Algorithms in Topology

Today’s Plan:

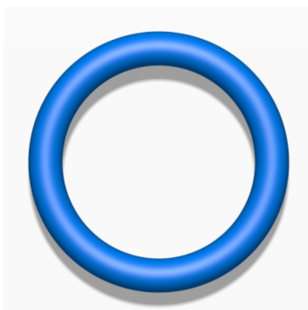
Review the history and approaches to two fundamental problems:

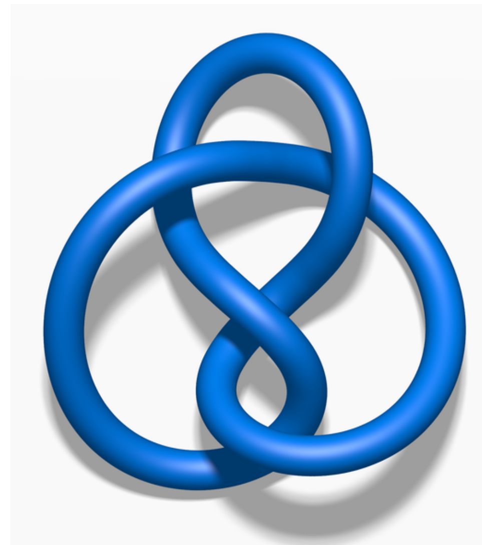
Unknotting

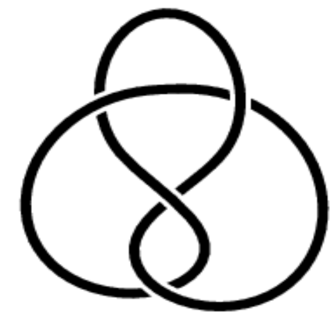
Manifold Recognition and Classification

What are knots?

Knots are closed loops in R^3, up to isotopy



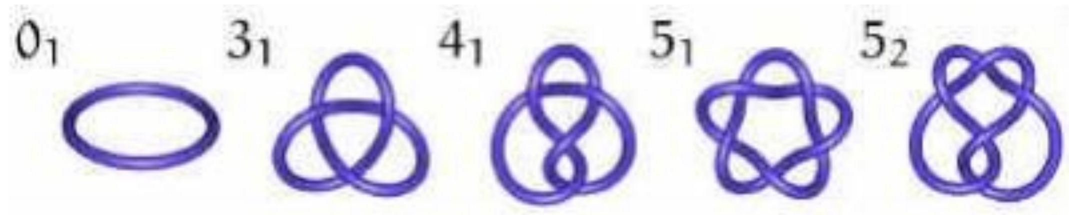




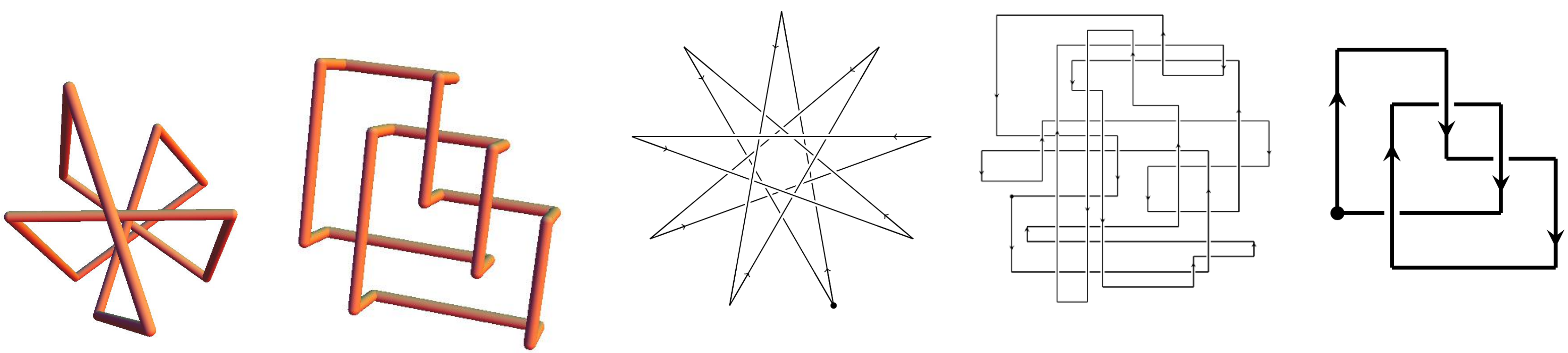
Diagram

$$S^1 \xhookrightarrow[\text{Embedding}]{} \mathbb{R}^3 \xrightarrow[\text{Projection}]{} \mathbb{R}^2$$

What are knots?

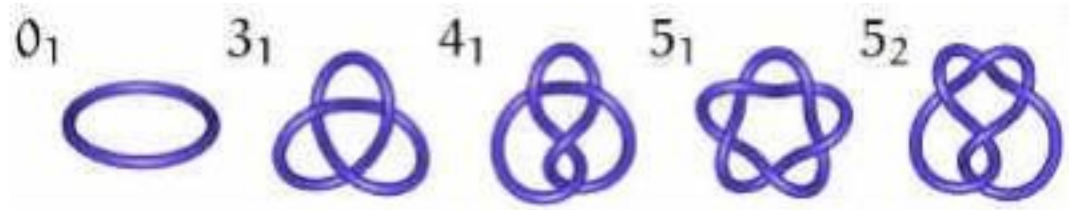


We can study smooth or polygonal knots. These give equivalent theories, but polygonal knots are more natural for computation.



What are knots?

Knots are closed loops in space, up to isotopy



We can study smooth or polygonal knots. These give equivalent theories, but polygonal knots are more natural for computation.

For algorithmic purposes, we can explicitly describe a knot as a polygon in Z^3.

K = {(0,0,0), (1,2,0), (2,3,8), ... , (0,0,0)}

We can also use several equivalent descriptions.

Some Basic Questions

- Can we classify knots?
- Can we recognize a particular knot, such as the unknot?
- How hard is it to recognize a knot?
- Does topology say something new about complexity classes?
- Do undecidable problems arise in the study of knots and 3-manifolds.
- Does the study of topological and geometric algorithms lead to new insight into classical problems? (Isoperimetric inequalities, P=NP? NP=coNP?)

Basic Questions about Manifolds

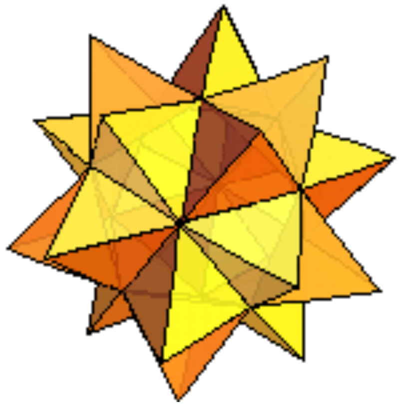
- Can we classify manifolds?
- Can we recognize a particular manifold, such as the sphere?
- How hard is it to recognize a manifold? (What is the complexity of an algorithm)
- What undecidable problems arise in the study of knots and 3-manifolds?
- Does the study of topological and geometric algorithms lead to new insight into classical problems? (Isoperimetric inequalities, P=NP? NP=coNP?)

Describing Surfaces and 3-Manifolds

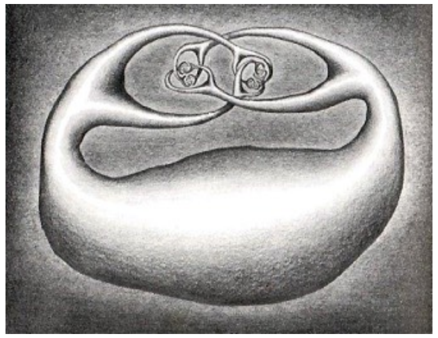
What type of surfaces and manifolds do we consider?
There are three main categories to choose from:

