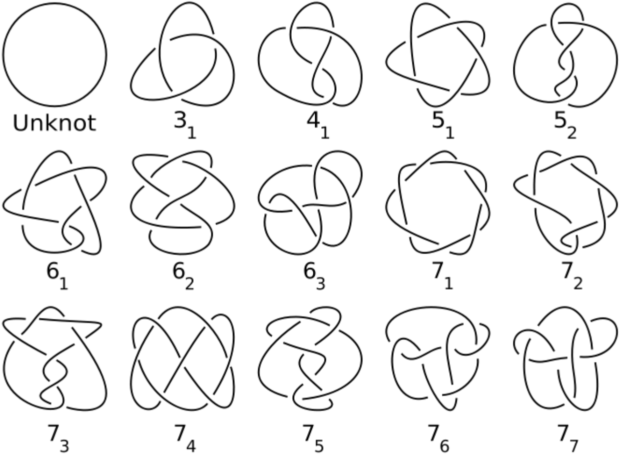
-SpaceX Presentation-
Combinatorial Reconfiguration Program

Miles Clikeman

August 31, 2020

Knot Theory - Background

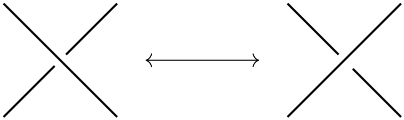


Crossing Change

- **Crossing Change** - An operation on a knot diagram where the over and under strands of a single crossing are switched.

Crossing Change

- **Crossing Change** - An operation on a knot diagram where the over and under strands of a single crossing are switched.

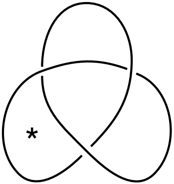


Region Crossing Change (RCC) Move

- **Region Crossing Change** - An operation on a knot diagram where all crossings adjacent to a chosen region are changed.

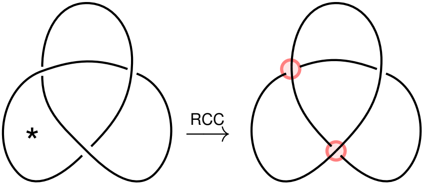
Region Crossing Change (RCC) Move

- **Region Crossing Change** - An operation on a knot diagram where all crossings adjacent to a chosen region are changed.



Region Crossing Change (RCC) Move

- **Region Crossing Change** - An operation on a knot diagram where all crossings adjacent to a chosen region are changed.



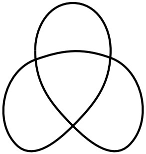
RCC Moves and Reconfiguration

Theorem: Any set of crossing changes in a knot diagram can be accomplished through RCC moves.

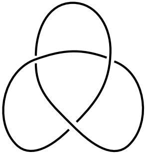
RCC Moves and Reconfiguration

Theorem: Any set of crossing changes in a knot diagram can be accomplished through RCC moves.

There are 2^c diagrams associated with a projection with c crossings.



Projection



Diagram

Combinatorial Reconfiguration

- **Combinatorics** - Problems involving the combination and permutation of sets of elements.
  - What crossing changes are accomplished by RCC moves in different combinations of regions?

Combinatorial Reconfiguration

- **Combinatorics** - Problems involving the combination and permutation of sets of elements.
  - What crossing changes are accomplished by RCC moves in different combinations of regions?
- **Reconfiguration** - Computational problems involving reachability or connectivity of state spaces.
  - What knot diagrams are connected via RCC moves? What properties of reachability are there for different diagrams?

Diameter of a Knot Diagram

The **distance** between two knot diagrams is the minimum number of RCC moves needed to move from one diagram to the other.

Diameter of a Knot Diagram

The **distance** between two knot diagrams is the minimum number of RCC moves needed to move from one diagram to the other.

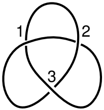
The **diameter** of a knot diagram is the maximum distance between any two diagrams with the same projection.

Diagram and Region Vectors

We can think of diagram vectors (black) as well as region vectors (red). The initial diagram is assigned the zero vector.

