in any manner we choose, we can choose to label K such that the first m_1 crossings correspond to the m_1 crossings from K_1, crossings $m_1 + 1$ through $m_1 + m_2$ correspond to the m_2 crossings from K_2, and so on. We can then choose overcrossing arrows in any manner. Consider forming the trip matrix, beginning with row (and column) 1. We begin at crossing 1 and travel in the direction of the overcrossing arrow. There are two possibilities from this point: either this path leaves the section of K corresponding to K_1 before it reaches crossing 1 again, or it does not. If it does not, then the entries in row 1 beyond column m_1 will all be zeroes since the path never goes through any of those crossings. If it does enter a section of K corresponding to a component other than K_1, then it must necessarily go through the entirety of that other component before returning to crossing 1. If this were not the case, then this path must have traveled through part of K_1, entered part of a different component (say K_2 without loss of generality), and then returned to crossing 1 before touching the remainder of K_2. But then K_2 consists of two disjoint components of K, meaning it was actually two distinct components. But we supposed K_2 was a prime knot and so this is a contradiction.

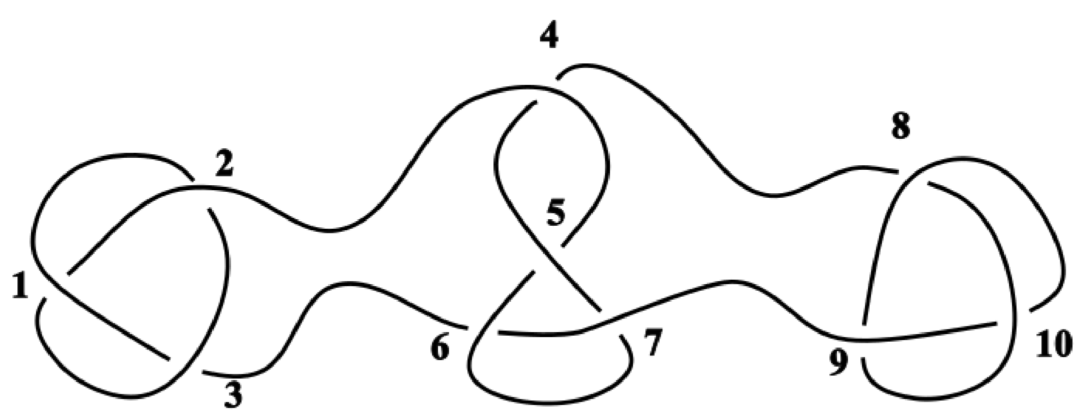
It is also worth noting that the path from crossing 1 might run through more than one other component of K before returning to crossing 1. Example 2 demonstrates this possibility. After entering K_2, the path enters and runs through the entirety of K_3 (therefore passing through every crossing in K_3 exactly twice) as well before returning to K_2 and eventually to K_1. In any case, the path from crossing 1 to itself must either pass through each other component of K completely or not at all. This means that beyond column m_1, all the entries in row 1 must be 0.

Additionally, if the path from crossing 1 to itself does enter a different component, when it inevitably returns to K_1, it will return in the exact same spot and direction from where it left. As a result the path relative to the other crossings in K_1 is the exact same as if it were an isolated knot, and so the entries corresponding to crossing 1 from K_1 are identical to those in the K_1 trip matrix.

The arguments above hold for every crossing in K. The result is a block matrix where each block corresponding to component K_i is an $m_i \times m_i$ block along the diagonal corresponding to the trip matrix for K_i on its own, with the remainder of entries in those rows and columns being 0. □

In Example 2 we provide an example of Theorem 3.1:

Example 2. Consider the knot K with diagram shown below and its corresponding trip matrix. Notice crossings are labelled in a way that respects the "obvious" components.



$$T_K = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

3.2 Properties Preserved In Trip Matrices of Connect Sums

Because the trip matrix of a composite knot is simply made up of smaller matrices of its component knots in a block matrix, it would be natural to assume that many of the properties of the smaller matrices are preserved in the large, composite matrix. In this subsection we show that all of the important information pertaining to the Jones Polynomial is preserved.

Lemma 3.2. A trip matrix for a knot with minimal crossing representation cannot have a column of all zeroes.

Proof. Suppose by way of contradiction that knot K, represented with minimal crossings, has a column of all zeroes in its trip matrix. Call this column i. Every other crossing must be passed through either twice or zero times on the path from i^+ to i^-, and also the undercrossing arrow on i^- must lead the path forward. We can divide the other crossings into two sets: those that are met twice along this path and those that are not met at all, which we refer to as sets A and B respectively. We construct a knot that fits this criteria, shown in Figure 6. Begin by placing crossing i, along with the overcrossing arrow and the undercrossing arrow. Then, all the crossings from set A must lie between i^+ and i^-, while all the crossings from set B must lie on the "other side" of this knot. This results in a knot with two clear "sections" connected by a half twist, which is crossing i. These sections must be disjoint; if they weren't, then the path from i^+ to i^- would necessarily pass through section B and result in crossings that are only met once along this path, generating ones in column i, which we know don't exist.

If we visualize this knot in 3-space, it is clear that this half-twist can be undone by simply twisting the entirety of section B, removing crossing i in the process. Since this can be removed in 3-space, there must be some sequence of Reidemeister moves that can be performed to remove crossing i in the plane as well by Theorem 2.5. Therefore this crossing is extraneous and contradicts our claim that K has been represented with minimal crossing number. □

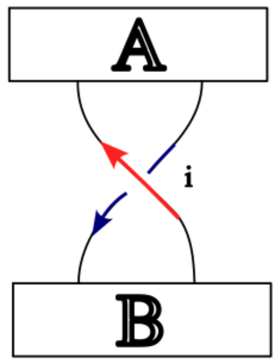


Figure 6: The knot structure resulting from column i having all zeroes. Boxes A and B represent arbitrary knot sections.

Theorem 3.3. Let T be the trip matrix for composite knot K, and let $T_1, \ldots T_n$ be the respective blocks corresponding to component knots $K_1, \ldots K_n$ which form K. Then $nul(T) = nul(T_1) + \cdots + nul(T_n)$.

Proof. Let K be a composite knot formed from prime knots K_1 through K_n, and let T_K and T_1 through T_n be the corresponding trip matrices for the composite knot and its prime components, respectively. Consider each of the trip matrices and their respective blocks in T. A given block will be a $m_i \times m_i$ matrix; consider in addition to this each of the columns of T that this block occupies. Every row excluding the k rows in the block will be exclusively zeroes since T is a block matrix.