Smooth

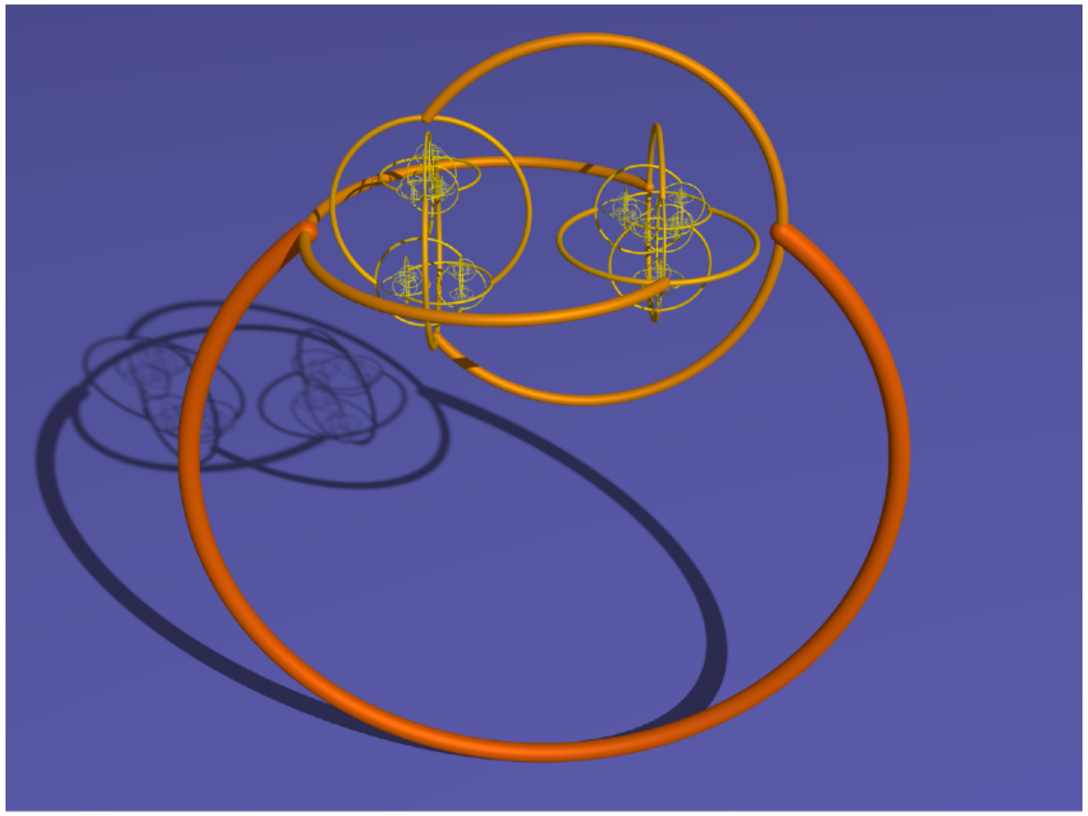
Piecewise Linear

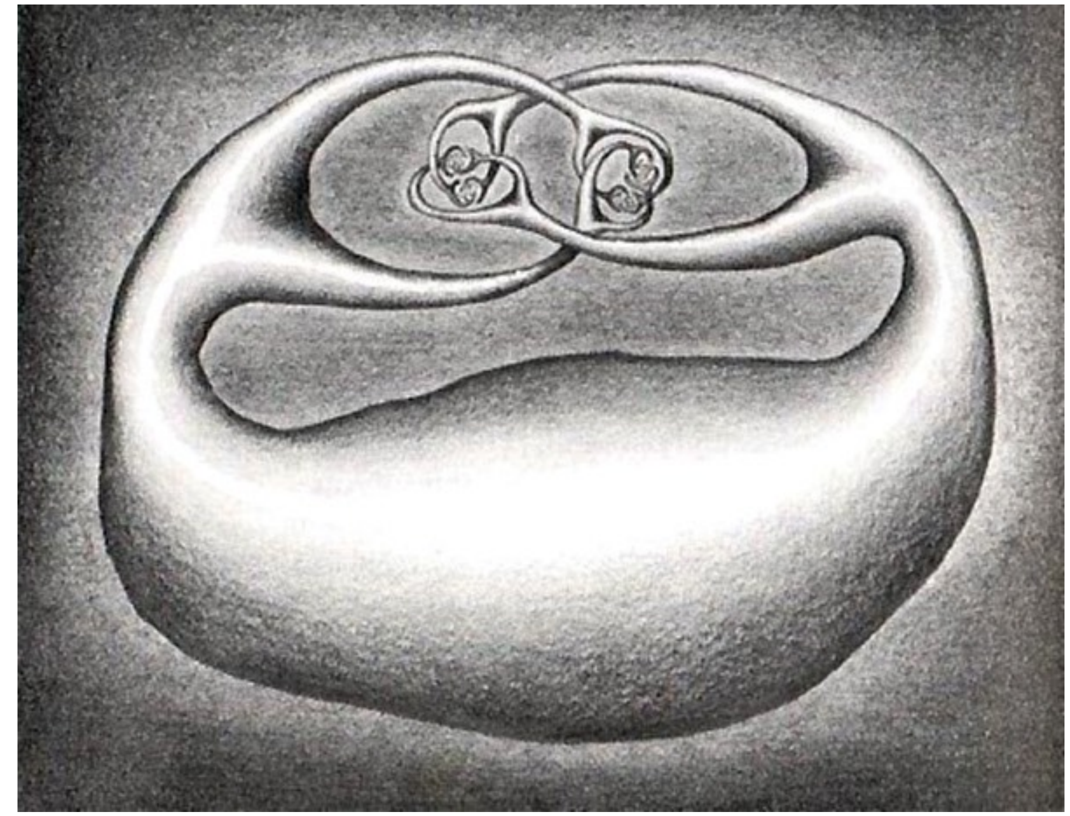


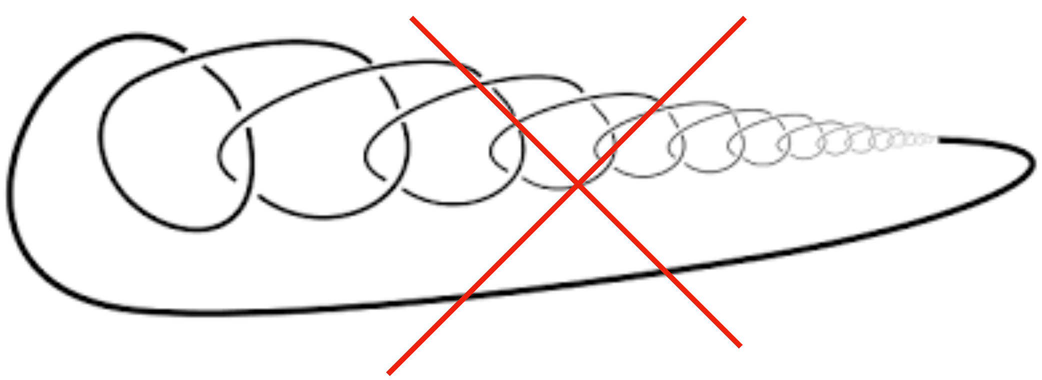
Continuous

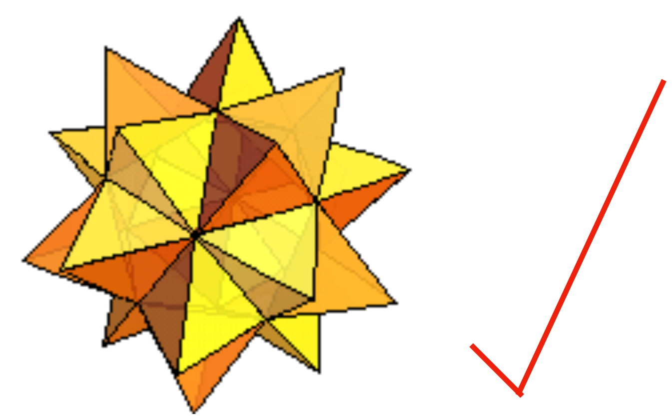


Describing Surfaces and 3-Manifolds







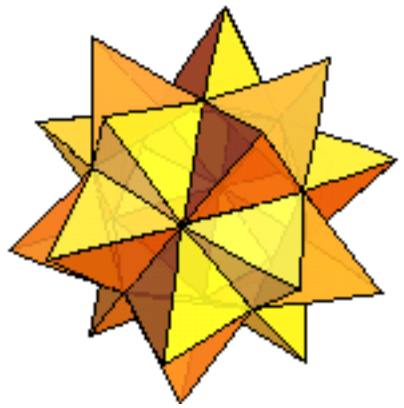


The continuous theory allows for more pathological examples.

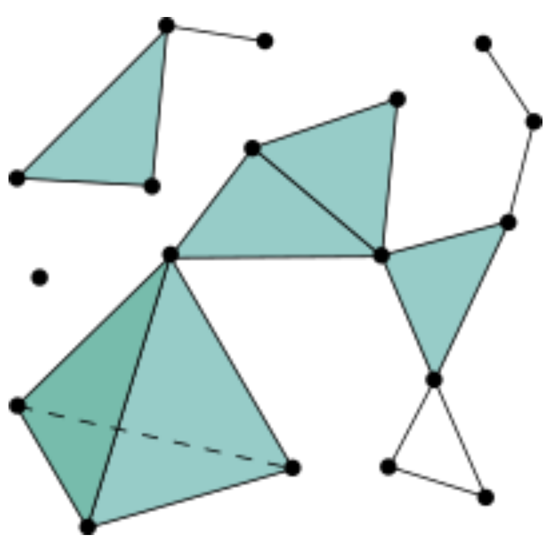
For algorithms, Piecewise Linear manifolds give the most natural setting

Describing Surfaces and 3-Manifolds

Piecewise Linear



A manifold is described as a triangulation, or a *simplicial complex*.



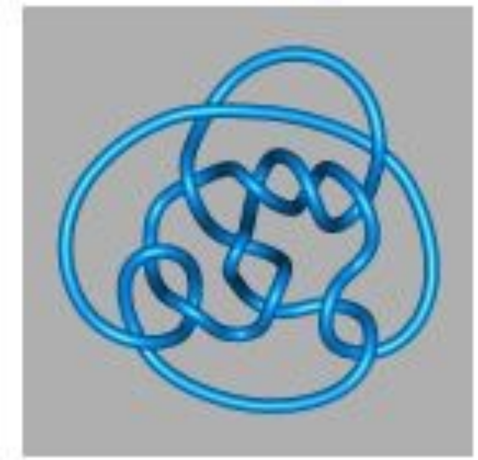
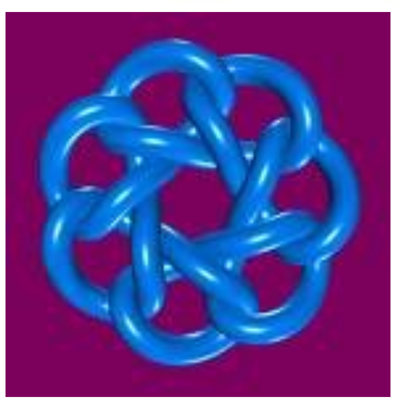
Simplicial complex: A collection of simplices satisfying:
Every face of a simplex from K is also in K
The intersection of any two simplices in K is a face of each.

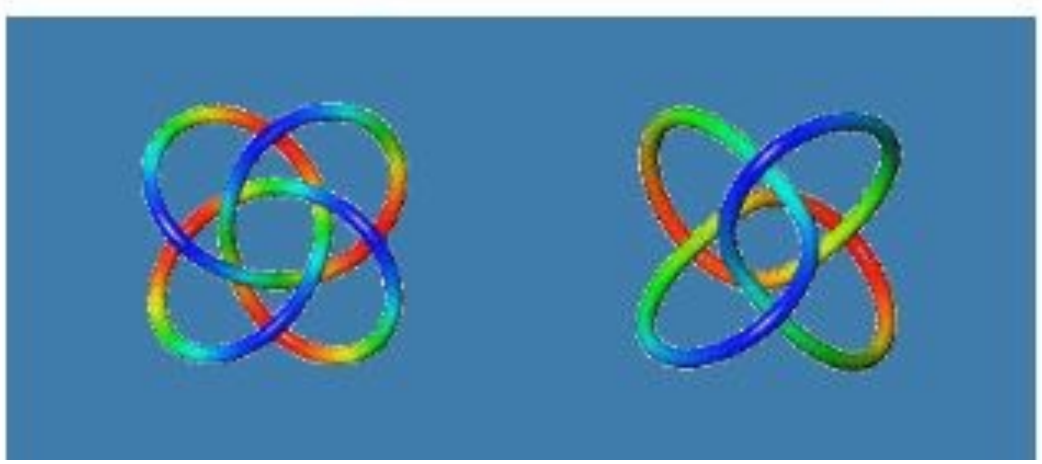
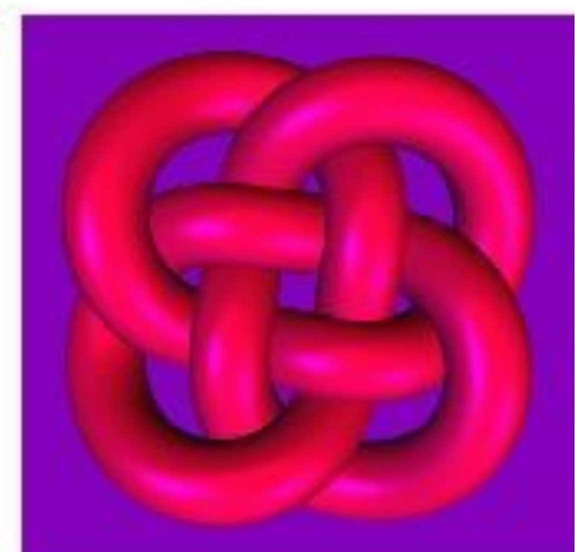
PL n-Manifold: A simplicial complex satisfying properties that ensure it is locally homeomorphic to R^n.
The link of every face is a sphere of appropriate dimension.

Note: This is one motivation for the problem of determining whether a given simplicial complex (the link) is a sphere.