Diagram and Region Vectors

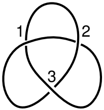
We can think of diagram vectors (black) as well as region vectors (red). The initial diagram is assigned the zero vector.



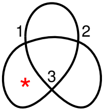
[0 0 0]

Diagram and Region Vectors

We can think of diagram vectors (black) as well as region vectors (red). The initial diagram is assigned the zero vector.



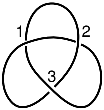
[0 0 0]



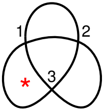
[1 0 1]

Diagram and Region Vectors

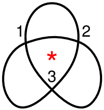
We can think of diagram vectors (black) as well as region vectors (red). The initial diagram is assigned the zero vector.



[0 0 0]



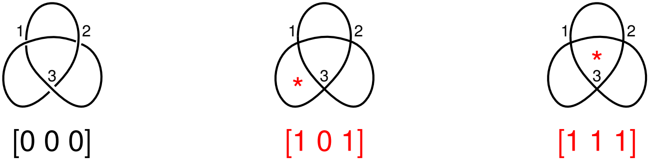
[1 0 1]



[1 1 1]

Diagram and Region Vectors

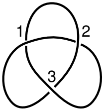
We can think of diagram vectors (black) as well as region vectors (red). The initial diagram is assigned the zero vector.



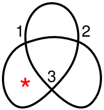
RCCs correspond to adding region vectors to diagram vectors.

Diagram and Region Vectors

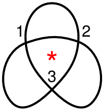
We can think of diagram vectors (black) as well as region vectors (red). The initial diagram is assigned the zero vector.



[0 0 0]



[1 0 1]



[1 1 1]

RCCs correspond to adding region vectors to diagram vectors.

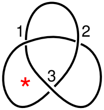
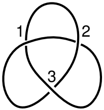
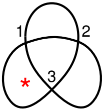


Diagram and Region Vectors

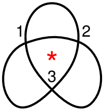
We can think of diagram vectors (black) as well as region vectors (red). The initial diagram is assigned the zero vector.



[0 0 0]

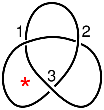


[1 0 1]



[1 1 1]

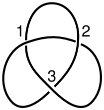
RCCs correspond to adding region vectors to diagram vectors.



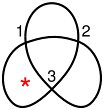
$[0\ 0\ 0] + [1\ 0\ 1]$

Diagram and Region Vectors

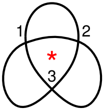
We can think of diagram vectors (black) as well as region vectors (red). The initial diagram is assigned the zero vector.



[0 0 0]

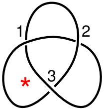


[1 0 1]



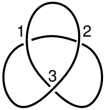
[1 1 1]

RCCs correspond to adding region vectors to diagram vectors.



[0 0 0] + [1 0 1]

$\xrightarrow{\text{RCC}}$



[1 0 1]

Diameter from Region Vectors

- **Basic Idea:** take combinations of increasing numbers of region vectors until all diagram vectors have been generated.

Diameter from Region Vectors

- **Basic Idea:** take combinations of increasing numbers of region vectors until all diagram vectors have been generated.

- The largest number of region vectors needed in a combination is the diameter.

Diameter from Region Vectors

- **Basic Idea:** take combinations of increasing numbers of region vectors until all diagram vectors have been generated.

- The largest number of region vectors needed in a combination is the diameter.

```
>>> d[3_1].ediameter()
level 0:
[000]

level 1:
[111]
[110]
[101]
[011]

level 2:
[001] = 111 + 110
[010] = 111 + 101
[100] = 111 + 011

The diameter is 2
```

Diameter from Region Vectors

Algorithm 1 Diameter()

Input: Region Vectors of a Knot Diagram
Output: Diameter of the Diagram

initial Diagram d.reached $\leftarrow$ TRUE
x $\leftarrow$ 0
while not all Diagrams reached **do**
    x $\leftarrow$ x + 1
    for all combinations of x region vectors **do**
        Diagram d $\leftarrow$ xor of vectors
        d.reached $\leftarrow$ TRUE
return x