The additional rows containing all zeroes have no effect on the dimension of the column space nor the null space. The dimension of the null space of that subset of the columns of T will always be the same as the dimension of that block as a $k \times k$ matrix. The only concern here is a column corresponding to a different block that can be written as a linear combination of columns from this block. However, this is only possible if such a column contained all zeroes; if it had ones, then it would have ones in a row which columns from every other block would only have zeroes due to the structure of the block matrix. Naturally, no linear combination of zeroes can yield a one.

If such a column had all zeroes, then it would be possible for some linear combination of the columns from one block to give us a column of all zeroes from another block (since we are working modulo 2, this would most likely be two identical columns summed together) thereby breaking this transitivity. However, Lemma 3.2 shows that this situation is impossible. Therefore, nullity is transitive over block trip matrices as claimed. □

Example 3. Here is an example of the previous theorem:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

On the left is a 3×3 matrix corresponding to some knot K_1, while on the right, there are 3 additional rows of all zeroes, which exist when considering this matrix as a block in a larger, 6×6 composite matrix. The last 3 rows of the matrix on the right have no effect on the null space, because they are all zeroes.

Theorem 3.4. Let T be the trip matrix for composite knot K, and let $T_1, \ldots T_n$ be the respective blocks corresponding to component knots $K_1, \ldots K_n$ which form K. Then $w(T) = w(T_1) + \cdots + w(T_n)$.

Proof. Let $o(T)$ and $z(T)$ denote the number of ones and zeroes along the diagonal of a given trip matrix T. By definition, $w(T) = o(T) - z(T)$. Also, $o(T) = o(T_1) + \cdots + o(T_n)$ and $z(T) = z(T_1) + \cdots + z(T_n)$ by Theorem 3.1. Then $w(T) = o(T_1) + \cdots + o(T_n) - (z(T_1) + \cdots + z(T_n)) = o(T_1) - z(T_1) + \ldots o(T_n) - z(T_n) = w(T_1) + \cdots + w(T_n)$. □

Theorem 3.5. Let T be the trip matrix for composite knot K, and let $T_1, \ldots T_n$ be the respective blocks corresponding to component knots $K_1, \ldots K_n$ which form K. Then for a given state S, $A(T) = A(T_1) + \cdots + A(T_n)$ and similarly, $B(T) = B(T_1) + \cdots + B(T_n)$.

Proof. A state for an m-crossing knot K is by definition a string of A's and B's of length m. We can break this word up into smaller words, each of which corresponds to the component knots $K_1 \ldots K_n$. Naturally, the number of A's and B's in the large string will be the sum of the number of A's and B's in each of the smaller strings. □

Note 2. While the previous 3 theorems only reference trip matrices, the results extend to any toggled trip matrix as well (and in fact, any block matrix over $\mathbb{Z}_2$).

4 The Jones Polynomial is Multiplicative Over Connect Sums

With the theorems from Section 3, we can now prove our first major result, first proved in [2].

Theorem 4.1. The Jones Polynomial is multiplicative over connect sums. That is, if $K = K_1 \# K_2 \# \cdots \# K_n$,

$$V_K = \prod_{i=1}^{n} V_{K_i}$$

.

Proof. Let K be a composite knot that is formed via connect sums by the prime knots $K_1 \dots K_n$. Let K have trip matrix T and let the component knots have trip matrices $T_1 \dots T_n$. Then by Definition 2.10 the Jones Polynomials for the given knots are:

$$V_K = (-t^{\frac{3}{4}})^{w(K)} \sum_{S \in \mathcal{S}(K)} t^{-\frac{1}{4}^{A(S)}} t^{\frac{1}{4}^{B(S)}} (-t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{nul(T_S)}$$

$$V_{K_1} = (-t^{\frac{3}{4}})^{w(K_1)} \sum_{S \in \mathcal{S}(K_1)} t^{-\frac{1}{4}^{A(S)}} t^{\frac{1}{4}^{B(S)}} (-t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{nul({T_1}_S)}$$

$$\vdots$$

$$V_{K_n} = (-t^{\frac{3}{4}})^{w(K_n)} \sum_{S \in \mathcal{S}(K_n)} t^{-\frac{1}{4}^{A(S)}} t^{\frac{1}{4}^{B(S)}} (-t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{nul({T_n}_S)}$$

Our claim is that $V_K = \prod_{i=1}^{n} V_{K_i}$. To begin showing this equivalence we first need to show that the product of the n sums for the component knots has the same number of terms as the sum for V_K, which will make this process significantly easier. Recall that the number of terms in a given sum here is 2^m, where m is the number of crossings in the given knot. Let K have m crossings and let $K_1, K_2, \dots K_n$ have $m_1, m_2, \dots m_n$ crossings, respectively. Then we know that $m = m_1 + \cdots + m_n$. Therefore, $2^m = 2^{m_1} \times \cdots \times 2^{m_n}$. This means that when expanding the product of the n sums of the component knots, we will have 2^m terms, each of which corresponds to one of the 2^m states for the component knot K. As an example, consider the connect sum of a trefoil and figure 8 knot. Suppose we are considering the state $AAABABA$ of the composite knot. This single state will yield one term in the sum V_K. This term corresponds to the product of the AAA term in the V_{K_1} sum and the $BABA$ term in the V_{K_2} sum by Theorem 3.5. This extends to an arbitrary number of components in the connect sum.

Because the number of terms on both sides of our alleged equality is the same, and because we can break up the states of the composite knot into smaller states in a very nice manner, our proof becomes very straightforward, if a little tedious: we can show that the large sum for the composite knot and the product of the smaller sums for the component knots are the same by going term by term. Consider a random state $S \in \mathcal{S}(K)$. Then this state S can be divided up into its component substates $S_1 \ldots S_n$, where each substate S_i is the section of the word S that corresponds to the current state of the component knot K_i.

