# Instructions for `kauffmanstates.py`: Python code that implements Kauffman state-sum algorithm for the Alexander polynomial

Allison H. Moore

**Disclaimer 1.**

If your only goal is to compute an Alexander polynomial, don't use this code! Instead you should use the Mathematica package *KnotTheory* available at `katlas.org` (or some other more efficient method). The present algorithm is relatively inefficient, but independently useful because it produces a generating set for the knot Floer complex. In favorable scenarios this algorithm determines whether a given knot is fibered, which cannot be determined by the Alexander polynomial alone.

**Disclaimer 2.**

The Python code `kauffmanstates.py` contains subroutine which generates all of the spanning trees of a graph, and it also locates a 'minimum weight' spanning tree. If your only goal is to count the number of spanning trees or find a minimum weight spanning tree then don't use this code! There are much better algorithms and formulas to use for these purposes. In particular, any cofactor of the graph Laplacian provides the number of spanning trees (just take a determinant!) and Kruskal's algorithm will identify a (non-unique) min or max spanning tree.

## 1 Intro

The purpose of this note is to give an expository description and practical instructions to implement an algorithm which computes the Alexander polynomial from a state-sum model. This model also produces a generating set for the knot Floer complex. The existence of the state-sum model of the Alexander polynomial is due to Kauffman [Kau83], and the version presented here more-or-less follows Ozsváth and Szabø's reinterpretation, as in [OS03], where its relation to the knot Floer complex is proved. Another exposition of this algorithm may be found in [LM13].

The intended audience of this note consists of:

1. Students enrolled in Math 499 or other new-comers to the Alexander polynomial
2. People seeking instructions to create input for the Python program `kauffmanstates.py`. (Skip to Section 4.)
3. Other people who want to implement the Kauffman state algorithm by hand.

For the benefit of students and new-comers, there are several easy exercises peppered throughout Section 2. These may be omitted by those in a hurry.

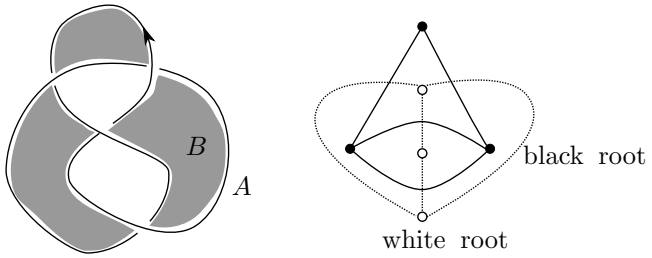


Figure 1: The figure eight knot has been checkerboard shaded, and its corresponding black and white graphs are pictured to the right. The regions incident to the distinguished edge are marked with $A$ and $B$, and the corresponding roots are indicated.

# 2 A state sum for the Alexander polynomial

The Alexander polynomial of a knot $K \subset S^3$ is a Laurent polynomial $\Delta_K(t) \in \mathbb{Z}[t, t^{-1}]$. It is an invariant of a knot $K$ in the three-sphere, meaning that any diagram of $K$ that one uses to compute $\Delta_K(t)$ should result in the same polynomial.

## 2.1 Kauffman states

It is a theorem of Kauffman [Kau83] that the Alexander polynomial of $K$ can be described by a state sum formula. In general this is an expression of the following form,

$$\Delta_K(t) = \sum_{\mathbf{x} \in \mathcal{S}} (-1)^{M(\mathbf{x})} t^{A(\mathbf{x})}, \tag{1}$$

where each state $\mathbf{x}$ is equipped with a bigrading $(A(\mathbf{x}), M(\mathbf{x})) \in \mathbb{Z} \oplus \mathbb{Z}$ such that the symmetrized Alexander polynomial of $K$ is given by the above sum. The particular reformulation of the Kauffman state sum which is explained here follows [OS03].

Let $K$ be oriented knot in $S^3$, and fix a decorated knot projection $D(K)$ for the knot $K$. By a decorated knot projection we mean a knot projection with a distinguished edge. Notice that a knot projection (when thought of as a planar, four-valent graph) divides the plane into $m + 2$ regions, where $m$ is the crossing number of the diagram. We give the diagram $D(K)$ a checkerboard coloring, and in doing so, the distinguished edge of $D(K)$ gives rise to a distinguished black region and a distinguished white region, labelled $A$ and $B$, with $A$ referring to the unbounded region. Thus there are $m + 2$ total regions in the checkerboard coloring, and the sets of crossings and undistinguished regions both have cardinality $m$. See Figure 1.