Program So Far

Diagram
diameter (region vectors) : diameter of the diagram

Program So Far

Diagram
diameter (region vectors) : diameter of the diagram

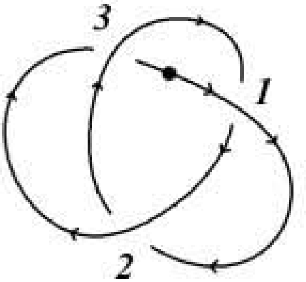
- Next up: compute the region vectors directly

Knot Codes

- **Knot Code** - A sequence encoding information about a knot diagram.

Knot Codes

- **Knot Code** - A sequence encoding information about a knot diagram.



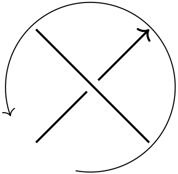
O1-U2-O3-U1-O2-U3-

PD Notation

- **Planar Diagram (PD) Notation** - A knot code describing how the arcs of the diagram intersect at each crossing.

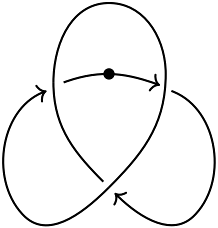
PD Notation

- **Planar Diagram (PD) Notation** - A knot code describing how the arcs of the diagram intersect at each crossing.



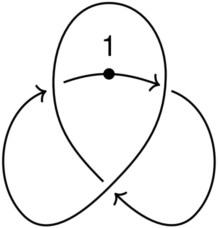
PD Notation

- Choose a starting point and orientation and label each arc with a number.



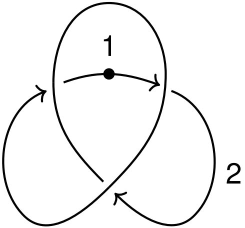
PD Notation

- Choose a starting point and orientation and label each arc with a number.



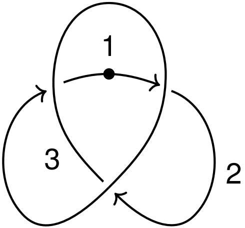
PD Notation

- Choose a starting point and orientation and label each arc with a number.



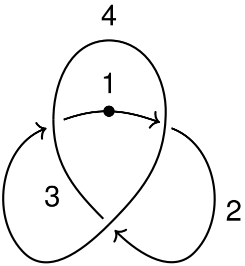
PD Notation

- Choose a starting point and orientation and label each arc with a number.



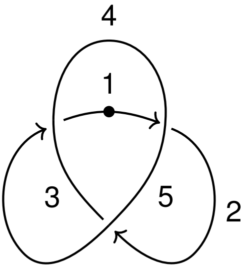
PD Notation

- Choose a starting point and orientation and label each arc with a number.



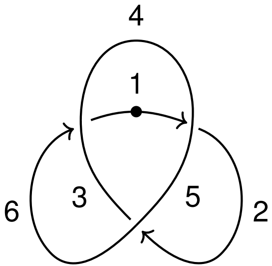
PD Notation

- Choose a starting point and orientation and label each arc with a number.



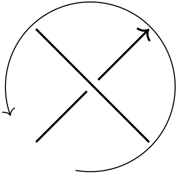
PD Notation

- Choose a starting point and orientation and label each arc with a number.



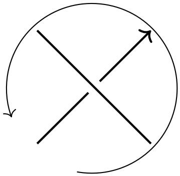
PD Notation

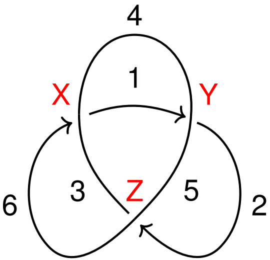
- To classify each crossing we list the arcs counterclockwise starting with the incoming under arc.



PD Notation

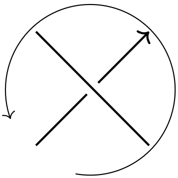
- To classify each crossing we list the arcs counterclockwise starting with the incoming under arc.