Now consider the product of the terms from the sums for the component knots:

$$\prod_{i=1}^{n} t^{-\frac{1}{4}^{A(S_i)}} t^{\frac{1}{4}^{B(S_i)}} (-t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{nul(T_{i_{S_i}})}$$

We can break this product into three separate products for each like term in the expression:

$$\prod_{i=1}^{n} t^{-\frac{1}{4}^{A(S_i)}} \prod_{i=1}^{n} t^{\frac{1}{4}^{B(S_i)}} \prod_{i=1}^{n} (-t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{nul(T_{i_{S_i}})}$$

By Theorem 3.5, $A(S) = A(S_1) + \cdots + A(S_n)$ and $B(S) = B(S_1) + \cdots + B(S_n)$. Therefore by exponent rules,

$$t^{-\frac{1}{4}^{A(S)}} = \prod_{i=1}^{n} t^{-\frac{1}{4}^{A(S_i)}}$$

$$t^{-\frac{1}{4}^{B(S)}} = \prod_{i=1}^{n} t^{-\frac{1}{4}^{B(S_i)}}$$

By Theorem 3.3, $nul(T_S) = nul(T_{1_{S_1}}) + \cdots + nul(T_{n_{S_n}})$. Again by exponent rules we get

$$(-t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{nul(T_S)} = \prod_{i=1}^{n} (-t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{nul(T_{i_{S_i}})}$$

All together, this gives us the expression:

$$t^{-\frac{1}{4}^{A(S)}} t^{\frac{1}{4}^{B(S)}} (-t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{nul(T_S)} = \prod_{i=1}^{n} t^{-\frac{1}{4}^{A(S_i)}} t^{\frac{1}{4}^{B(S_i)}} (-t^{-\frac{1}{2}} - t^{\frac{1}{2}})^{nul(T_{i_{S_i}})}$$

Where the right hand side is a term from the sum for the composite knot and the left hand side is the product of the terms corresponding to this same state. Therefore on a term by term basis the large sum is the product of the smaller products as claimed.

All that is left to consider is the term from outside the summation, which is dependent on the writhe. By Theorem 3.4 we know that

$$w(K) = w(K_1) + \cdots + w(K_n)$$

Again by exponent rules this means

$$(-t^{\frac{3}{4}})^{w(K)} = \prod_{i=1}^{n} (-t^{\frac{3}{4}})^{w(K_i)}$$

Therefore

$$V_K = \prod_{i=1}^{n} V_{K_i}$$

as claimed. □

Example 4. Let us examine this theorem, along with Theorems 3.3 through 3.5, in action. Below is the composite knot formed by the trefoil on the left and the figure 8 knot on the right, along with its corresponding trip matrix:

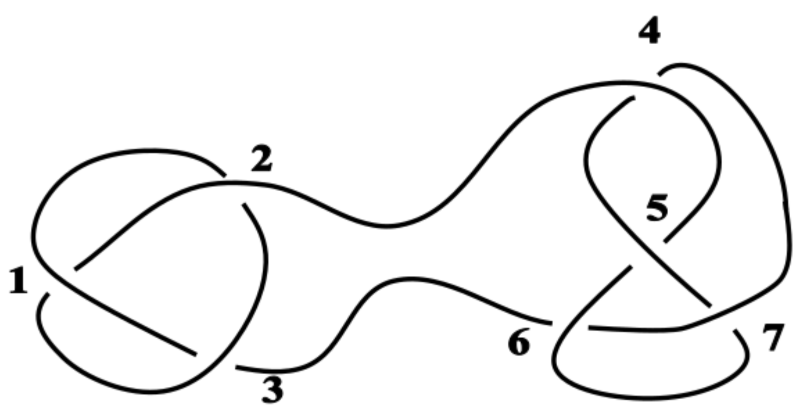


Figure 7: The trefoil and figure 8 connect summed.

$$T_K = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

First, let us consider the writhe, as it is dependent only on the trip matrix and not the toggled matrices. We calculate this as $5 - 2 = 3$. If we were to take the trip matrices for these two knots separately (see Appendix A) we get writhes of $3 - 0 = 0$ and $2 - 2 = 0$, which when added together, give 3 as well. So writhe is preserved as expected.

Let us focus on the specific state $ABABBAA$. This results in the following toggled matrices for the trefoil, figure 8, and their connect sum respectively:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

With a little linear algebra which we omit for the sake of brevity, we find that the first matrix has nullity 1, while the second has nullity 0. The third, as expected has nullity 1, which is the sum of the nullities of the two component matrices. Finally, it is clear that the number of A's and B's is preserved. The state of the first matrix is ABA, the second is $BBAA$, while the composite matrix is just their states joined together forming $ABABBAA$. This of course holds for all 2^7 states of the composite knot. For each one of these states in the Jones Polynomial for the composite knot, there is a corresponding pair of states from the two component knots for which all of the important information is additively preserved.

5 Utilizing the Trip Matrix to Determine When a Knot is Prime or Composite

The block structure of trip matrices of composite knots begs the question of whether or not the trip matrix could be used to determine if a given knot is prime or composite, and in the latter case, which prime knots make up the composite knot. In this section, we explore how a trip matrix may change under a relabelling of the crossings, give a stronger version of Theorem 3.1, and determine when a given trip matrix may be a block matrix in disguise due to a different labelling of the crossings.

5.1 Crossing Relabelling and the Trip Matrix

A natural question to ask when first working with the trip matrix is what happens if we choose a different labelling of the crossings. There are $n!$ labellings for a knot with n crossings, which come from $n!$ ways we can rearrange the numbering of the crossings (changing arrow direction does not change the trip matrix). While these different labellings generate the same Jones polynomial, the matrices themselves may look very different. We can see this in Figure 8, followed by the corresponding trip matrices for the two labellings. The same knot, when labelled in two different manners, gives two distinct matrices.

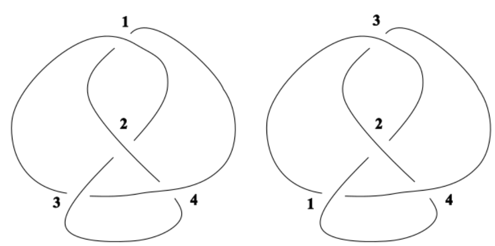


Figure 8: Different labellings for the figure 8 knot.

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

However, it turns out that every possible trip matrix is equivalent up to an operation we refer to as row/column swapping:

Definition 5.1 (Row/Column Swapping)**.** *Row/column swapping* is defined by first swapping column i with column j, then swapping row i with row j. This operation is denoted by $\Delta(i, j)$.

Lemma 5.2. The Δ operation is the same independent of whether rows or columns are swapped first.

