Knots and links

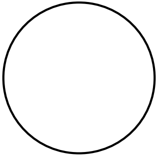
Ádám Gyenge

Department of Algebra and Geometry
BUTE

2025

Knots

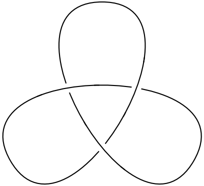
- A **knot** is an embedding of a circle in 3-dimensional space, $\mathbb{R}^3$.
- Knots are studied up to **equivalence**, meaning they can be deformed into one another without cutting.
- The **unknot** is a trivial knot that can be deformed into a simple loop.



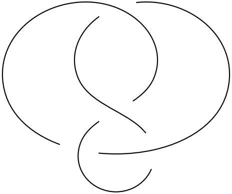
Unknot

Examples of Knots

- **Trefoil knot**:



- **Figure-eight knot**:



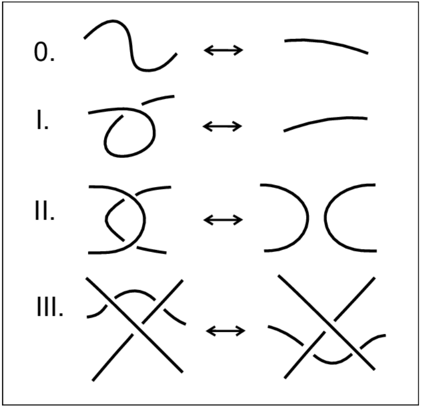
Knot Diagrams

- A useful way to visualise and manipulate knots is to project the knot onto a plane
- A small change in the direction of projection will ensure that it is one-to-one except at the double points, called crossings, where the "shadow" of the knot crosses itself once.
- A **knot diagram** is a two-dimensional projection of a knot onto a plane, with crossing information indicated (over-strand vs under-strand).

Reidemeister Moves

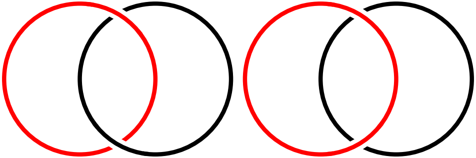
Theorem (Reidemeister, 1927)

Two knots are equivalent if and only if their diagrams are related by a sequence of the following steps.



Links

- A **link** consists of multiple knots that may be intertwined or separated.
- Examples: **Hopf link** and **unlink**



Knot Invariants

Definition

A **knot invariant** is a "quantity" that is the same for equivalent knots.

For example, if the invariant is computed from a knot diagram, it should give the same value for two knot diagrams representing equivalent knots. Warning: non-equivalent knots/links may have identical invariants.

Similarly, **link invariants**.

Example

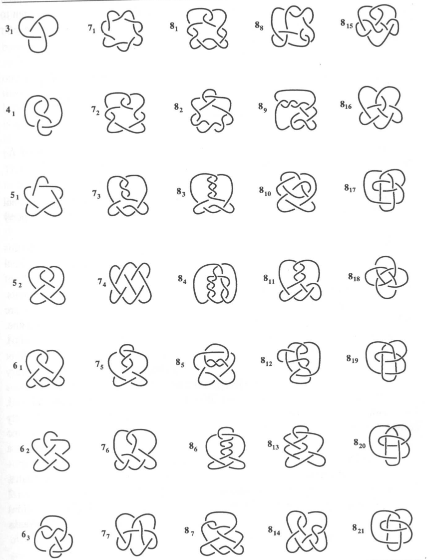
The number of components is a link invariant.

Crossing number

- The **crossing number** of a knot is the minimal number of crossings needed for a diagram of the knot.
- It is a knot invariant.

Crossing nr.	Nr. of knots
3	1
4	1
5	2
6	3
7	7
8	21
9	49
10	165
11	552
12	2176

TABLE 1.1. The Knot Table to Eight Crossings



Unknotting Number

- The **unknotting number** of a knot is the minimum number of crossing changes needed to transform the knot into the trivial knot (the *unknot*).
- Example:
  - The unknot has an unknotting number of 0.
  - The trefoil knot has an unknotting number of 1.
- Also a knot invariant.
- Issue: difficult to compute.

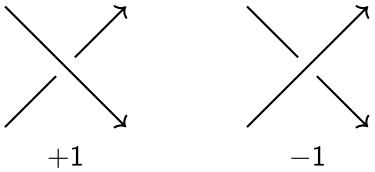
Writhe of a diagram

Oriented link: every component is given an orientation.

Definition

The **writhe** $w(D)$ of an oriented link diagram D is the sum of the signed crossings in D:

$$w(D) = \sum_{\text{crossings}} \text{sign}(\text{crossing})$$



Writhe of a link

Lemma

- *Reidemeister moves of Type II and Type III do not affect the writhe*
- *Move Type I, however, increases or decreases the writhe by 1.*

This implies that the writhe of a knot is not an isotopy invariant of the knot itself — only the diagram.
By a series of Type I moves one can set the writhe of a diagram for a given knot to be any integer at all.