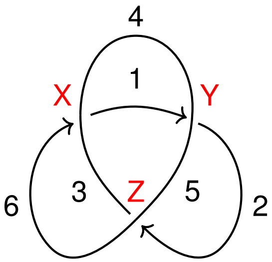




PD Notation

- To classify each crossing we list the arcs counterclockwise starting with the incoming under arc.





X(6,3,1,4),
Y(1,5,2,4),
Z(2,5,3,6).

PD Notation

KnotInfo: Table of Knots

Search Results

Export to CSV

PD Notation
[[1,5,2,4],[3,1,4,6],[5,3,6,2]]
[[4,2,5,1],[8,6,1,5],[6,3,7,4],[2,7,3,8]]
[[2,8,3,7],[4,10,5,9],[6,2,7,1],[8,4,9,3],[10,6,1,5]]
[[1,5,2,4],[3,9,4,8],[5,1,6,10],[7,3,8,2],[9,7,10,6]]
[[1,7,2,6],[3,10,4,11],[5,3,6,2],[7,1,8,12],[9,4,10,5],[11,9,12,8]]
[[1,8,2,9],[3,11,4,10],[5,1,6,12],[7,2,8,3],[9,7,10,6],[11,5,12,4]]
[[4,2,5,1],[8,4,9,3],[12,9,1,10],[10,5,11,6],[6,11,7,12],[2,8,3,7]]
[[1,9,2,8],[3,11,4,10],[5,13,6,12],[7,1,8,14],[9,3,10,2],[11,5,12,4],[13,7,14,6]]
[[2,10,3,9],[4,14,5,13],[6,12,7,11],[8,2,9,1],[10,8,11,7],[12,6,13,5],[14,4,1,3]]
[[1,9,2,8],[3,11,4,10],[5,1,6,14],[7,13,8,12],[9,3,10,2],[11,5,12,4],[13,7,14,6]]

[First] [Previous] [Next] [Last] **[Show All]** Total 2977 knots found: show 10 knots starting from 1 Go

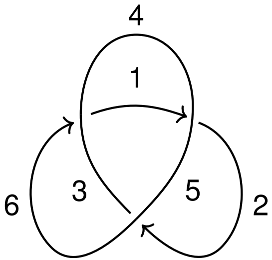
```
d = {}
d[3_1] = Diagram(((1,5,2,4),(5,3,6,2),(3,1,4,6)))
d[4_1] = Diagram(((6,2,7,1),(2,6,3,5),(8,3,1,4),(4,7,5,8)))
d[5_1] = Diagram(((1,6,2,7),(7,2,8,3),(3,8,4,9),(9,4,10,5),(5,10,6,1)))
d[5_2] = Diagram(((1,4,2,5),(7,2,8,3),(3,8,4,9),(5,10,6,1),(9,6,10,7)))
d[6_1] = Diagram(((1,4,2,5),(9,3,10,2),(3,9,4,8),(5,12,6,1),(11,6,12,7),(7,10,8,11)))
d[6_2] = Diagram(((1,4,2,5),(9,3,10,2),(3,9,4,8),(5,10,6,11),(11,6,12,7),(7,12,8,1)))
d[6_3] = Diagram(((4,2,5,1),(2,8,3,7),(8,4,9,3),(10,5,11,6),(6,11,7,12),(12,9,1,10)))
d[7_1] = Diagram(((1,8,2,9),(9,2,10,3),(3,10,4,11),(11,4,12,5),(5,12,6,13),(13,6,14,7),(7,14,8,1)))
d[7_2] = Diagram(((1,4,2,5),(9,2,10,3),(3,10,4,11),(5,14,6,1),(13,6,14,7),(7,12,8,13),(11,8,12,9)))
d[7_3] = Diagram(((6,2,7,1),(2,10,3,9),(10,4,11,3),(4,12,5,11),(12,6,13,5),(14,8,1,7),(8,14,9,13)))
d[7_4] = Diagram(((6,2,7,1),(2,12,3,11),(10,4,11,3),(4,10,5,9),(12,6,13,5),(14,8,1,7),(8,14,9,13)))
```

PD Notation to Region Vectors

- **Basic Concept:** Trace the regions of the knot diagram by traversing each arc and turning clockwise at its endpoint, keeping track of which crossings have been encountered.