Proof. Let $i, j, k \in \mathbb{N}$ and consider $\Delta(i, j)$. Let T be a trip matrix. We begin by swapping rows first, then columns. The following hold, where the first arrow denotes the swapping of rows and the second columns:

$$T_{ii} \to T_{ji} \to T_{jj}$$

$$T_{jj} \to T_{ij} \to T_{ii}$$

$$T_{ij} \to T_{jj} \to T_{ji}$$

$$T_{ji} \to T_{ii} \to T_{ij}$$

$$T_{ik} \to T_{jk} \to T_{jk}$$

$$T_{ki} \to T_{ki} \to T_{kj}$$

The above scenarios cover all possible cases; it remains only to show that when swapping columns first the results are the same:

$$T_{ii} \to T_{ij} \to T_{jj}$$

$$T_{jj} \to T_{ji} \to T_{ii}$$

$$T_{ij} \to T_{ii} \to T_{ji}$$

$$T_{ji} \to T_{jj} \to T_{ij}$$

$$T_{ik} \to T_{ik} \to T_{jk}$$

$$T_{ki} \to T_{kj} \to T_{kj}$$

We can see that reversing the order yields the exact same results, proving that the Δ operation is commutative. □

Theorem 5.3. The Δ operation in the trip matrix corresponds to a swapping of labels in the corresponding knot.

Proof. Let T_1 be the trip matrix for a knot K under some relabelling, and consider performing $\Delta(i, j)$ on T_1 to get a new matrix T_2. In T_1 row i, and thus column i, detail the relationship between crossing i and the other crossings in the knot, as well as itself. After row/column swapping, these exact same relationships are catalogued into row and column j as described in Lemma 5.2. Similarly the information pertaining to crossing j is now located where the information regarding crossing i previously was. This exact same result would be achieved if we were to instead swap the labels in K of crossings i and j; the same relationships would exist, but they would correspond to the rows and columns we swapped. Therefore, a swapping of two labels in the knot diagram is equivalent to performing the Δ operation on the corresponding rows and columns in the trip matrix. □

Definition 5.4 (Δ-equivalent)**.** We say two matrices are Δ-*equivalent* if there exists some finite sequence of Δ operations that can take one matrix to the other. If there is no such sequence of moves, then the two matrices are said to be Δ-*distinct*.

Theorem 5.5. For any given knot, there are up to $n!$ possible trip matrices, which are all Δ-equivalent.

Proof. Let K be a knot with n crossings. Then, there are $n!$ different ways to label the crossings of K. Each of these yields a trip matrix, and there are up to $n!$ unique trip matrices; since two rows can be the same, there might be less than $n!$ unique trip matrices as swapping the labels of these two crossings will not change the matrix. Let T_1 and T_2 correspond to labellings l_1 and l_2, respectively. There exists a finite number of swaps of labels to change l_1 to l_2. Since each of these is equivalent to a row/column swap by Theorem 5.3, the two matrices T_1 and T_2 are separated only by this same sequence of row/column swaps. Therefore, any two trip matrices for a given knot K are Δ-equivalent. □

5.2 The Trip Matrix as a Tool to Identify Compositeness

With our new Δ operation well-defined, we can finally get a stronger version of Theorem 3.1.

Lemma 5.6. All composite knots have block trip matrices up to some relabelling.

Proof. Let K be a composite knot. Then by Theorem 3.1 there exists a labelling that gives a block matrix structure for at least one possible trip matrix T for K. By Theorem 5.5, any other trip matrix for K generated by a different labelling is Δ equivalent to T. Therefore, any labelling for a composite knot will give a trip matrix that is Δ-equivalent to a block matrix. □

Theorem 5.7. A knot is composite if and only if its trip matrix is Δ-equivalent to a block matrix. Additionally, each block in the trip matrix represents the prime components from which the composite knot has been formed.

Proof. Theorem 3.1 satisfies the forward direction.

To show that a block matrix structure implies that the corresponding knot is composite, we can construct a copy of S^1 with labels for the undercrossings and overcrossings in order that respects the information obtained from the trip matrix. We informally refer to this construction as a “knot circle”. The order in which one encounters crossings when travelling around this circle represents the order they would meet this crossings when travelling along this as an actual knot. Also note that the construction that follows is by no means unique; our goal is not to find the only knot from a given trip matrix, rather it is to show if a trip matrix has a block structure, then any possible corresponding knot must be composite.

Let T be a trip matrix which has a block structure after some relabelling, and let us label its blocks B_1 through B_n. Label the jth crossing from the ith block by $B^{\pm}_{i_j}$, so the third overcrossing from the fifth block would be denoted $B^{+}_{5_3}$. Begin by placing B_1 on the circle. Note that the order in which we place the individual crossings from B_1 would be determined by the structure of B_1. However, we don’t care about that level of detail, only that all of the crossings from B_1 are laid out in a continuous sequence around the circle.

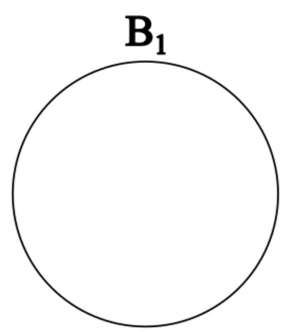


Figure 9: The Circle With The First Block Placed

Let’s begin by placing B_2. We have only one option, which is to place all the crossings from B_2 consecutively. Each pair $B^{+}_{2_i}$ and $B^{-}_{2_i}$ must contain either zero or both crossings from all pairs from B_1 between them, or else we would get a one in the trip matrix where there should be a zero. Additionally, the entirety of B_2 must lie on this same “side” of the circle relative to B_1. If we were to place some of the pairs from B_2 in one place and the others pairs in a distinct area separated by some crossings from B_1 then any path from a crossing from B_2 in the former section to itself would never see any crossing from latter section; this would mean splitting B_2 into two distinct blocks which contradicts our assumption that B_2 is a single block.

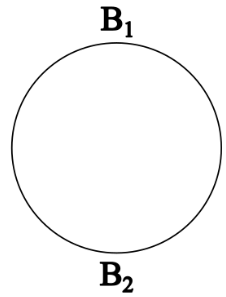


Figure 10: The Circle After Placing The Second Block

Next we must place B_3. We have only two distinct options here: either we place B_3 "next to" B_2, or we can place it "inside" B_2 (without loss of generality; we could just as easily use B_1 here). What we mean by this is that we place all of the crossings from B_3 consecutively between two of the over/undercrossings from B_2, just as we placed B_2 between two crossings from B_1. In the former case it is clear why this would create a block structure, since either path from a crossing in B_2 would either miss the entirety of B_3 or it would go through all of B_3. In the latter case, consider a random pair of crossings $m_{2_i}^+$ and $m_{2_i}^-$. Either both of these are clockwise from B_3, in which case the paths from one to the other either see or miss all of B_3, or one is clockwise and the other counterclockwise. In this case again one path sees all of B_3 and the other sees none of it.