**Definition 2.1.** A *Kauffman state* is a map which associates to each crossing in the decorated projection one of its four incident regions in such a way that every undistinguished region is associated to a crossing, and every crossing is associated to an undistinguished region.

**Exercise 2.2.** Draw all of the Kauffman states for the figure eight knot.

**Exercise 2.3.** Another way to think of a Kauffman state is that it specifies a particular bijection between the set of $m$ crossings and the set of $m$ distinguished regions. Prove this is equivalent.

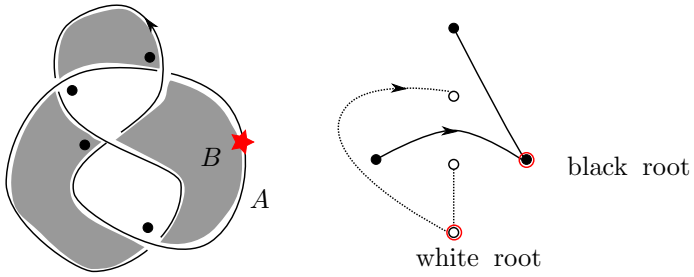


Figure 2: One of the five Kauffman states of this figure eight knot is pictured along with its corresponding pair of spanning trees.

## 2.2 Spanning trees

Also associated to the checkerboard coloring of the knot projection are two standard graphs, which we denoted $G_B$ and $G_W$ (for 'black graph' and 'white graph'). The regions $A$ and $B$ of the checkerboard colored knot projection correspond to a pair of vertices in $G_B$ and $G_W$ which we call the *roots*. Recall that a spanning tree of a connected graph $G$ is is a subgraph which contains every vertex of $G$ and does not contain any cycles. It is not hard to see that a Kauffman state corresponds with a spanning tree of $G_B$ and vice versa.

Each black spanning tree $T \subset G_B$ uniquely determines a white spanning tree $T^* \subset G_W$ which is *dual* to the black spanning tree. For a fixed Kauffman state $\mathbf{x} \in \mathcal{S}$, let $\mathcal{T}_\mathbf{x} = T_\mathbf{x} \cup T^*_\mathbf{x}$ denote the black and white spanning trees which correspond to $\mathbf{x}$. From now on, when we mention a Kauffman state we simultaneously mean $\mathbf{x}$ and the spanning trees $\mathcal{T}_\mathbf{x}$. The set of Kauffman states $\mathcal{S}$ is in bijective correspondence with the set of spanning trees of the black graph, which itself is in bijective correspondence with the set of spanning trees of the white graph. Thus the spanning trees will become the 'states' in our state-sum model for the Alexander polynomial. We typically work with spanning trees because they are easier to draw and easier to code than knot diagrams.

**Example 2.4.** A checkerboard coloring of a decorated projection of the figure eight knot is pictured in Figure 1 along with its corresponding black and white graphs. One of the Kauffman states of the figure eight knot and its corresponding spanning trees are pictured in Figure 2.

**Exercise 2.5.** Draw all of the Kauffman states and their corresponding spanning trees for the right-handed trefoil knot.

**Exercise 2.6.** Show that the set of Kauffman states are in a bijective correspondence with the set of spanning trees of $G_B$.

## 2.3 The two gradings

Each state $\mathbf{x}$ (or equivalently, its corresponding spanning trees $T_\mathbf{x} \cup T^*_\mathbf{x}$) is equipped with two integers $(A(\mathbf{x}), M(\mathbf{x}))$, called the $A$-grading and the $M$-grading. In this section we describe how to compute the two maps

$$A : \mathcal{S} \to \mathbb{Z} \text{ and } M : \mathcal{S} \to \mathbb{Z}$$

which determine these gradings.