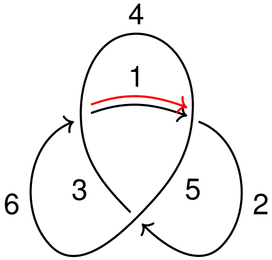
PD Notation to Region Vectors

- **Basic Concept:** Trace the regions of the knot diagram by traversing each arc and turning clockwise at its endpoint, keeping track of which crossings have been encountered.



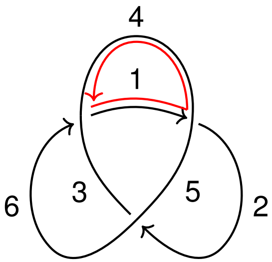
PD Notation to Region Vectors

- **Basic Concept:** Trace the regions of the knot diagram by traversing each arc and turning clockwise at its endpoint, keeping track of which crossings have been encountered.



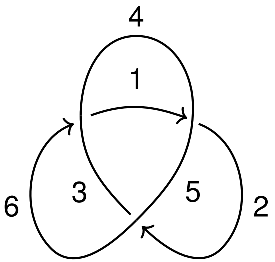
PD Notation to Region Vectors

- **Basic Concept:** Trace the regions of the knot diagram by traversing each arc and turning clockwise at its endpoint, keeping track of which crossings have been encountered.



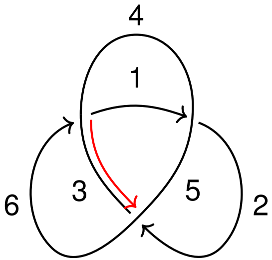
PD Notation to Region Vectors

- **Basic Concept:** Trace the regions of the knot diagram by traversing each arc and turning clockwise at its endpoint, keeping track of which crossings have been encountered.



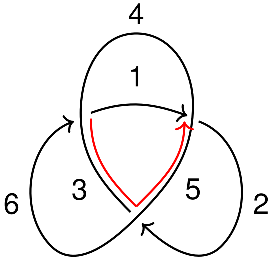
PD Notation to Region Vectors

- **Basic Concept:** Trace the regions of the knot diagram by traversing each arc and turning clockwise at its endpoint, keeping track of which crossings have been encountered.



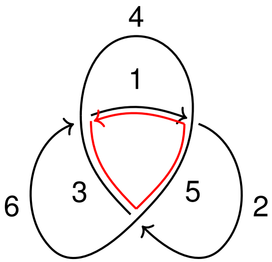
PD Notation to Region Vectors

- **Basic Concept:** Trace the regions of the knot diagram by traversing each arc and turning clockwise at its endpoint, keeping track of which crossings have been encountered.



PD Notation to Region Vectors

- **Basic Concept:** Trace the regions of the knot diagram by traversing each arc and turning clockwise at its endpoint, keeping track of which crossings have been encountered.



PD Notation to Region Vectors

$$\mathit{idx}: \left[\,(\underset{0}{6},\ \underset{1}{3},\ \underset{2}{1},\ \underset{3}{4})\ (\underset{4}{1},\ \underset{5}{5},\ \underset{6}{2},\ \underset{7}{4})\ (\underset{8}{2},\ \underset{9}{5},\ \underset{10}{3},\ \underset{11}{6})\,\right]$$

PD Notation to Region Vectors

$$idx: \big[\,(\underset{0}{6},\ \underset{1}{3},\ \underset{2}{1},\ \underset{3}{4})\ (\underset{4}{1},\ \underset{5}{5},\ \underset{6}{2},\ \underset{7}{4})\ (\underset{8}{2},\ \underset{9}{5},\ \underset{10}{3},\ \underset{11}{6})\,\big]$$