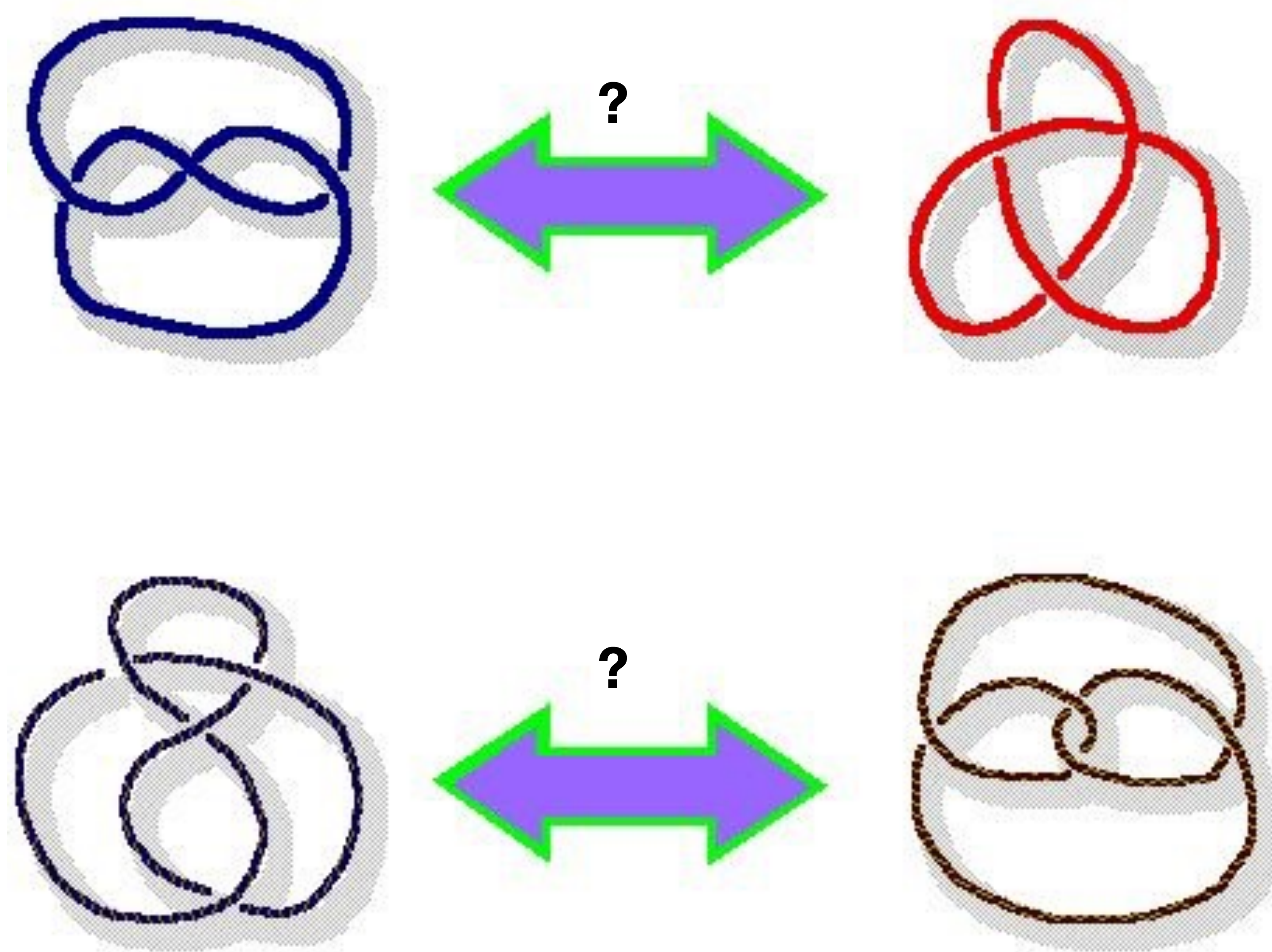
Can we recognize and classify knots?



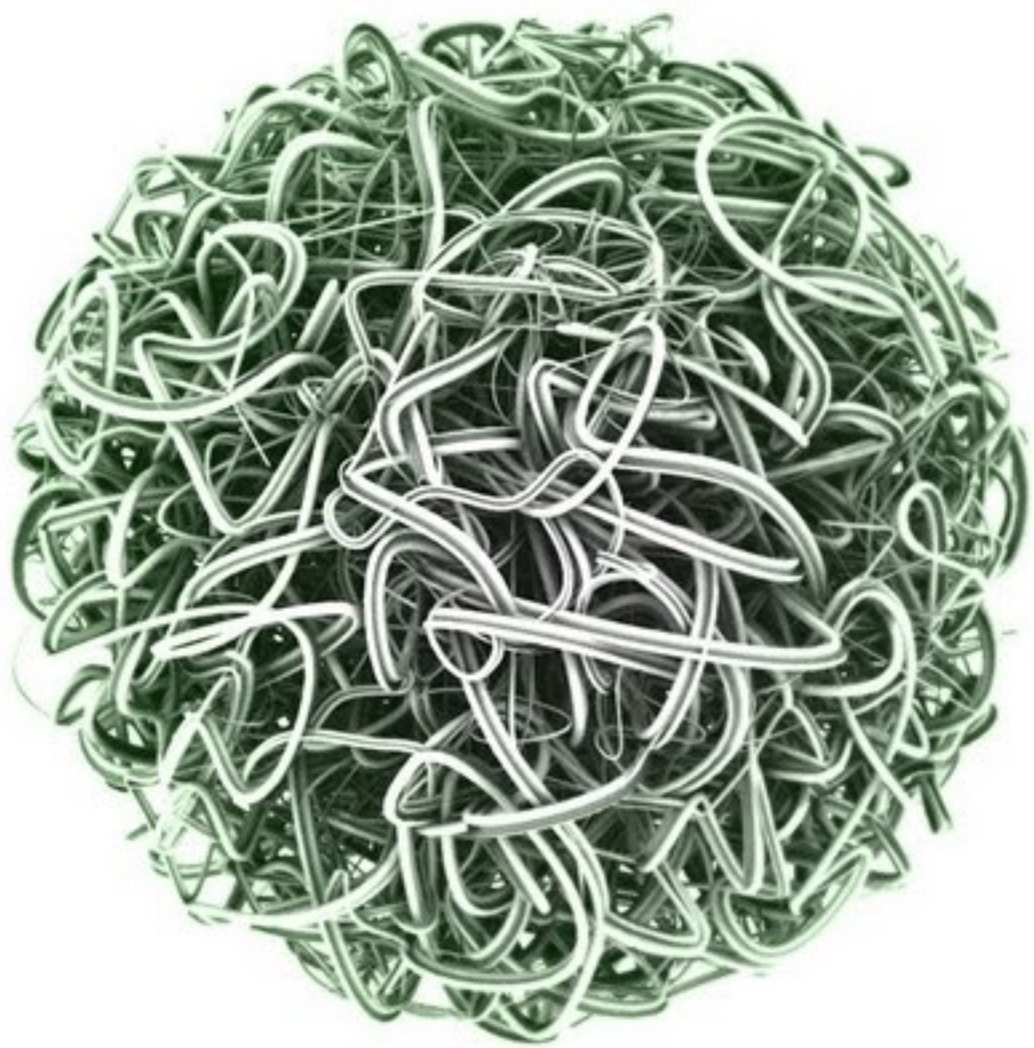




Can we recognize and classify knots?

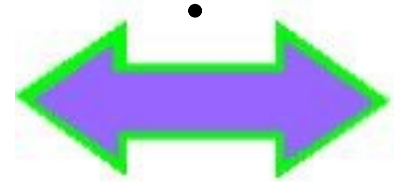


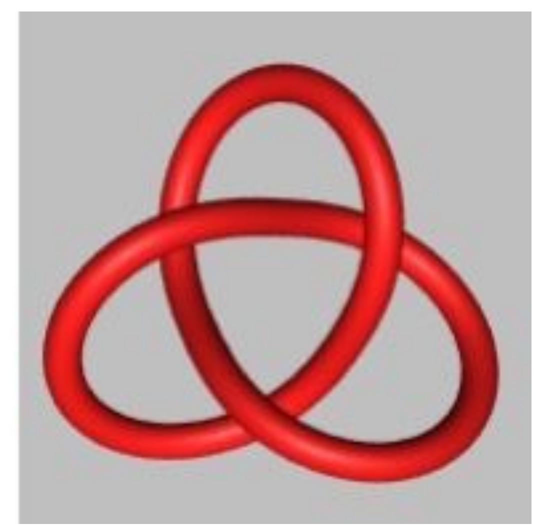
Knots can be drawn in different ways.
How can we tell, systematically, if two diagrams give the same knot?

Are these knots the same?

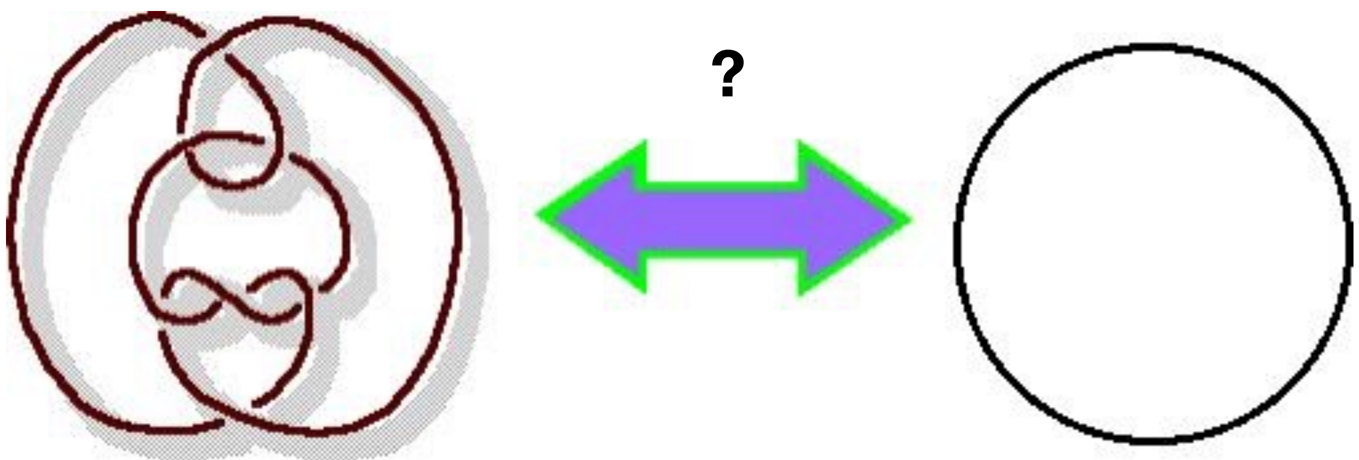


?