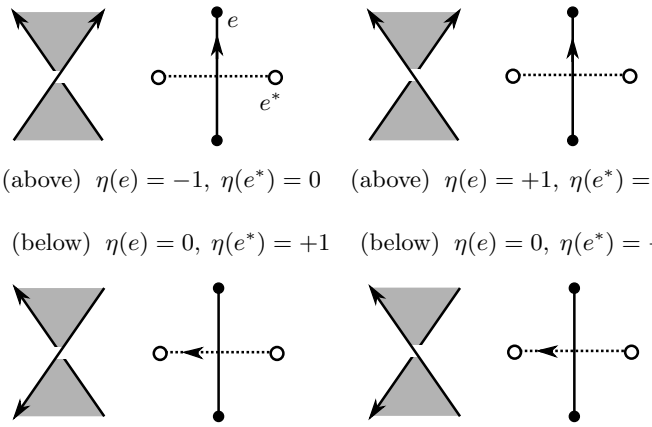


Figure 3: The labels $\eta(e)$ and $\eta(e^*)$ for the edges $e \in G_B$ and $e^* \in G_W$. The edge orientations pictured are those induced by $K$ on $G_B$ or $G_W$.

Label each edge $e$ of $G_B$ and $G_W$ with the incidence number $\eta(e) \in \{-1, 0, 1\}$ according to Figure 3. These incidence numbers apply to any subgraphs of $G_B$ and $G_w$, and in particular, apply to any spanning trees. We describe two partial orientations on the edges of $T_{\mathbf{x}}$ and $T^*_{\mathbf{x}}$. The first orientation is a total orientation on the edges of the trees which flows away from the black and white roots of $T_{\mathbf{x}}$ and $T^*_{\mathbf{x}}$. The second orientation is a partial orientation (only some of the edges get oriented) and it is induced by the natural orientation of the knot. See Figure 3, and note that at each crossing exactly one of the edges of $T_{\mathbf{x}}$ or $T^*_{\mathbf{x}}$ is oriented.

Then, $A(\mathbf{x})$ is defined by

$$A(\mathbf{x}) = \frac{1}{2} \sum_{e \in \mathcal{T}_{\mathbf{x}}} \sigma(e)\eta(e), \tag{2}$$

where

$$\sigma(e) = \begin{cases} 0 & \text{if } e \text{ is not oriented by } K \\ +1 & \text{if the two induced orientations on } e \text{ agree} \\ -1 & \text{if the two induced orientations on } e \text{ disagree.} \end{cases}$$

Though it is not indicated in the notation, $\sigma(e)$ depends on $\mathbf{x}$, and which $\mathbf{x}$ will be clear from the context; $\eta(e)$ does not depend on $\mathbf{x}$. Next, $M(\mathbf{x})$ is defined by summing only over edges on which the two orientations agree,

$$M(\mathbf{x}) = \sum_{\substack{e \in \mathcal{T}_{\mathbf{x}} \\ \sigma(e) = +1}} \eta(e). \tag{3}$$

**Example 2.7.** The figure eight knot pictured in Figure 1 has five Kauffman states; the trees corresponding to these five states are displayed in Figure 4. The $A$ and $M$-gradings are as follows:

$$\begin{aligned} A(\mathbf{x}_1) &= 0, & M(\mathbf{x}_1) &= 0 \\ A(\mathbf{x}_2) &= -1, & M(\mathbf{x}_1) &= -1 \\ A(\mathbf{x}_3) &= 1, & M(\mathbf{x}_1) &= 1 \\ A(\mathbf{x}_4) &= 0, & M(\mathbf{x}_1) &= 0 \\ A(\mathbf{x}_5) &= 0, & M(\mathbf{x}_1) &= 0. \end{aligned}$$