- *idx* // 4 = crossing number

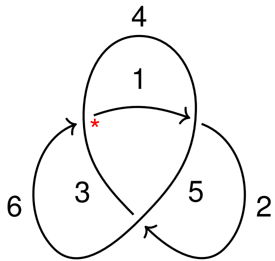
PD Notation to Region Vectors

$$\underset{idx:}{} [\,(\underset{0}{6}, \underset{1}{3}, \underset{2}{1}, \underset{3}{4})\ (\underset{4}{1}, \underset{5}{5}, \underset{6}{2}, \underset{7}{4})\ (\underset{8}{2}, \underset{9}{5}, \underset{10}{3}, \underset{11}{6})\,]$$

- *idx* // 4 = crossing number
- To turn clockwise, if *idx* % 4 = 0 go to *idx* + 3, otherwise go to *idx* - 1

PD Notation to Region Vectors

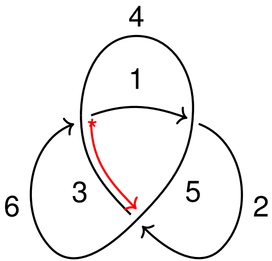
$$\begin{array}{r|cccc|cccc|cccc|}
 & (6, & \mathbf{3}, & 1, & 4) & (1, & 5, & 2, & 4) & (2, & 5, & 3, & 6) \\
idx: & 0 & \mathbf{1} & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11
\end{array}$$



[**1** 0 0]

PD Notation to Region Vectors

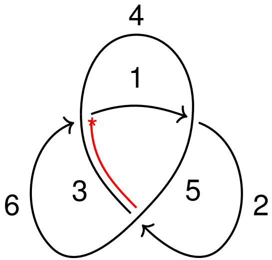
$$\begin{array}{lcccccccccccccccc} & [& (6, & \mathbf{3}, & 1, & 4) & (1, & 5, & 2, & 4) & (2, & 5, & \mathbf{3}, & 6) &] \\ \mathit{idx}: & & 0 & \mathbf{1} & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & \mathbf{10} & 11 & \end{array}$$



[1 0 **1**]

PD Notation to Region Vectors

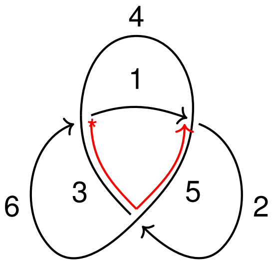
$$\begin{array}{r} \\ idx: \end{array} \left[\begin{array}{cccccccccccc} (6, & 3, & 1, & 4) & (1, & 5, & 2, & 4) & (2, & \mathbf{5}, & 3, & 6) \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \mathbf{9} \leftarrow & 10 & 11 \end{array} \right]$$



[1 0 1]

PD Notation to Region Vectors

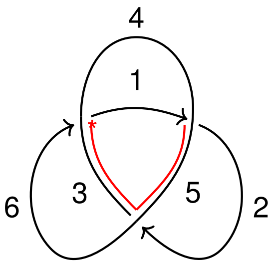
$$\mathit{idx}: \; [\,(\underset{0}{6}, \underset{1}{3}, \underset{2}{1}, \underset{3}{4})\; (\underset{4}{1}, \underset{\mathbf{5}}{\mathbf{5}}, \underset{6}{2}, \underset{7}{4})\; (\underset{8}{2}, \underset{\mathbf{9}}{\mathbf{5}}, \underset{10}{3}, \underset{11}{6})\,]$$



[1 **1** 1]

PD Notation to Region Vectors

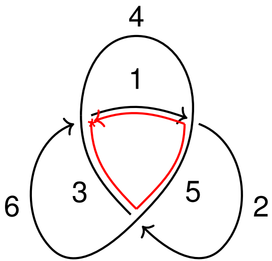
$$\mathit{idx}:\ \big[\ (\underset{0}{6},\ \underset{1}{3},\ \underset{2}{1},\ \underset{3}{4})\ \ (\underset{\mathbf{4}}{\mathbf{1}},\ {}_{\leftarrow}\underset{5}{5},\ \underset{6}{2},\ \underset{7}{4})\ \ (\underset{8}{2},\ \underset{9}{5},\ \underset{10}{3},\ \underset{11}{6})\ \big]$$