Even unknots can be drawn in deceiving ways

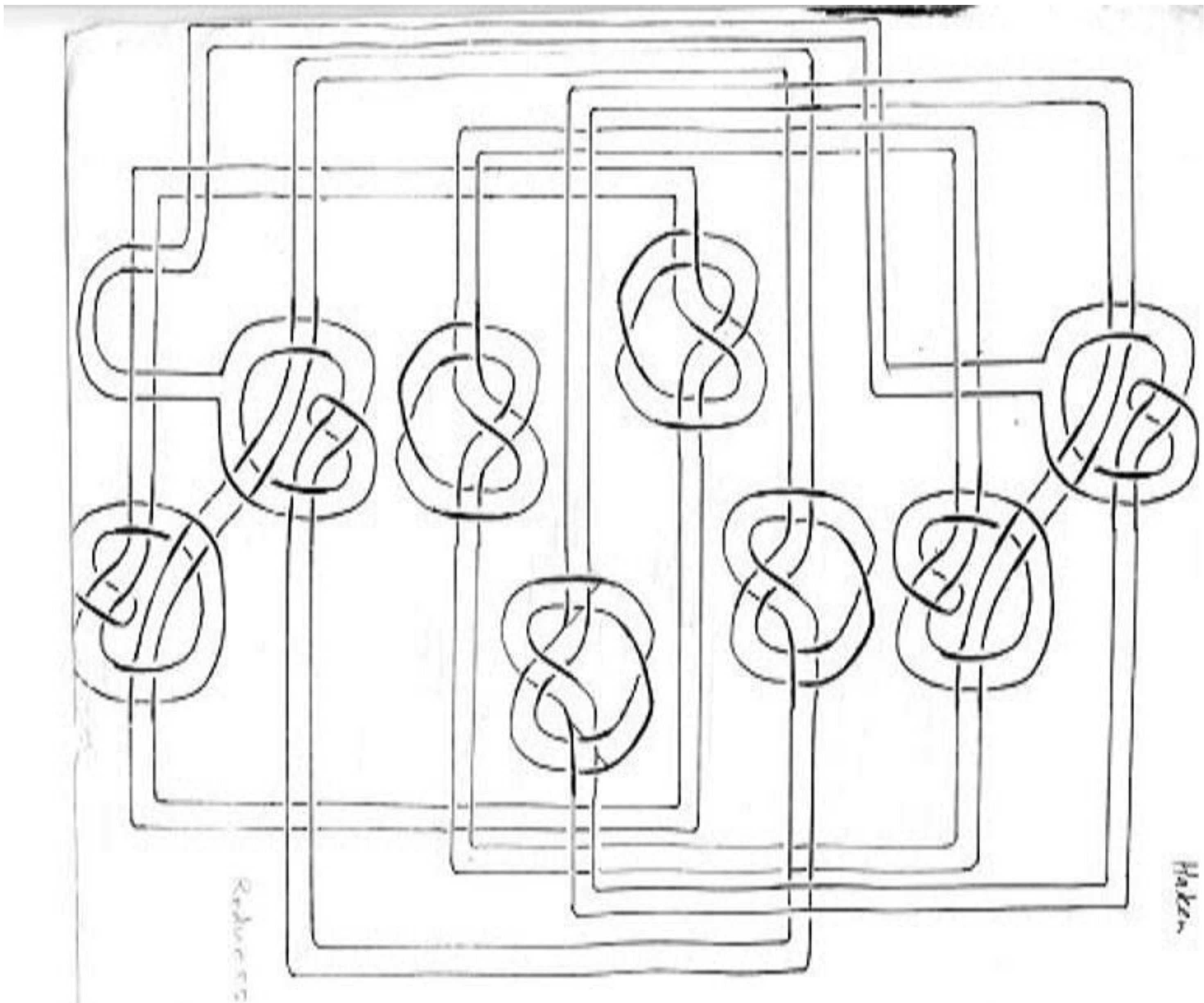


Hard to Recognize Unknots

Hard to Recognize Unknots

Haken's Unknot

Hard to Recognize Unknots



Another Haken Unknot

Some basic decision Problems for Knots and Links

Problem: UNKNOTTING
Instance: A knot K in S^3.
Question: Is K unknotted?

Problem: SPLIT LINK
Instance: A link L in S^3 with complement M_L.
Question: Does M_L contain a 2-sphere that separates the components of L?

Problem: KNOT GENUS
Instance: A knot K in S^3 and an integer g.
Question: Does K bound a surface of genus at most g?

Problem: KNOT RECOGNITION
Instance: A pair of knots K_1 and K_2 in S^3.
Question: Are K_1 and K_2 equivalent knots?

Haken's approach gives algorithms for each of these.

Some basic decision Problems for Knots and Links

Problem: UNKNOTTING
Instance: A knot K in S^3.
Question: Is *K* unknotted?

Problem: SPLIT LINK
Instance: A link *L* in S^3 with complement M_L.
Question: Does M_L contain a 2-sphere that separates the components of *L*?

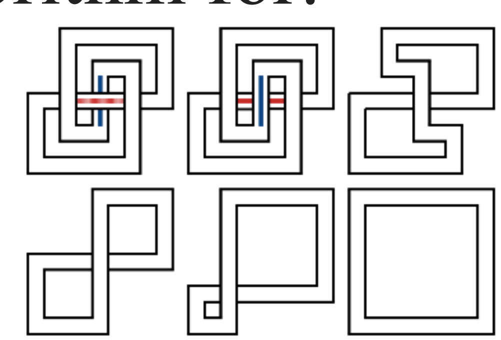
Problem: KNOT GENUS
Instance: A knot K in S^3 and an integer *g*.
Question: Does K bound a surface of genus at most *g*?

Problem: KNOT RECOGNITION
Instance: A pair of knots K_1 and K_2 in S^3.
Question: Are K_1 and K_2 equivalent knots?

Haken's approach gives algorithms for each of these.

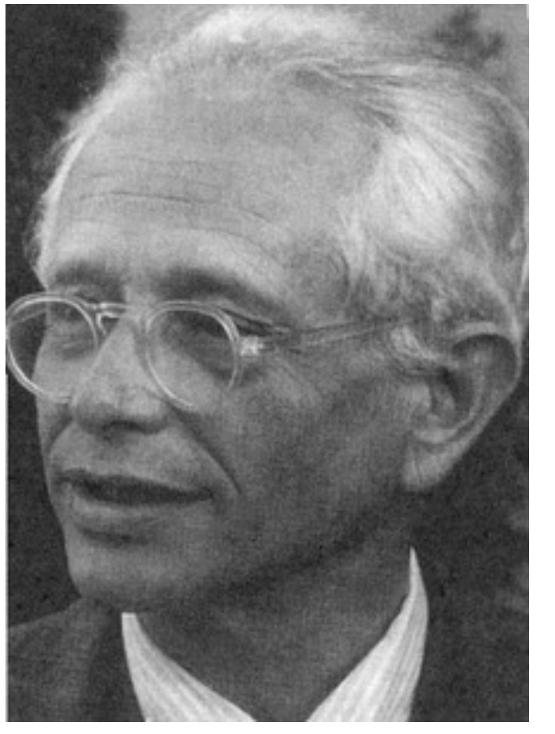
But not for everything. For example, we don't have an algorithm for:

Problem: UNKNOTTING NUMBER
Instance: A knot K in S^3 and an integer *n*.
Question: Does K have unknotting number at most *n*?



Some history

UNKNOTTING has historical connections to the foundations of theoretical computer science.



Max Dehn (1878 - 1952)

Wrote one of the first topology books (1907)

Proposed the search for a procedure to determine if a curve is knotted (1910).

This predated the definition of an ***algorithm***.

In 1961 Haken published a proof of the UNKNOTTING problem

It took about 50 years to find. (Dehn 1910 to Haken 1961)

Theorem (Haken) There is an algorithmic procedure to

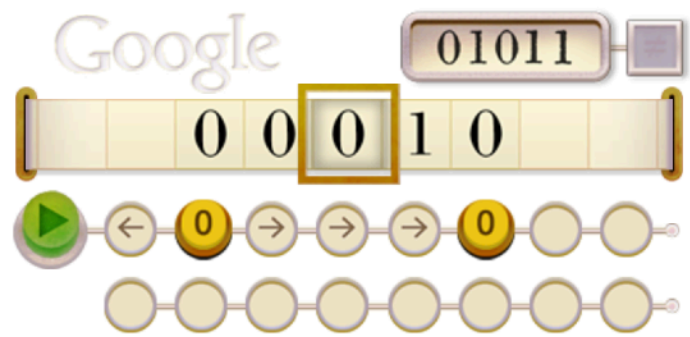
1. Recognize the Unknot
2. Classify knots
3. Compute the genus of a knot
4. Determine if a link is split

We will look at the ideas he used to prove this theorem.

We will also look at the running times of such algorithms.
How practical are they?

Turing machines specify algorithms

An algorithm is a procedure to solve a class of problems



This is made precise using the idea of a **Turing machine**

Formal definition:

Turing Machine Symbols

The symbolic definition of a Standard Turing Machine is:

$\mathbf{M} = (\mathbf{Q}, \Sigma, \Gamma, \delta, q_0, \mathbf{B}, \mathbf{F})$

M stands for the Turing Machine.

Q is a set of states within the Turing Machine.

Σ is called the **input alphabet.** This is a set of symbols that the Turing Machine will be working with. The input alphabet includes all of the characters in the tape alphabet except B.

Γ is called the **tape alphabet.** The tape alphabet are the set of symbols that appear on the tape, including B.

δ is the transition function. δ is defined as $\mathbf{Q} \times \Gamma = \mathbf{Q} \times \Gamma \times \{\mathbf{L}, \mathbf{R}\}$
This means that when the read head encounters a symbol on the tape, it can change state, write a symbol on the tape and move either left L, or right R. The x symbol in the formula means that all of the elements from each set is used. This is called the ***Cartesian Product.***

B is a blank symbol.

q_0 is the initial, or starting state.

F is a set of final states.

Figure 2

How can we think about an algorithm?

An algorithm is an unambiguous procedure to solve a class of problems.

We focus on algorithms for "decision problems"

A **decision problem** formulates a yes-no question for an input value

Examples

GRAPH 3-COLOR
Instance: A graph.
Question: Can the graph be 3-colored?

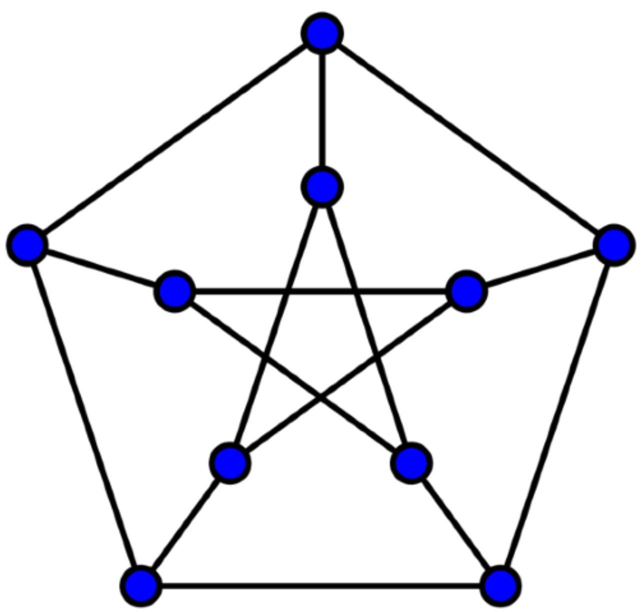
PRIME
Instance: An integer.
Question: Is the integer prime?