Linking number

Consider a link of two oriented components: $L_1 \sqcup L_2$

Definition

Linking number:

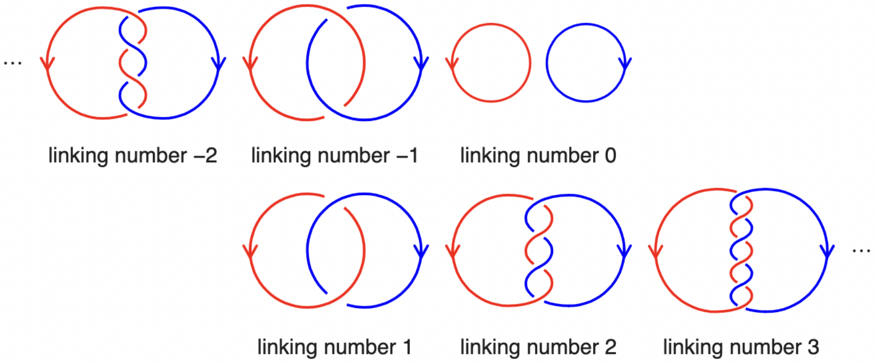
$$\mathrm{Lk}(L_1, L_2) = \frac{1}{2} \sum_{\text{crossings of } L_1 \text{ and } L_2} \mathrm{sign}(\text{crossing}),$$

where the sum runs over all crossings between L_1 and L_2.

Properties:

- The linking number is an integer invariant of the link.
- $\mathrm{Lk}(L_1, L_2) = \mathrm{Lk}(L_2, L_1)$

Examples



...
linking number –2
linking number –1
linking number 0
linking number 1
linking number 2
linking number 3
...

Twist of a ribbon

A **ribbon** (or strip) is the combination of a smooth space curve and a unit normal vector.

Definition

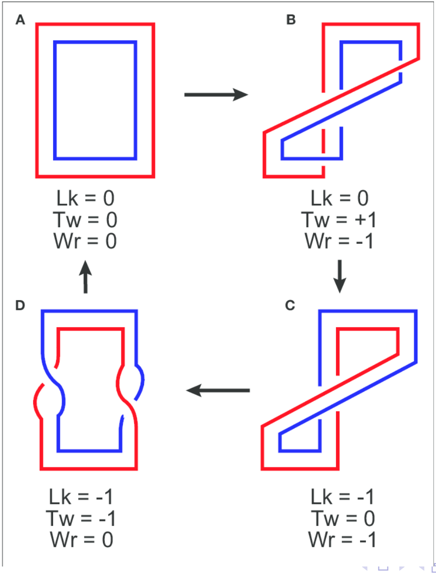
The **twist** Tw of a ribbon is a measure of how one edge of the ribbon twists around the other. For a ribbon defined by a curve $\gamma(t)$ with a unit normal vector field $v(t)$ along it, the twist is given by:

$$\mathrm{Tw} = \frac{1}{2\pi} \int_0^L \left(v \times \frac{\mathrm{d}v}{dt} \right) \cdot \frac{\mathrm{d}\gamma}{dt}\, dt,$$

where:

- $\frac{\mathrm{d}\gamma}{dt}$: unit tangent vector to γ.
- L: Length of the curve $\gamma(t)$.

Example



More generally

Theorem (Călugăreanu-White-Fuller)

$$Lk = Wr + Tw$$

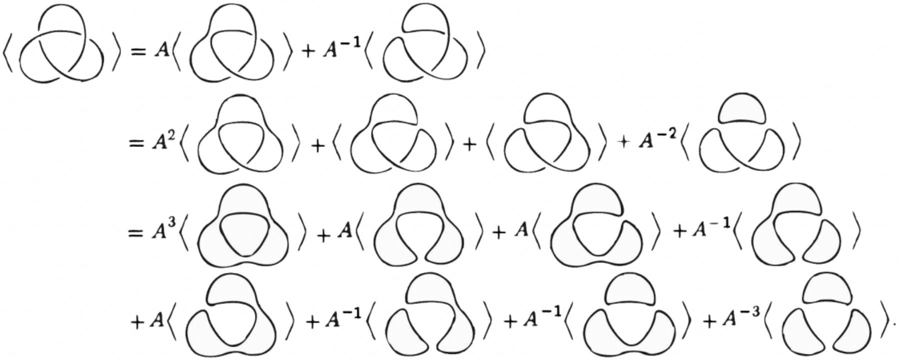
The Kauffman Bracket

- The **Kauffman bracket** is a polynomial invariant of a knot or link defined through a recursive process.
- It assigns to a knot diagram D a Laurent polynomial $\langle D \rangle \in \mathbb{Z}[A, A^{-1}]$ via the following rules:

Rules for the Kauffman Bracket

1. $\langle \bigcirc \rangle = 1$
2. $\langle \times \rangle = A\langle \mid \; \mid \rangle + A^{-1}\langle = \rangle$
3. $\langle \bigcirc \cup L \rangle = (-A^2 - A^{-2})\langle L \rangle$