[1 1 1]

PD Notation to Region Vectors

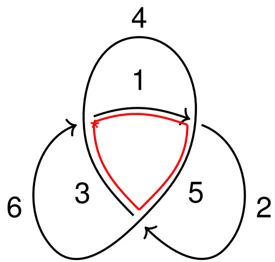
$$\begin{array}{rcccccccccccccc} & [& (6, & 3, & \mathbf{1}, & 4) & (\mathbf{1}, & 5, & 2, & 4) & (2, & 5, & 3, & 6) &] \\ idx: & & 0 & 1 & \mathbf{2} & 3 & \mathbf{4} & 5 & 6 & 7 & 8 & 9 & 10 & 11 & \end{array}$$



[**1** 1 1]

PD Notation to Region Vectors

$$\underset{idx:}{}\left[\,(\underset{0}{6},\ \underset{\mathbf{1}}{\mathbf{3}},\ \underset{\leftarrow 2}{1},\ \underset{3}{4})\ \ (\underset{4}{1},\ \underset{5}{5},\ \underset{6}{2},\ \underset{7}{4})\ \ (\underset{8}{2},\ \underset{9}{5},\ \underset{10}{3},\ \underset{11}{6})\,\right]$$



[1 1 1]

Pseudocode

Algorithm 2 Region_Vectors()

Input: PD Notation of a Knot Diagram
Output: Region Vectors of the Diagram

region_vectors $\leftarrow \emptyset$
assign indices to each arc by crossing
for $idx \leftarrow 0,1,\ldots,4c-1$ **do**
    vector $\leftarrow \vec{0}$
    while loop not complete **do**
        trace arc to other endpoint
        make a left turn at crossing onto new arc
        update *vector* with crossing adjacency
    add *vector* to *region_vectors*
return *region_vectors*

Program Now

Diagram
crossings : set of Crossings (PD code)
region_vectors() : region vectors of the diagram **diameter()** : diameter of the diagram

Crossing
i,**j**,**k**,**l** : arc number

Program Now

Diagram

crossings : set of Crossings (PD code)

region_vectors() : region vectors of the diagram
diameter() : diameter of the diagram

Crossing

i,**j**,**k**,**l** : arc number

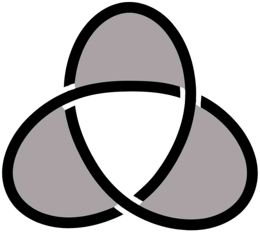
```
3_1 = 2
4_1 = 2
5_1 = 3
5_2 = 3
6_1 = 3
6_2 = 3
6_3 = 4
7_1 = 4
7_2 = 4
   |
   v
12_1284 = 6
12_1285 = 6
12_1286 = 6
12_1287 = 6
12_1288 = 6
```

Checkerboard Coloring

- **Checkerboard Coloring** - An assignment of the colors black and white to the regions of a knot diagram such that adjacent regions are assigned different colors.

Checkerboard Coloring

- **Checkerboard Coloring** - An assignment of the colors black and white to the regions of a knot diagram such that adjacent regions are assigned different colors.



Identifying Black and White Regions

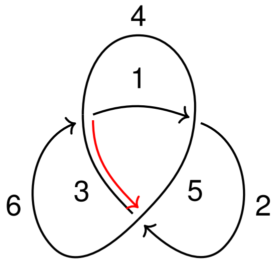
$$\underset{idx:}{}[\ (\underset{0}{6},\ \underset{\mathbf{1}}{\mathbf{3}},\ \underset{2}{1},\ \underset{3}{4})\ (\underset{4}{1},\ \underset{5}{5},\ \underset{6}{2},\ \underset{7}{4})\ (\underset{8}{2},\ \underset{9}{5},\ \underset{\mathbf{10}}{\mathbf{3}},\ \underset{11}{6})\ ]$$

- Must now deduce and track which indices are used to trace black (white) regions.