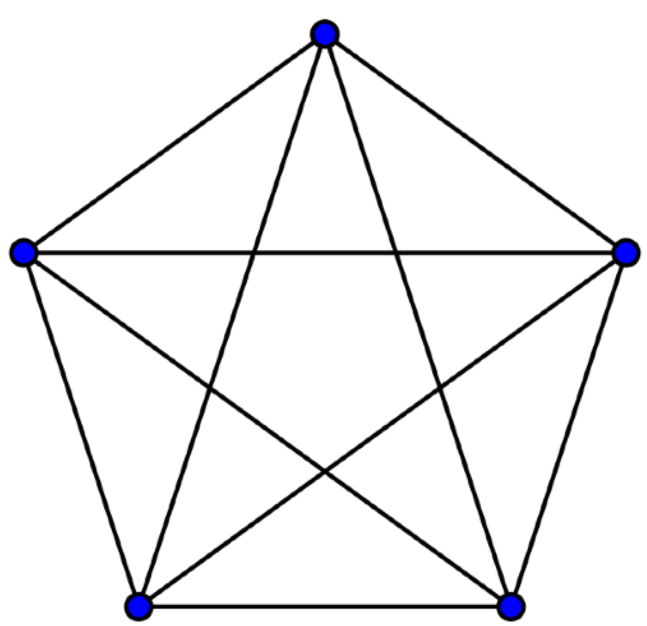
SAT
Instance: A boolean expression.
Question: Is there a truth assignment to the variables which makes the expression true?

What is an algorithm?

A **decision problem** gives a yes-no question for an input value.

GRAPH 3-COLOR
Instance: A graph G.
Question: Can G be 3-colored?

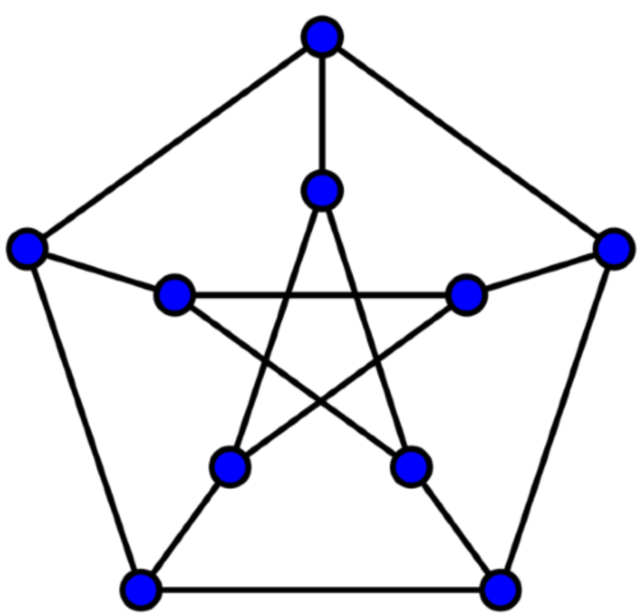




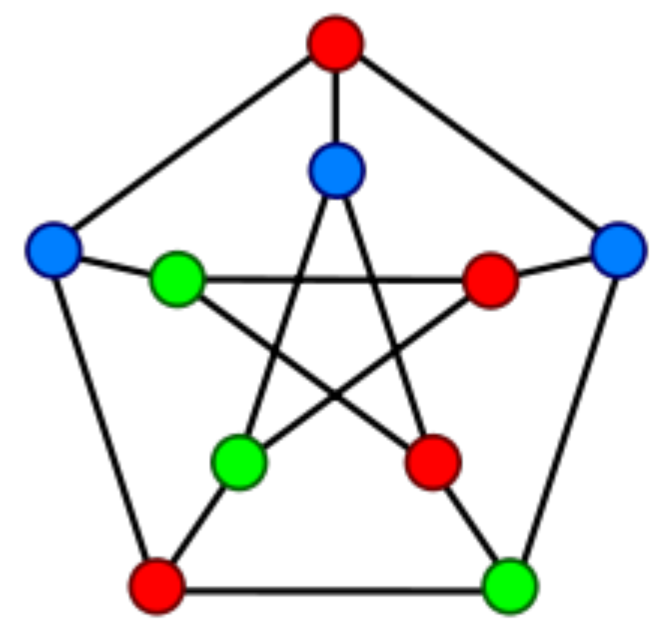
What is an algorithm?

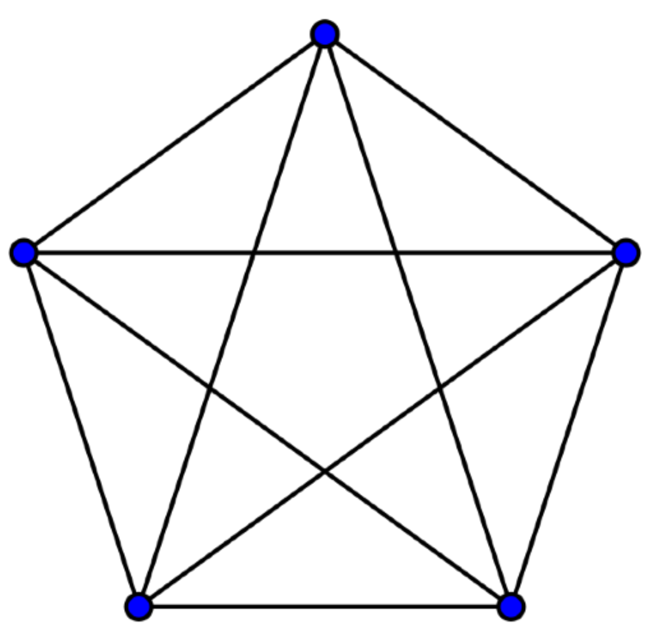
A **decision problem** gives a yes-no question for an input value

GRAPH 3-COLOR
Instance: A graph *G*.
Question: Can *G* be 3-colored?

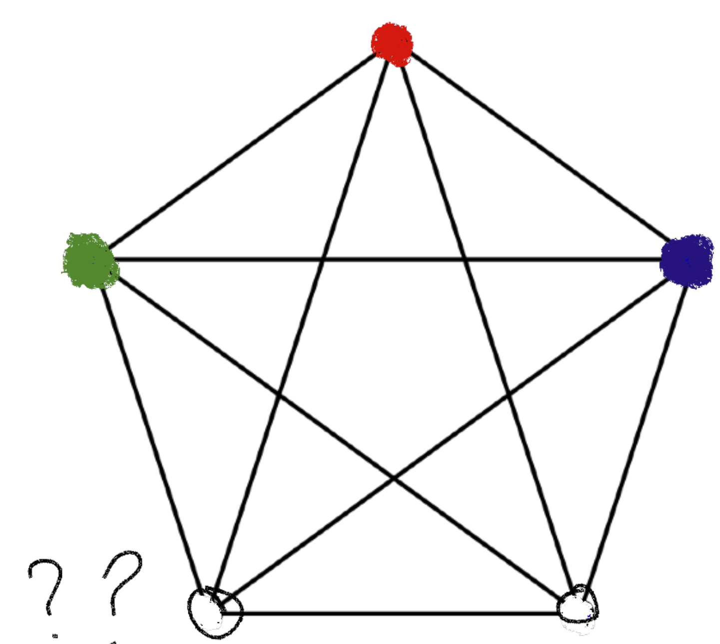


YES





NO



What is an algorithm?

A decision problem formulates a yes-no question for an input value

PRIME
Instance: An integer K
Question: Is K prime?

1003047282 0909643626 9999457312 5637852723 4362147276 6640900005 5271790903 5637310037

1003047282 0909643626 9999457312 5637852723 4362147276 6640900005 5271790903 5637310039

What is an algorithm?

A decision problem formulates a yes-no question for an input value

PRIME
Instance: An integer *K*
Question: Is *K* prime?

1003047282 0909643626 9999457312 5637852723 4362147276 6640900005 5271790903 5637310037

YES

1003047282 0909643626 9999457312 5637852723 4362147276 6640900005 5271790903 5637310039

NO

What is an algorithm?

A **decision problem** gives a yes-no question for an input value

SATISFIABILITY (SAT)
Instance: A Boolean Expression.
Question: Is there a truth assignment that satisfies the expression?

$$(\neg x_1 \vee x_2 \vee \neg y_3) \wedge (\neg x_1 \vee \neg x_2 \vee y_3) \wedge (x_1 \vee x_2 \vee y_3) \wedge (x_1 \vee \neg x_2 \vee \neg y_3)$$

What is an algorithm?

A **decision problem** gives a yes-no question for an input value

SATISFIABILITY (SAT)
Instance: A Boolean Expression.
Question: Is there a truth assignment that satisfies the expression?

$$(\neg x_1 \vee x_2 \vee \neg y_3) \wedge (\neg x_1 \vee \neg x_2 \vee y_3) \wedge (x_1 \vee x_2 \vee y_3) \wedge (x_1 \vee \neg x_2 \vee \neg y_3)$$

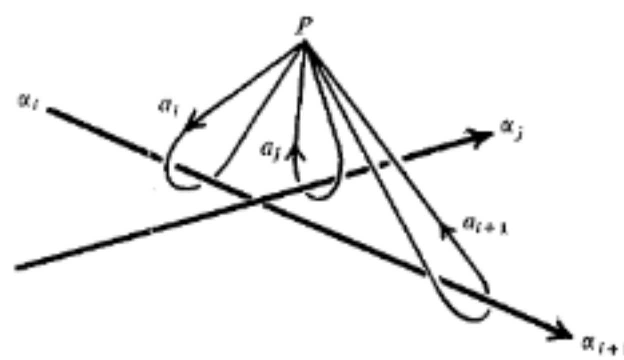
Set $x_1 = \text{False}, \quad x_2 = \text{True}, \quad y_3 = \text{False}.$

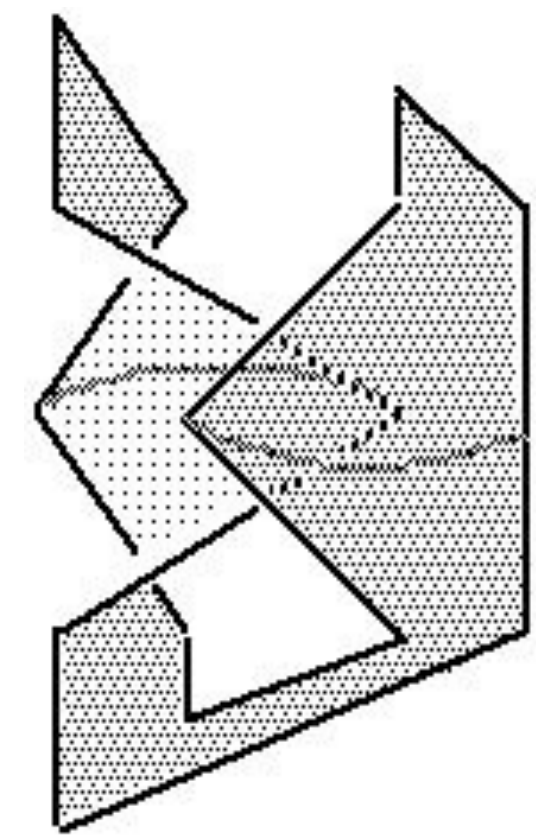
Then each clause evaluates to True, as does the whole boolean expression.