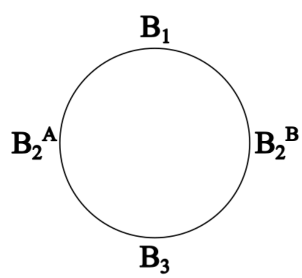


Figure 11: The Circle After Placing The Third Block Inside The Second

We can continue placing blocks in this manner until every block has been placed. Because of the numerous choices of where we can place each subsequent block, this method will create a multitude of different knots. However, as mentioned previously we only need to show that by placing blocks in this manner, we get a composite knot. And any circle representation of a knot constructed in this manner will in fact be composite, as we now show.

Consider the manner in which we placed the blocks. Because at the time each block was placed it was placed continuously, and because there was one block that we placed last, we know that at least one block is still continuous and hasn't been split up by the placement of another block. Consider what this looks like in the form of a knot. We start from the last crossing from some other component and then enter the block placed last. Because this block is continuous, we then travel through the entirety of this block before leaving and meeting some other crossing from a different block. This is precisely the result of a connect sum where we attach this block to the preexisting knot along the strand between the two other crossings seen before and after this block. Therefore, we can simply "cut out" this block from our knot circle to indicate that we have undone that connect sum with the corresponding knot and set it aside.

The removal of the block that is guaranteed to be connected will either result in a disjoint block being reconnected, or it won't. But if it doesn't, that means the penultimate block must also be connected, because this means the only opportunity for it to become disjoint (the placement of the final block, which we just excised) did not make it disjoint. Therefore we know we have a connected block which we can excise in the same manner. We can keep removing the blocks in reverse order to the manner in which they were placed because we know that they must be connected by the time we come to them for removal. This continues until every block has been excised, and what we have left is n distinct knot circles each representing a distinct block B_i. Each of these knot circles are in fact our prime components K_1 through K_n. So a block trip matrix does imply that the corresponding knot is composite, with each block in the matrix representing a prime component knot.

□

Example 5.

Consider Figure 12 below. This is an example of how one might construct a knot circle from a trip matrix with 6 blocks of unspecified size. Whenever a block has been split into multiple nonconsecutive sections by the placement of a subsequent block, we refer to that as B_i^A, B_i^B, etc. Consider the last block placed. In the example below that block could be B_4, B_5, or B_6. We will assume it was B_6 without loss of generality, and because the numbering suggests that we placed it last.

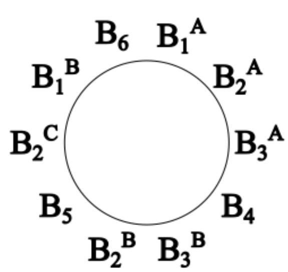


Figure 12: One example of a fully constructed circle.

Begin by removing B_6, which we claimed above was the last block placed:

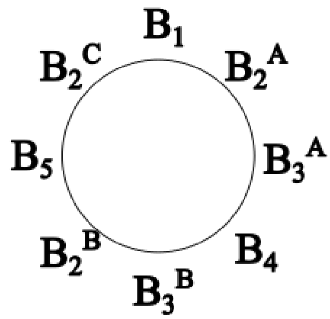


Figure 13: Our knot circle example with B_6 excised.

Now that B_6 has been removed, we see that the two previously disjoint sections of B_1 are now connected, and so we now consider it a continuous block labelled B_1. In our example, we have now have 3 connected blocks, which are B_1, B_4, and B_5. Remove B_5 and B_4 in one step, since they are both connected blocks. This results in Figure 14 below:

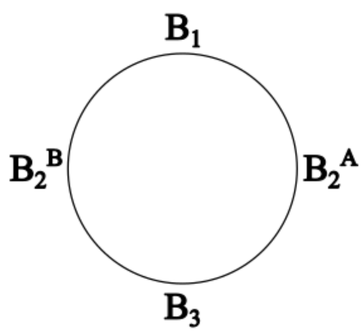


Figure 14: Knot circle with B_4 and B_5 removed.

The removal of B_5 connected B_2^B and B_2^C while the removal of B_4 connected all of B_3. Next we can remove B_3, which fully connects B_2:

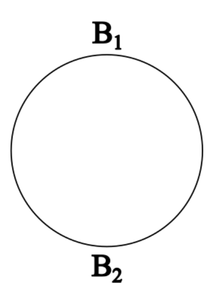


Figure 15: Removing B_3

We are now left with just B_1 and B_2 (just how we started) which we can simply break into two knot circles, and we are done.

The theorem above is extremely powerful because of its direct consequence in the following corollary. We now know that being prime or composite are equivalent to either not having a block structure in the trip matrix or having one.

Corollary 5.7.1. A knot is prime if and only if its trip matrix is not Δ-equivalent to a block matrix.

Proof. Knots can only be prime, composite, or the unknot, while trip matrices are either a block matrix up to some relabelling or not. Since the unknot has no crossings in its minimal crossing representation it cannot generate a trip matrix. Theorem 5.7 shows that a knot being composite and its trip matrix having a block structure are equivalent. Therefore, if a knot is prime, it must necessarily not have a block matrix structure, and a matrix which cannot attain a block structure under any relabelling must necessarily be prime. □

5.3 Relabelling Algorithm to Break a Trip Matrix Into Individual Blocks

We now have a method to determine if a given knot (with minimal crossing representation) is either prime or composite. However, this depends on our ability to take a given trip matrix and determine if it has a block structure or not. This is not a trivial problem. In order to ensure that a knot is prime, we must check all $n!$ possible labellings and verify that none have a block structure. Even for computers this number grows much faster than we would like. Additionally, if after some relabellings, a trip matrix gives two blocks, we are not necessarily done. Perhaps these blocks can be broken down further into smaller blocks. Ideally, we can not only determine if a knot is prime or composite, but in the latter case, determine exactly which prime knots form the larger composite knot. Here we detail a system of quick checks that can be performed to determine primeness, and an algorithm which determines the exact breakdown into prime knots for any given trip matrix.

5.3.1 Criteria for Primeness

The following are 4 quick checks that can immediately identify if a given knot is prime from its trip matrix. While these checks are (probably) not all-encompassing, in practice we found that every prime knot we checked met at least one of these criteria. This suggests that perhaps every prime knot meets on these criteria, which we discuss in Section 7.