Identifying Black and White Regions

$$\underset{idx:}{} \left[\left(\underset{0}{6}, \underset{\mathbf{1}}{\mathbf{3}}, \underset{2}{1}, \underset{3}{4}\right) \left(\underset{4}{1}, \underset{5}{5}, \underset{6}{2}, \underset{7}{4}\right) \left(\underset{8}{2}, \underset{9}{5}, \underset{\mathbf{10}}{\mathbf{3}}, \underset{11}{6}\right) \right]$$

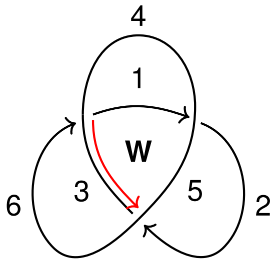
- Must now deduce and track which indices are used to trace black (white) regions.



Identifying Black and White Regions

$$_{idx:}\left[\,(\underset{0}{6},\ \underset{\mathbf{1}}{\mathbf{3}},\ \underset{2}{1},\ \underset{3}{4})\ \ (\underset{4}{1},\ \underset{5}{5},\ \underset{6}{2},\ \underset{7}{4})\ \ (\underset{8}{2},\ \underset{9}{5},\ \underset{\mathbf{10}}{\mathbf{3}},\ \underset{11}{6})\,\right]$$

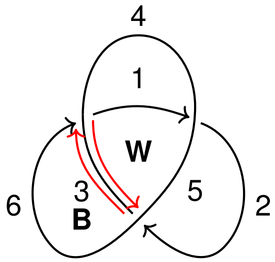
- Must now deduce and track which indices are used to trace black (white) regions.



Identifying Black and White Regions

$$\underset{idx:}{}[\,(\underset{0}{6},\ \underset{\mathbf{1}}{\mathbf{3}},\ \underset{2}{1},\ \underset{3}{4})\ (\underset{4}{1},\ \underset{5}{5},\ \underset{6}{2},\ \underset{7}{4})\ (\underset{8}{2},\ \underset{9}{5},\ \underset{\mathbf{10}}{\mathbf{3}},\ \underset{11}{6})\,]$$

- Must now deduce and track which indices are used to trace black (white) regions.



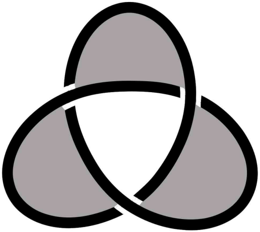
More Pseudocode

Algorithm 3 Black_White()

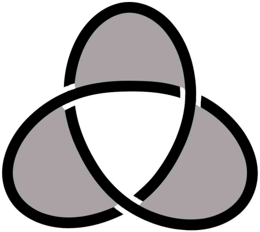
Input: PD Notation of a Knot Diagram
Output: Black and White Region Vectors

```
black_indices ← {0,2}
white_indices ← {1,3}
black_vectors, white_vectors ← ∅
while not all indices used do
    trace black (white) region, noting which indices are used
    and updating white (black) indices
    add region vector to proper set
return black_vectors + white_vectors
```

Black_White()



Black_White()



```
>>> d[3_1].black_white()
Diagram contains 3 black regions and 2 white regions

black region vectors:
110
101
011

white region vectors:
111
111
```

Final Program

Diagram
crossings : set of Crossings (PD code)
region_vectors() : region vectors of the diagram **ediameter()** : returns diameter and prints combination of vectors realizing each diagram **black_white()** : returns black and white region vectors **ineff_sets()** : returns ineffective sets of the diagram **dtCode()**, **gaussCode()** : returns knot code **alexanderPolynomial()** : calculates knot invariant

Crossing
i,**j**,**k**,**l** : arc number
crossing_change() : performs a crossing change