For example, the second clause $(\neg x_1 \vee \neg x_2 \vee y_3)$ is True because $\neg x_1$ is True.

Dehn’s Idea: Transform UNKNOTTING into an algorithmic problem in algebra.

Look at the knot group - the fundamental group of the knot complement. This is easy to describe with generators and relations.

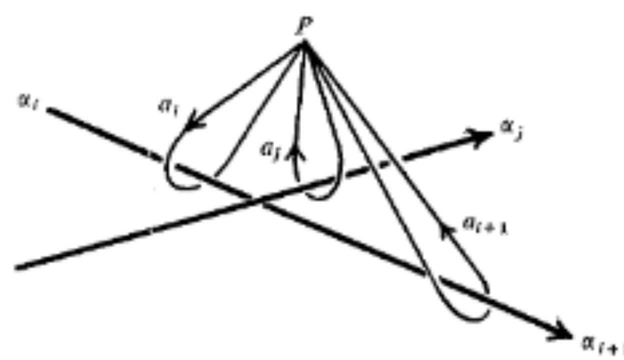


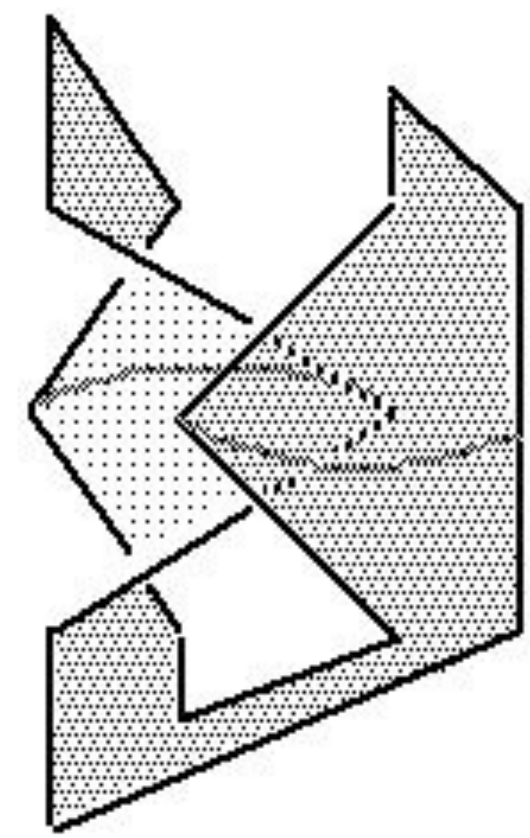


Dehn's Lemma (Proved by Papakyriakopoulos 1957).
A knot is trivial if and only if its group is infinite cyclic.

Dehn’s Idea: Transform UNKNOTTING into an algorithmic problem in algebra

Look at the knot group - the fundamental group of the knot complement. This is easy to describe with generators and relations.



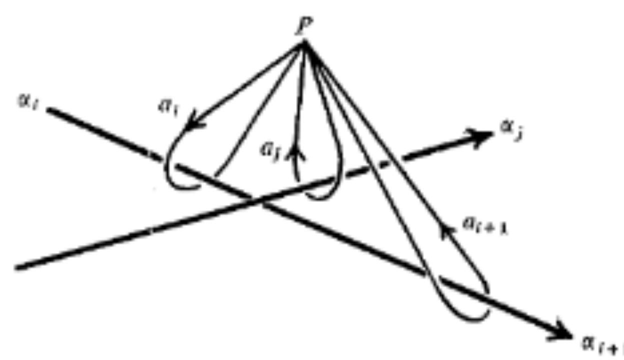


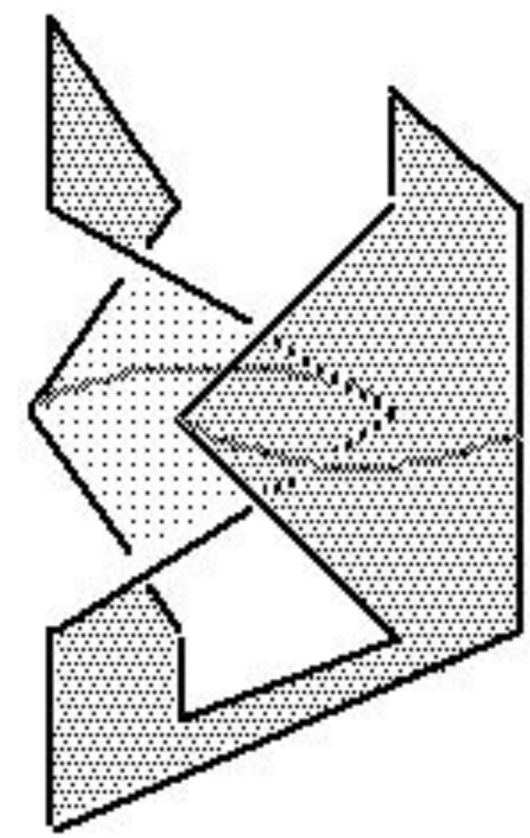
Dehn's Lemma (Proved by Papakyriakopoulos 1957).
A knot is trivial if and only if its group is infinite cyclic.

Unknotting reduces to**:**
Question: Is the knot group isomorphic to the infinite cyclic group?

Dehn’s Idea: Transform UNKNOTTING into an algorithmic problem in algebra

Look at the knot group - the fundamental group of the knot complement. This is easy to describe with generators and relations.





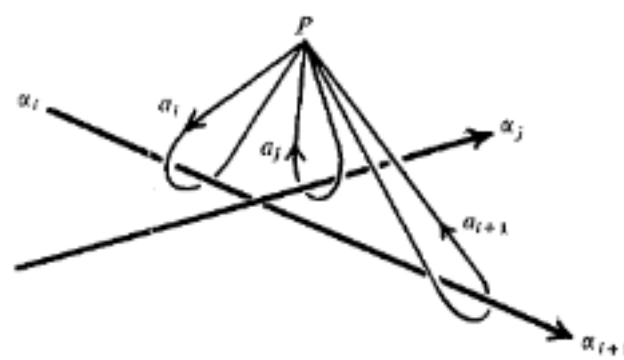
Dehn's Lemma (Proved by Papakyriakopoulos 1957).
A knot is trivial if and only if its group is infinite cyclic.

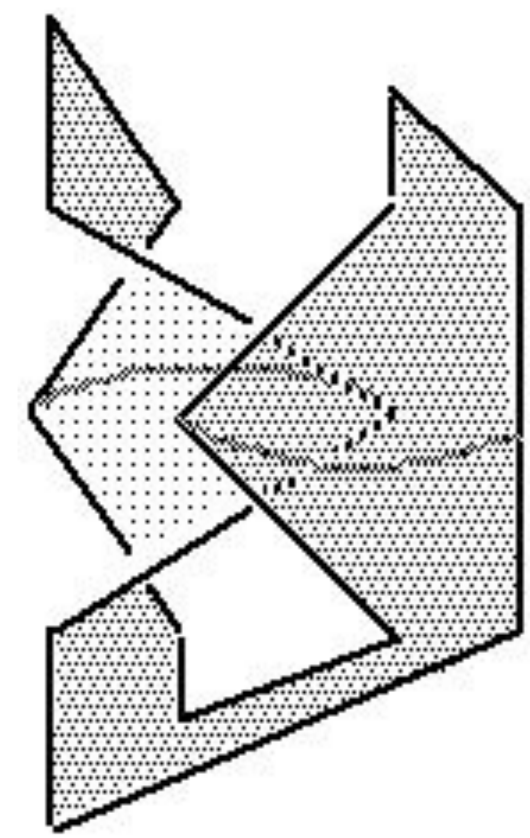
Unknotting reduces to**:**
Question: Is the knot group isomorphic to the infinite cyclic group?

Can we determine if a finitely presented group is isomorphic to the infinite cyclic group?

Dehn’s Idea: Transform UNKNOTTING into an algorithmic problem in algebra

Look at the knot group - the fundamental group of the knot complement. This is easy to describe with generators and relations.





Dehn's Lemma (Proved by Papakyriakopoulos 1957).
A knot is trivial if and only if its group is infinite cyclic.

Unknotting reduces to**:**
Question: Is the knot group isomorphic to the infinite cyclic group?

Can we determine if a finitely presented group is isomorphic to the infinite cyclic group?

No (Not in general - though yes for certain classes of groups)

Dehn formulated some basic decision problems for groups

The ***WORD PROBLEM***
The ***ISOMORPHISM PROBLEM***
The ***TRIVIALITY PROBLEM***
The ***CONJUGACY PROBLEM***

Dehn solved some of these for special groups
(free groups, surface groups).

Such decision problems for finitely presented groups are undecidable.
No algorithm exists! (1950's Novikov, Boone)

These were among the first undecidable problems found in mathematics.

Basic decision problems for groups

WORD PROBLEM
Instance: A finitely presented group $G = \langle g_1, g_2, \ldots, g_m ; r_1, r_2, \ldots, r_n \rangle$ and a word w in G.
Question: Does w represent the trivial word in G?

TRIVIALITY PROBLEM
Instance: A finitely presented group $G = \langle g_1, g_2, \ldots, g_m ; r_1, r_2, \ldots, r_n \rangle$.
Question: Is G isomorphic to the trivial group?

ISOMORPHISM PROBLEM
Instance: Two finitely presented groups G and H.
Question: Is G isomorphic to H?

Each of these problems is undecidable!
(Cantor, Hilbert, Godel, Turing 1950, Markov 1951, Novikov 1955, Adian 1955, Boone 1958, Rabin 1958)

No algorithm exists that will solve them for general groups.

Two Consequences of Undecidability of TRIVIALITY

TRIVIALITY
Instance: A finitely presented group $G = \langle g_1, g_2, \ldots, g_m ; r_1, r_2, \ldots, r_n \rangle$.
Question: Is G isomorphic to the trivial group?

Theorem (Markov 1958)
***n*-MANIFOLD RECOGNITION** is undecidable for $n \geq 4$.

There is no algorithm to decide if two closed 4-dimensional manifolds are homeomorphic.

Theorem (Novikov ~1959)
***n*-SPHERE RECOGNITION** is undecidable for $n \geq 5$.