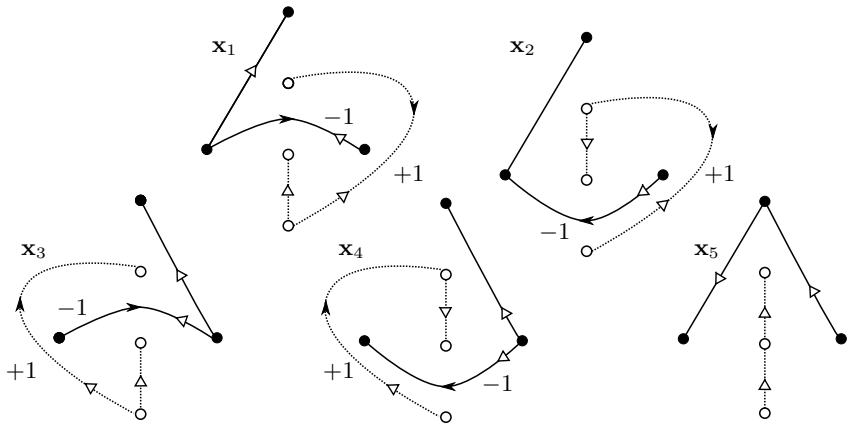


Figure 4: The five Kauffman states of the figure eight knot are depicted as trees. White arrows indicate the orientation on the tree which points away from root. Black arrows indicate the orientation induced by $K$. Edges are labeled by by their incidence numbers $\eta$ when $\eta(e) = \pm 1$; when $\eta(e) = 0$ the label is omitted.

Using formula (1) we obtain the Alexander polynomial $-t^{-1} + 3 - t$ for the figure eight knot.

**Exercise 2.8.** Use the state-sum model to show that the Alexander polynomials for the right-handed and left-handed trefoils are $t^{-1} - 1 + t$ (up to sign).

# 3 Fiberedness and the unique state property

For those of you enrolled in Math 499, we'll be spending some time exploring the graph theoretic property described below, and study different classes of knots that admit this property. For now, we mention the following facts and definitions without much exposition.

Let $\mathcal{S}_{D(K)}$ be the set of Kauffman states corresponding to some knot projection $D(K)$ of the knot $K$.

**Definition 3.1.** If $\mathcal{S}_{D(K)}$ contains exactly one state $\mathbf{x}$ with maximal $A$-grading, let us say that $\mathcal{S}_{D(K)}$ has the *unique maximal state* property. If $\mathcal{S}_{D(K)}$ contains exactly one state $\mathbf{x}$ with minimal $A$-grading, let us say that $\mathcal{S}_{D(K)}$ has the *unique minimal state* property.

**Definition 3.2.** If $K$ admits a diagram such that $\mathcal{S}_{D(K)}$ has either the unique maximal or minimal state property (or both), say that $K$ satisfies the *unique state property*.

Recall that the Alexander polynomial is symmetric, and can be written

$$\Delta_K(t) = a_d t^{-d} + a_{d-1} t^{-d+1} + a_1 t^{-1} + \cdots + a_o + \cdots + a_{d+1} t^{d-1} + a_d t^d,$$

where $a_i \in \mathbb{Z}$. Notice that if $D(K)$ has the unique maximal state property, then the coefficient $a_d$ of $t^d$ is one. By the symmetry of the Alexander polynomial, this implies that the coefficient of $t^{-d}$ is also one, though it may be the case that there are multiple states in $A$-grading $-d$ which cancel in formula (1) to give the overall coefficient one. In other words, it is possible for $\mathcal{S}_{D(K)}$ to have

the unique maximal state property yet not have the unique minimal state property (or vice versa). However, we do have the following remark.

**Remark 3.3.** If $K$ has the unique state property, then the Alexander polynomial of $K$ is monic.

Be cautioned though — the converse statement is not true! There are certainly knots which do not satisfy the unique state property, but whose Alexander polynomial coefficients work out to be monic via cancellation in formula (1).

The following is a theorem of Ni [Ni07] in knot Floer homology.

**Theorem 3.4.** *The knot Floer group $\widehat{HFK}(K, g(K))$ is isomorphic with $\mathbb{Z}$ if and only if $K$ is fibered.*

Ni proved that the knot Floer homology of $K$ is of rank one in the Alexander grading corresponding with the genus of the knot if and only if the knot is fibered. In the event that our knot diagram satisfies the unique state property, then the knot Floer chain complex has only a single generator in either the maximal or minimal Alexander grading. This implies that the homology in that grading is also rank one, because the differential vanishes. Furthermore, the grading must correspond with the genus. In summary, we have the following important observation:

**Observation 3.5.** If any particular knot diagram of a knot $K$ admits the unique state property, then $K$ must be fibered.