Kauffmann bracket of the trefoil



Hence,

$$\langle D \rangle = (-A^2 - A^{-2})(-A^5 - A^{-3} + A^{-7}).$$

Bracket and Reidemeister moves

Lemma

1. *Type I Reidemesiter move changes the bracket in the following way*

$\Big\langle$  $\Big\rangle = -A^3 \Big\langle \,|\, \Big\rangle$

2. *Type II and Type III Reidemesiter moves do not change the bracket*

The Jones Polynomial

Theorem (Vaughan Jones, 1980s)

Let D be a diagram of an oriented link L. Then the expression

$$(-A)^{-3w(D)}\langle D\rangle$$

is an invariant of the oriented link.

Definition

The **Jones polynomial** of an oriented link L is

$$V_L(t) = (-A^3)^{-w(D)}\langle D\rangle\big|_{A=t^{-1/4}} \in \mathbb{Z}[t^{-1/2}, t^{1/2}],$$

a Laurent polynomial in $t^{1/2}$.

Properties:

- $V_L(t)$ distinguishes many knots but not all.
- For the unknot, $V_{\text{unknot}}(t) = 1$.
- For the trefoil knot:

$$V_{\text{trefoil}}(t) = t + t^3 - t^4.$$

- For links with odd number of components, including knots, it contains only integer powers of t.

Another characterisation of the Jones polynomial

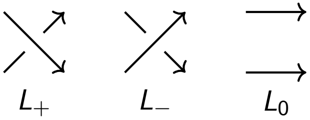
The Jones polynomial invariant is a function

$$V : \{\text{Oriented knots in } \mathbb{R}^3\} \to \mathbb{Z}[t^{1/2}, t^{-1/2}]$$

such that

1. $V_{\bigcirc}(t) = 1$
2. whenever three oriented links L_+, L_- and L_0 are the same except in the neighbourhood of a point where they are as shown below, then

$$tV_{L_+}(t) - t^{-1}V_{L_-}(t) = (t^{1/2} - t^{-1/2})V_{L_0}(t)$$



HOMFLY polinomial (1985)

The **HOMFLY polynomial** invariant is a function

$$\overline{V} : \{\text{Oriented knots in } \mathbb{R}^3\} \to \mathbb{Z}[t^{1/2}, t^{-1/2}, q]$$

such that ... (similar rules)

Advantages:

1. Two variable polynomial
2. With setting $q = 1$, we get back the Jones polynomial
3. Can recognize/differentiate more knots (so it is a finer invariant)

Applications of Knot Theory

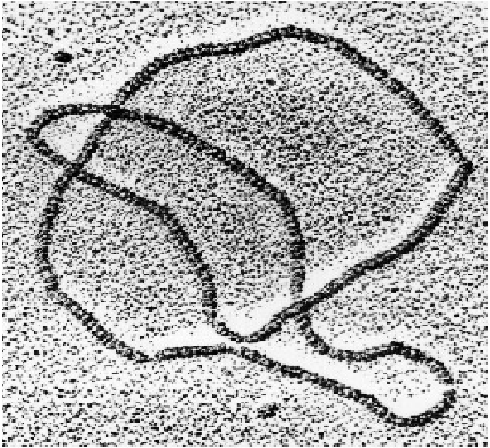
Knot theory has applications across many fields:

- **Biology:** Understanding DNA topology (e.g., supercoiling, recombination, and knotted proteins).
- **Chemistry:** Designing and analyzing molecular knots for nanoscale structures.
- **Physics:**
  - Study of magnetic flux tubes in plasma.
  - Topological quantum field theory and anyons in quantum computing.
- **Engineering:** Knot analysis in rope strength and stability.

Knots in Biology

- Viruses attack cells in order to alter the DNA inside them.
- To do this, they bring closer certain parts of the DNA, then cut them and stick them back together differently, in such a way that the molecule of DNA is transformed into a knot.
- One of the essential aspects of the struggle against viruses is to recognize the signature of different viruses by their effects on the DNA.
- One can characterize these effects by the type of knot which results from the action of the virus.
- But then, it is necessary to be able to recognize the knot in question if one wants to find out which virus it is.
- It is here that the work of mathematicians enters into play

DNA trefoil



Knots in Python

With the SnapPy library:

```
import snappy

# Load the trefoil knot
trefoil = snappy.Manifold("3_1")
```

The `snappy.Manifold("3_1")` loads the trefoil knot, identified as 3_1 in the Rolfsen knot table.

Knots in Sage/Cocalc

```
# Define the trefoil knot (3_1)
trefoil = Knot(3, 1)

# Compute and display knot invariants
print("Name:", trefoil.name())
print("Alexander Polynomial:", trefoil.alexander_polynomial())
print("Jones Polynomial:", trefoil.jones_polynomial())
print("HOMFLY Polynomial:", trefoil.homfly_polynomial())

# Draw the knot diagram
trefoil.plot()
```