There is no algorithm to decide if a closed 5-dimensional manifold is S^5.

4-Manifold Recognition is Undecidable

Theorem (Markov 1958)
***n*-MANIFOLD RECOGNITION** is undecidable for $n \geq 4$.

Proof: We show that an algorithm to recognize 4-manifolds implies an algorithm to solve the problem of whether a given group presentation represents the trivial group, TRIVIALITY. The latter problem is among those known to be undecidable.

We *reduce*
TRIVIALITY
to
4-MANIFOLD RECOGNITION.

1. Start with a presentation $G = \langle g_1, g_2, \ldots, g_m ; r_1, r_2, \ldots , r_n \rangle$

2. Construct a 4-dimensional manifold M with the property that
M is diffeomorphic to $\#_n S^2 \times S^2 \Leftrightarrow G$ is isomorphic to the trivial group.

Handles (5-dimensional Morse Theory)

Morse Theory shows us how to build manifolds using 1-handles and 2-handles whose fundamental group is isomorphic to a given group presentation.

1-handle: $D^1 \times D^4$, attached along $\partial D^1 \times D^4$, or $S^0 \times D^4$

2-handle: $D^2 \times D^3$, attached along $\partial D^2 \times D^3$, or $S^1 \times D^3$

Handles (5-dimensional Morse Theory)

Morse Theory shows us how to build manifolds using 1-handles and 2-handles whose fundamental group is isomorphic to a given group presentation.

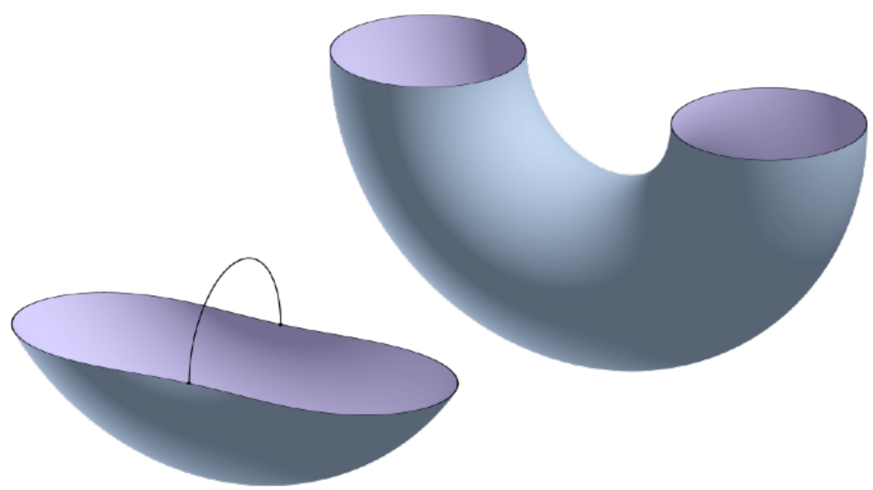
1-handle: $D^1 \times D^4$, attached along $\partial D^1 \times D^4$, or $S^0 \times D^4$

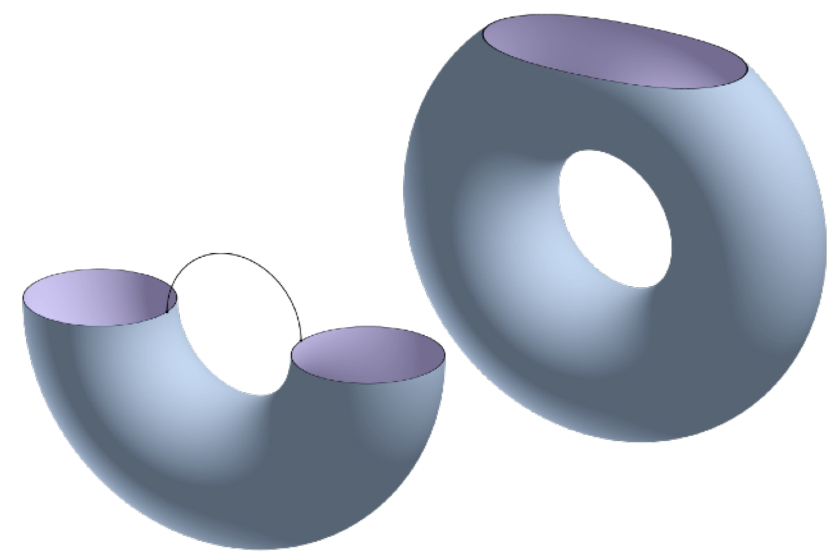
For 5D handles, take these 3D pictures and take a product of the second factor with D^2

2-handle: $D^2 \times D^3$, attached along $\partial D^2 \times D^3$, or $S^1 \times D^3$

Smooth manifolds are built with handles

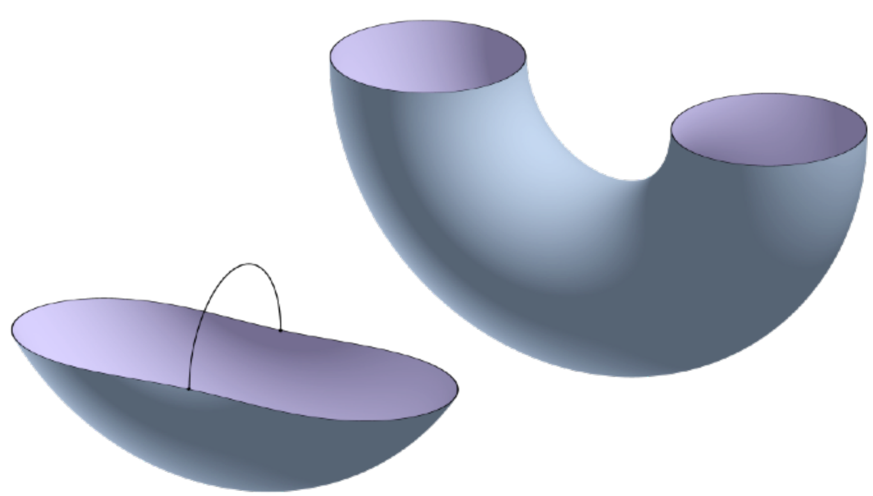
Morse Theory shows us how to build manifolds using 1-handles and 2-handles whose fundamental group is isomorphic to a given group with a finite presentation.

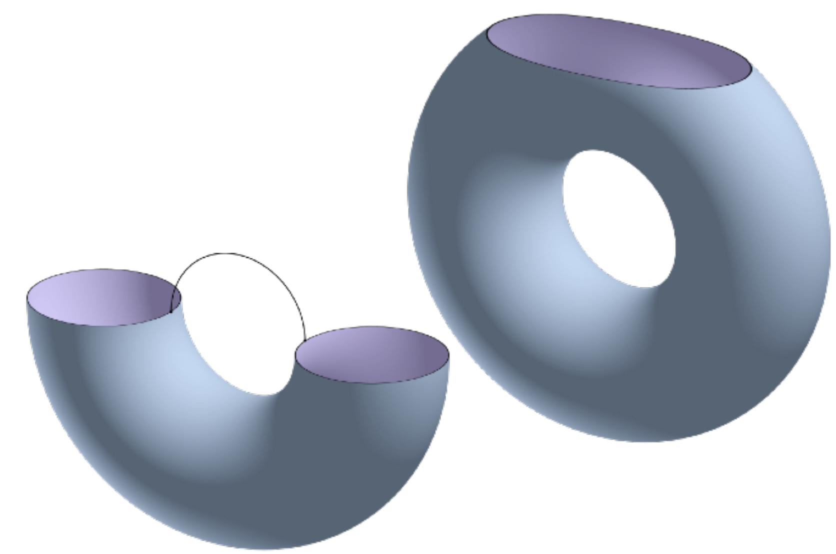




Smooth manifolds are built from handles

Morse Theory shows us how to build manifolds using 1-handles and 2-handles whose fundamental group is isomorphic to a given group presentation.

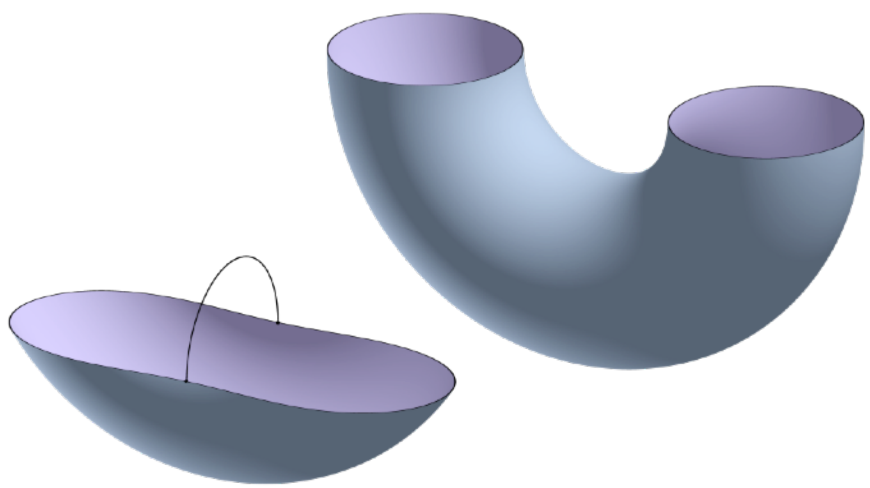


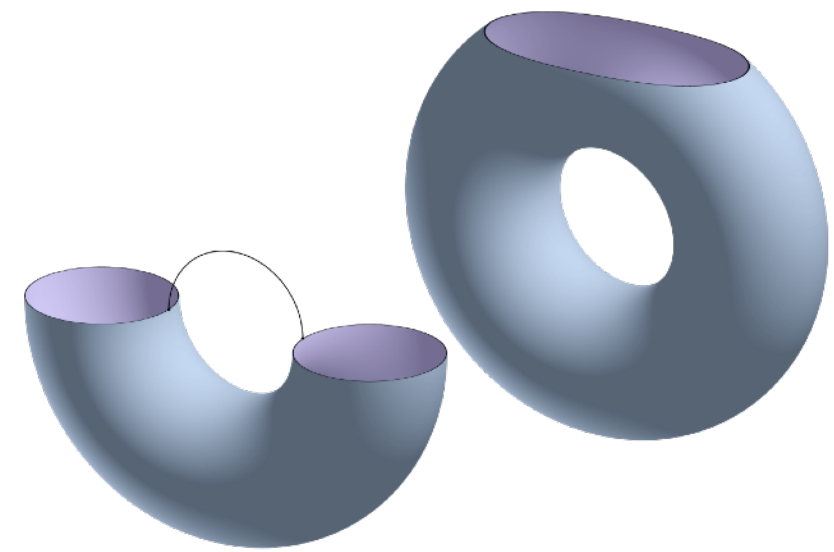


Start with a 0-handle (**B^5)** and add one 1-handle for each generator and one 2-handle for each relation.