Theorem 5.8. If the trip matrix T for knot K has less than 6 rows/columns, then K is prime.

Proof. The proof is trivial. The smallest possible composite knot is the connect sum of two trefoils which has 6 crossings and thus a trip matrix with 6 rows and columns. Anything smaller can only be prime. □

Theorem 5.9. If a trip matrix of any size has a row with less than 3 zeroes off the diagonal, then it must be prime.

Proof. As previously mentioned we are operating under the assumption that crossing number is preserved under connect sums, which implies that each block must have at least 3 rows and columns since the smallest prime knot, the trefoil, has 3 crossings. This means that for a knot to be composite, it must contain a block of at least rank 3. This means that the largest possible block in a matrix of rank n is $n-3$. Ignoring the diagonal (since any relabelling keeps entries on the diagonal and will thus always be part of any block) and assuming the worst case scenario that this entire block is made up of only ones, we get a maximum of $n-3$ ones in any row (and thus a minimum of 3 off-diagonal zeroes). Therefore if there exists a row with less than 3 off-diagonal zeroes it must be prime, since even the largest possible block could not contain this row. □

Theorem 5.10. If the total number of zeroes in a trip matrix of rank n is less than $6(n-3)$, then the corresponding knot must be prime.

Proof. Let's construct a trip matrix with the most possible ones that is still composite. To do so, we construct a matrix with 2 blocks, one of size 3×3 and the other of size $n-3 \times n-3$, both of which are made up entirely of ones. Including more blocks will increase the number of zeroes. In this case, the number of zeroes is precisely $2 \times 3 \times (n-3)$. This comes from there being two rectangles in the matrix (after the matrix is relabelled so that the blocks are intact) which are $n-3$ units long and 3 wide. We choose blocks of size 3 and $n-3$ since 3 minimizes the function $x(n-x)$ where x is the size of the smaller block, and thus gives us the minimum number of zeroes. Since this is the lower bound for the number of zeroes in a block matrix, any trip matrix with fewer zeroes must necessarily be prime. □

Theorem 5.11. If every row of a matrix of rank n has greater than $\frac{n}{2}$ ones, it must be prime

Proof. Suppose T_K meets the above criteria. We know that in order to be prime, T must have at least two blocks. Therefore, at least one of these must have number of rows and columns $\leq \frac{n}{2}$. If every row in T contains more than this number of ones, then it is impossible for a block of this size to exist, and so the corresponding knot must be prime. □

5.3.2 Block Matrix Relabelling Algorithm

The following algorithm can be used to determine when a given trip matrix possesses a block structure. While in the worst case scenario is can be time-consuming, it is significantly more effective than checking all $n!$ possible matrices.

1. Begin by making the 4 checks for primeness described above. If any of these are met, then the trip matrix is prime and we are done.

2. If none of the checks are met, then further work is needed. Count the number of ones in each row, and perform row/column swaps so that the rows are arranged in ascending order by the number of ones they contain. Label the rows 1 through n. If multiple rows have the same number of ones the order of them doesn't matter.

3. Go through the rows in this ascending order, and when the number of ones in a given row is greater than the label of the row, stop; this is the smallest possible block this matrix can contain. Call this number k. If this is not reached by the time we reach row number $\lceil \frac{n}{2} \rceil$, then the matrix must be prime.

4. If this initial swapping results in a block, then we can simply cut this block out of the matrix and set it aside as one of the components of K, and start from step 1 on the remaining matrix. If not, then some of the entries in the first k rows in the columns to the right of k must be ones. If the first k rows of column $k+1$ are all zeroes, then we may also swap this column with a column to the right of it that contains ones. We continue this process until the column to the right of our attempted block (now of size $k' \times k'$) contains some ones in the first k' rows. Again, if we attain a block at any point we can cut it out and start again.

5. Now, it is possible that the number of rows with less than k' ones is actually greater than k' itself (call the number of these rows p). This means that there are actually $\binom{p}{k'}$ possible choices for the rows that we include in our attempted $k' \times k'$ size block. We must now check each of these possibilities to see if any result in a block. Note that while theoretically this can be a large number, in practice p is often not much bigger than k' and so the check is generally very short.

6. If at this point we have still not found a block, we now repeat every step from step 3 onward with an attempted block of size $k+1$. We continue this process either until the entire matrix has been divided into prime blocks or we reach row $\lceil \frac{n}{2} \rceil$, in which case the matrix must be prime.

As previously mentioned, in the worst case scenario that the matrix in question corresponds to a prime knot, this algorithm is time consuming. However in the case that it is composite, we have found in practice that blocks tend to form very quickly. While this comment is purely anecdotal, it indicates that if blocks exist they will generally appear quickly in this process, and if no blocks appear soon, there is a decent chance that the knot is prime. We also do not claim that this algorithm is guaranteed to break a composite matrix down into blocks or confirm that it is prime, nor do we claim that it is optimal, hence the omission of any proof.

Example 6. Relabelling Algorithm
We begin with an arbitrary 6×6 trip matrix (which does not necessarily correspond to an actual knot) and work through the algorithm as explained:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

We begin with the 4 checks for primeness. This matrix has rank 6, so the first fails. Going through each row, they all have at least 3 zeroes off the diagonal, and there are rows with less than 4 ones, so the first and fourth checks fail. Finally counting all the zeroes, we have 24, which is greater than the minimum $18 = 6(6-3)$ required. Because all the checks fail, we move on to the algorithm itself.

The number of ones in each row is denoted by the sequence $2, 3, 1, 3, 1, 2$. Therefore, we perform Δ swaps so that this order is a non-decreasing sequence. We have multiple options here, so without loss of generality we can perform $\Delta(2,5)$, $\Delta(1,3)$, $\Delta(4,6)$. This gives us the following Δ-equivalent trip matrix:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

By the criteria from the algorithm, the smallest possible block here is 3. Column 4 contains a one its first 3 rows, so we move on to the next step. Now, we come to a tricky spot. Every row has less than 3 ones, so any of the 6! possible relabellings could potentially yield a block. In general the number we need to check at this step is much smaller, but in this case we got unlucky. We could brute force this, or we can move with finesse. Entries $(1, 4)$, $(3, 5)$, and $(3, 6)$ are all ones where we need zeroes. Performing $\Delta(3, 4)$ could potentially solve this problem:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

And voila! Our finesse yielded a block matrix. Since each of these blocks is rank 3 they cannot be broken down further, and so we are done. Note that this matrix is not actually possible; there are only two trip matrices of rank 3 which correspond to the trefoil and its mirror image, and neither of these blocks are those matrices (see Appendix A for these trip matrices). This was merely an example of the algorithm in action. This example was also the worst-case scenario in terms of the number of potential checks. In our experience the number of checks is usually much smaller in practice, but this example demonstrates that some mathematical intuition can often be used to help address this problem just as much as the algorithm does.