# 4 Input instructions for Python

The Python program `kauffmanstates.py` implements the algorithm described above. The program can be accessed at the following url:
`http://www.math.rice.edu/~ahm6/code/kauffmanstates`

The input is modified in lines 11 – 16 of the program. **Do not change any other line of the program.**

## 4.1 Directions to create input

Input parameters

- `black_directed, white_directed`: Numpy arrays of integers
- `black_weights, white_weights`: Numpy arrays of integers
- `black_root, white_root`: integers

Enumerate the vertices of the black graph and white graph arbitrarily as $b_0, \ldots, b_k$ and $w_0, \ldots, w_\ell$, respectively. Enumerate the edges of the black graph arbitrarily as $e_0, \ldots, e_m$. The enumeration of edges of the white graph is determined by the black graph as $e_0^*, \ldots, e_m^*$. White edges *must* be numbered such that the $i$-th white edge is dual to the $i$-th black edge. (This is not strictly necessary for by-hand computations, but it is necessary for the Python algorithm to work.) Edges are directed (or undirected) and assigned incidence numbers according to the convention described Figure 3.

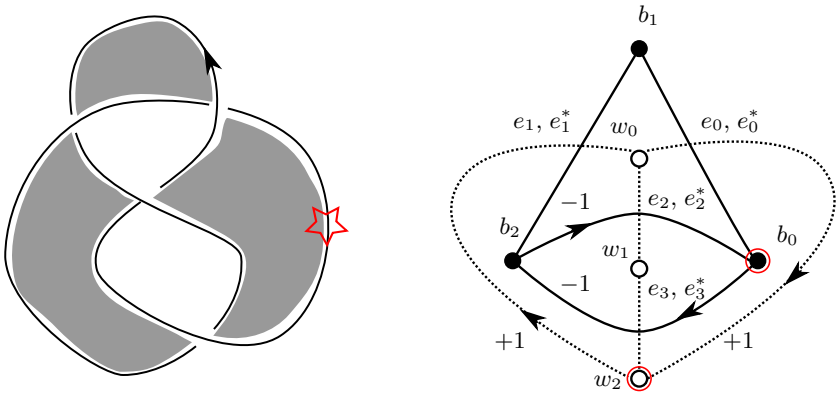


Figure 5: Black arrows indicate the orientation induced by $K$. Edges and vertices are labeled and the incidence numbers $\eta(e)$ are indicated when $\eta(e) \neq 0$.

The black and white graphs are each encoded as a directed incidence matrix `black_directed` or `white_directed` using a Numpy array. Rows correspond with vertices and columns correspond with edges. Columns should contain exactly two nonzero entries; for example, if $+1$ and $-1$ occur in spots $(j, i)$ and $(k, i)$, respectively, this indicates that edge $e_i$ points from vertex $b_k$ to vertex $b_j$, that is, from the negative towards the positive. Undirected edges are encoded with two positive ones.

The two lists, `black_edgeweights` and `white_edgeweights` must also be input. Each entry of the list indicates the incidence number $\eta(e) \in \{0, 1, -1\}$ (i.e. edge weight) assigned to the corresponding edge in the black and white graphs. The lists should agree in length with the number of columns of the black and white incidence matrices.

Finally the user must input two integers `black_root` and `white_root` indicating the index of the vertex in each of the black and white graphs which corresponds with the root. Remember that the black and white roots come from black and white regions that separated by a common arc in the checkerboard diagram.

Here is an example of correct input for the figure eight knot labelled as it is in Figure 5.

```
black_directed = array( [ (1,0,1,-1), (1,1,0,0), (0,1,-1,1) ] )
white_directed = array( [ (-1,1,1,0), (0,0,1,1), (1,-1,0,1) ] )
black_edgeweights = [ 0, 0, -1, -1 ]
white_edgeweights = [ 1, 1, 0, 0 ]
black_root = 0
white_root = 2
```

For comments, questions and corrections contact Allison Moore:
`allison.h.moore@rice.edu`.

# References


[Kau83] Louis H. Kauffman. *Formal knot theory*, volume 30 of *Mathematical Notes*. Princeton University Press, Princeton, NJ, 1983.

[LM13] Tye Lidman and Allison H. Moore. Pretzel knots with L-space surgeries. Preprint, 2013. arXiv:1306.6707 [math.GT].

[Ni07] Yi Ni. Knot Floer homology detects fibred knots. *Invent. Math.*, 170(3):577–608, 2007.

[OS03] Peter Ozsváth and Zoltán Szabó. Heegaard Floer homology and alternating knots. *Geom. Topol.*, 7:225–254 (electronic), 2003.