6 Exploration of the Effects of Reidemeister Moves on Trip Matrices

As discussed earlier, our work hinges on several assumptions. One of these is that we are working with knot diagrams that have the minimum number of crossings possible. Of course, this is generally not the case, and finding the minimal crossing number for a given knot is not a trivial problem. As such, we hoped to explore how the trip matrix behaves under Reidemeister moves to see if it could be used as a method to detect extraneous crossings, and therefore be used as a tool to calculate minimal crossing number. Unfortunately (albeit unsurprisingly) the trip matrix is simply not powerful enough of a tool to be utilized in this manner, as we will show later. This section details our exploration into this subject and what results we did find.

6.1 Type 1 Reidemeister Moves

A Type 1 Reidemeister amounts to effectively the same result as discussed in Lemma 3.2. Below, we see the two possibilities for a Type 1 Reidemeister move (depending on the direction the twist is introduced) with their overcrossings and undercrossings labelled. In Figure 16 the resulting structure in the matrix is a row/column of all zeroes, while in Figure 17 the row/column is all zeroes with a 1 on the diagonal. This structure is only possible when an R-1 move occurs or under the conditions previously established in Lemma 3.2, which, while not strictly an R-1 move, is an extraneous crossing that can nonetheless be removed. Note that if we were to point the overcrossing arrows in the opposite direction, the path would then go through every other crossing twice before returning to itself, which would similarly yield all zeroes.

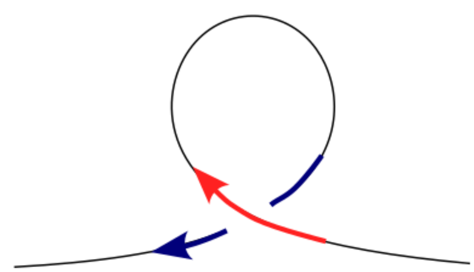


Figure 16: Type 1 Move twisted in one direction.

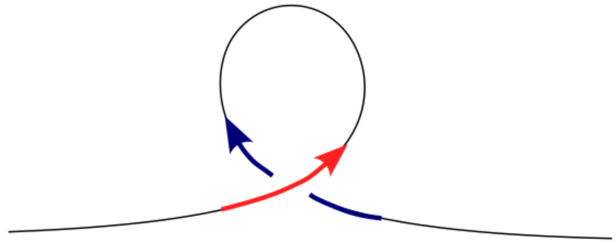


Figure 17: Type 1 Move twisted in the other direction.

6.2 Type 2 Reidemeister Moves

Naturally, Type 2 moves are more complex in nature. First we see what structure a Type 2 move yields in the trip matrix. Below in Figure 18 we see the 4 possible labellings for an isolated Type 2 move. Additionally, for each of these labellings there are two ways to connect the loose ends. If the top right and bottom right are connected, then we have a link instead of a knot, so we exclude that possibility. The other options are top right to bottom left and top right to top left. We considered all 8 cases but only illustrate the 4 different labellings without reattaching the strands for the sake of brevity.

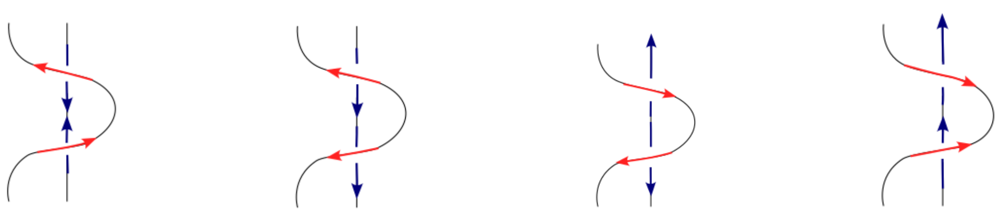


Figure 18: 4 general arrow labellings for a Type 2 Reidemeister Move.

In every case, the same result will occur regarding all other crossings in the knot (the crossings not shown). Let A and B denote the crossings shown that are formed by the Type 2 move. If the path from crossing A back to itself goes through a given crossing i once, then the same must be true for crossing B, and vice versa. This is similarly true if the path goes through i zero times or twice. This is because these two crossings are adjacent when approached from any direction; supposing one decides to start at crossing i and take the path back to i^-, then if the path crosses A or B at any point then the next crossing it immediately encounters is its pair.

The only other thing to consider is the 2×2 submatrix that details the relationship between crossings A and B, and of course their diagonal entries. After going through 8 possibilities (the details of which we again omit for brevity), we are are left with two possibilities, which are shown below:

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

The resulting change in the matrix is an additional two rows and columns, where in the 2×2 submatrix where they meet we have one of the above forms, and the remaining entries are in pairs, where for each other crossing the entries detailing the relationship between that crossing and crossings A and B are the same. Below is an example of a matrix where the last two rows and columns are the result of a Type 2 move:

$$\begin{bmatrix} x & x & x & x & x & 0 & 0 \\ x & x & x & x & x & 1 & 1 \\ x & x & x & x & x & 1 & 1 \\ x & x & x & x & x & 0 & 0 \\ x & x & x & x & x & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Unfortunately, this structure can appear in the absence of a Type 2 move. We found an example of a knot that generated this structure but the crossings that created this structure were not the result of a Type 2 move, shown in Figure 19. This example shut down our attempt to use the trip matrix as a tool to eliminate extraneous crossings, since we could not assume that this structure implied a Type 2 move.

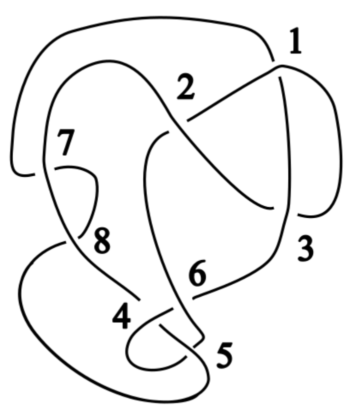


Figure 19: Type 2 Counterexample

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

What is interesting here is that there is in fact a clear Type 2 Reidemeister move (crossings 7 and 8). However, because rows 6 and 8 are identical, by looking at the trip matrix we can't tell which actually pairs with row 7 as a Type 2 move. Of course, we could simply swap the identical rows and assume we swapped it so that the rows we are removing are collectively the Type 2 move. But we do not know if this structure only appears when there is a removable Type 2 move somewhere in the knot, or if there are times when it can appear in the complete absence of a Type 2 move. We address this, and related questions, in Section 7.

6.3 Type 3 Reidemeister Moves

Upon the realization that the Type 2 structure in the trip matrix was not an if and only if situation, it became clear that exploring further into this topic would be fruitless, since there is no way of identifying Type 2 moves solely from the trip matrix (at least to our understanding). Because of this (and a lack of time) we did not explore how Type 3 moves affect the trip matrix.

7 Further Questions

There are a number of lingering questions from our research that could be the basis for interesting future work, which we detail below:

- Is there another way to determine extraneous crossings?
  - While our attempt to identify extraneous crossings by analyzing Reidemeister moves did not prove successful, it is possible that there are other patterns in matrices that could be used to answer this question. In this vein, we have included an appendix of trip matrices of prime knots (see Table A).
- Can we formalize the algorithm described in Section 5?
  - We did not prove that our algorithm performed as claimed. Future work could involve proving that this algorithm does indeed perform its intended, or alternatively coming up with a more efficient manner of relabelling trip matrices. This could also entail finding more criteria for primeness/compositeness that could be used to reduce the number of knots we actually need to implement the algorithm for.
- Does every prime knot meet one of the listed criteria for primeness, making our algorithm obsolete?
  - Every knot for which we computed a trip matrix met one of the prime criteria listed in Section 5. We conjecture that this list (perhaps with the addition of other easy checks) may in fact identify every prime trip matrix. If this is the case then the only use for the algorithm would be to rearrange and break down a composite trip matrix into its component blocks, rather than use it to check for primeness.
- What effect does a Type 3 Reidemeister move have on the trip matrix?
  - We did not attempt to answer this question because the investigation into Type 2 moves did not come to fruition. However this is still an interesting question that could be potentially useful, and it could reveal more about how the trip matrix behaves under ambient isotopies.
- Can distinct knots have identical trip matrices for representation with minimal crossings?
  - Naturally, this question is only applicable when the two distinct knots have the same crossings number. It is known that there are distinct knots that have the same Jones polynomial, and some of these pairs even have the same

crossing number. This would suggest that these knots may have identical trip matrices up to Δ-equivalence, although it is theoretically possible for Δ-distinct trip matrices to generate the same Jones polynomial. A thorough analysis of this subject would likely involve filling out the catalogue of primes found in Appendix A for all prime knots up to a certain crossing number, particularly those pairs of knots with the same Jones polynomial and crossing number. If the answer to this question is yes, it would open up further questions regarding why two distinct knots could have identical trip matrices and how powerful a tool the trip matrix itself is at identifying distinct knots.

- Can a knot have Δ-distinct trip matrices as a result of different minimal crossings representations?
  - For some knots, the minimal cross number representation is not unique, and there are multiple distinct knot diagrams for a given knot that all have minimal crossing number. Would these distinct diagrams yield Δ-distinct trip matrices or would they be Δ-equivalent? This is partially tied to the exploration of Type 3 Reidemeister moves, since that is a simple way to change a knot diagram without affecting the number of crossings.
- Can the Type 2 Reidemeister structure appear in knots with no Type 2 moves to remove?
  - As discussed in Section 6, our only counterexample still contained a Type 2 Reidemeister move. Is this the only way that this structure can appear in the trip matrix, or are the others? If the former is true, this reopens the possibility of using the trip matrix to undo Reidemeister moves.

8 References

References

[1] Colin C. Adams. *The knot book*. An elementary introduction to the mathematical theory of knots. W. H. Freeman and Company, New York, 1994, pp. xiv+306. ISBN: 0-7167-2393-X.

[2] V. F. R. Jones. "Hecke algebra representations of braid groups and link polynomials". In: *Ann. of Math. (2)* 126.2 (1987), pp. 335–388. ISSN: 0003-486X. DOI: 10.2307/1971403. URL: https://doi.org/10.2307/1971403.

[3] Louis H. Kauffman. "New invariants in the theory of knots". In: 163-164. On the geometry of differentiable manifolds (Rome, 1986). 1988, 6, 137–219, 282 (1989).

[4] Kurt Reidemeister. "Elementare Begründung der Knotentheorie". In: *Abh. Math. Sem. Univ. Hamburg* 5.1 (1927), pp. 24–32. ISSN: 0025-5858. DOI: 10.1007/BF02952507. URL: https://doi.org/10.1007/BF02952507.

[5] Dale Rolfsen. *Knots and links*. Vol. 7. Mathematics Lecture Series. Corrected reprint of the 1976 original. Publish or Perish, Inc., Houston, TX, 1990, pp. xiv+439. ISBN: 0-914098-16-0.

[6] Louis Zulli. "A matrix for computing the Jones polynomial of a knot". In: *Topology* 34.3 (1995), pp. 717–729. ISSN: 0040-9383. DOI: 10.1016/0040-9383(94)00041-I. URL: https://doi.org/10.1016/0040-9383(94)00041-I.

Emerson Worrell
Mathematics Department
Colorado College
Colorado Springs, CO 80946
Email: *e_worrell@coloradocollege.edu*

A Catalogue of Primes

In the process of conducting our research we calculated a number of trips matrices for different prime knots. We include them here for posterity, and so that the reader may potentially discover patterns in the nature of these matrices that we missed. Below are the trip matrices for 17 different prime knots up to crossing number 10, denoted with their Alexander-Briggs-Rolfsen notation. The reader is free to use the criteria for primeness listed in Section 5 to verify that every knot listed meets at least one of the criteria, supporting our conjecture in Section 7. Note that the first (3_1) knot is the left-handed trefoil while the second is the right-handed trefoil, the mirror image of the former.

(3_1)	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
(3_1)	$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$
(4_1)	$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$
(5_1)	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
(5_2)	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$

$$(6_1)\qquad \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$(6_2)\qquad \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$(7_1)\qquad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$(7_2) \qquad \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$(7_3) \qquad \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$(7_4) \qquad \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$(8_1) \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$(8_2) \quad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$(8_4) \quad \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$(8_{10}) \qquad \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$(9_4) \qquad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(9_{10})

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

(10_1)

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$