Smooth manifolds are built from handles

Morse Theory shows us how to build manifolds using 1-handles and 2-handles whose fundamental group is isomorphic to a given group presentation.





$$G = \langle g_1, g_2, \ldots, g_m ; r_1, r_2, \ldots, r_n \rangle$$

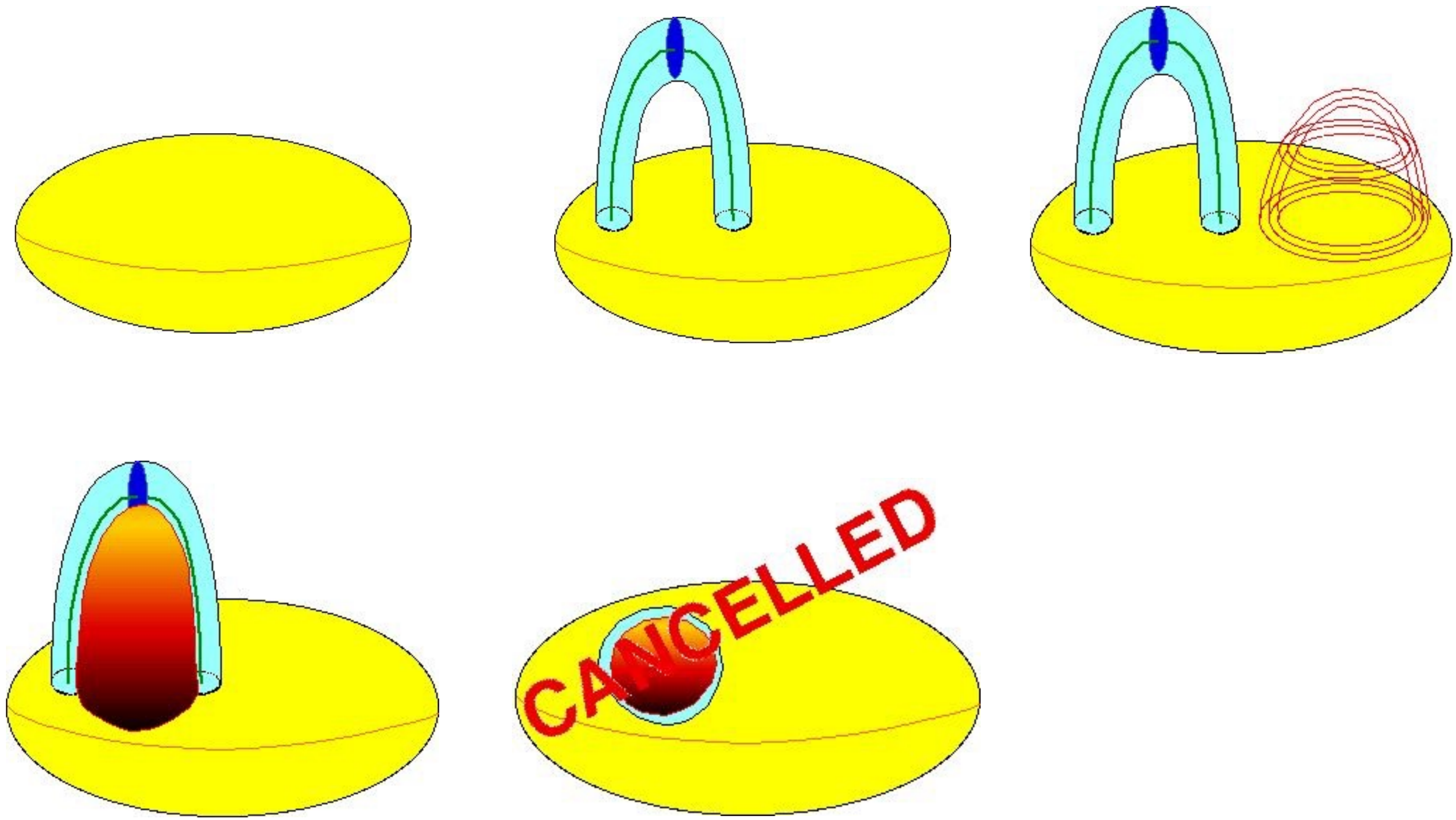
Start with a 0-handle (**B⁵**) and add one 1-handle for each generator and one 2-handle for each relation.

$\pi_1 \simeq \mathbb{Z} \simeq \langle g_1 ; \rangle$

$\pi_1 \simeq \mathbb{Z} \oplus \mathbb{Z} \simeq \langle g_1, g_2 ; \rangle$

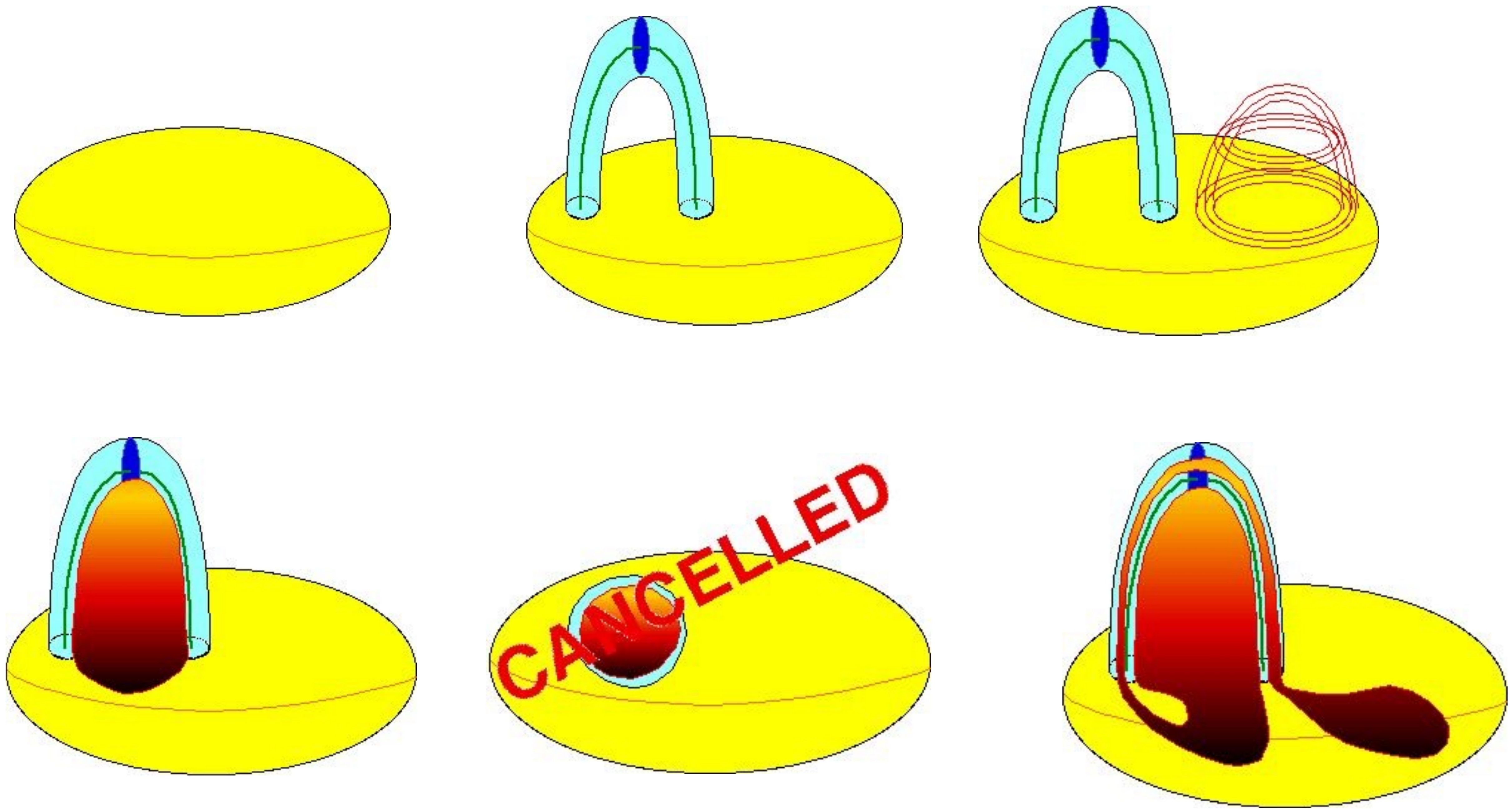
$\pi_1 \simeq \mathbb{Z} \simeq \langle g_1, g_2 ; g_1^2 g_2 \rangle$

Canceling Handles



A 1-handle and a 2-handle cancel each other if the attaching curve of the 2-handle runs once over the 1-handle.

Canceling Handles



A 1-handle and a 2-handle cancel each other if the attaching curve of the 2-handle runs once over the 1-handle.

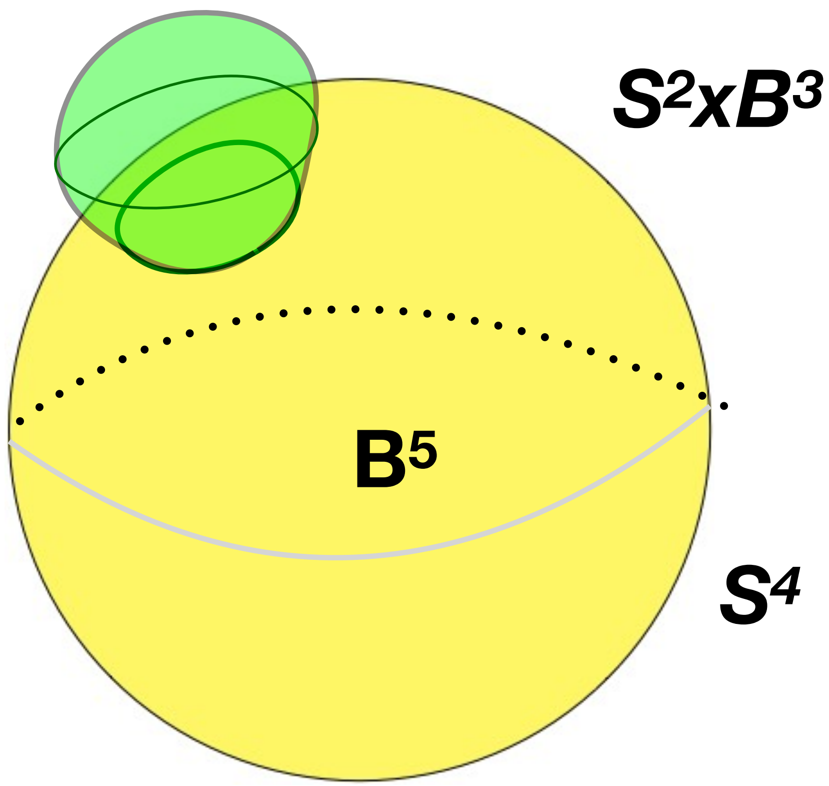
A 1-handle and a 2-handle cancel each other if the attaching curve of the 2-handle can be isotoped (deformed through embedding) to run once over the 1-handle.

In dimensions 4 and above, homotopy and isotopy of curves are the same.

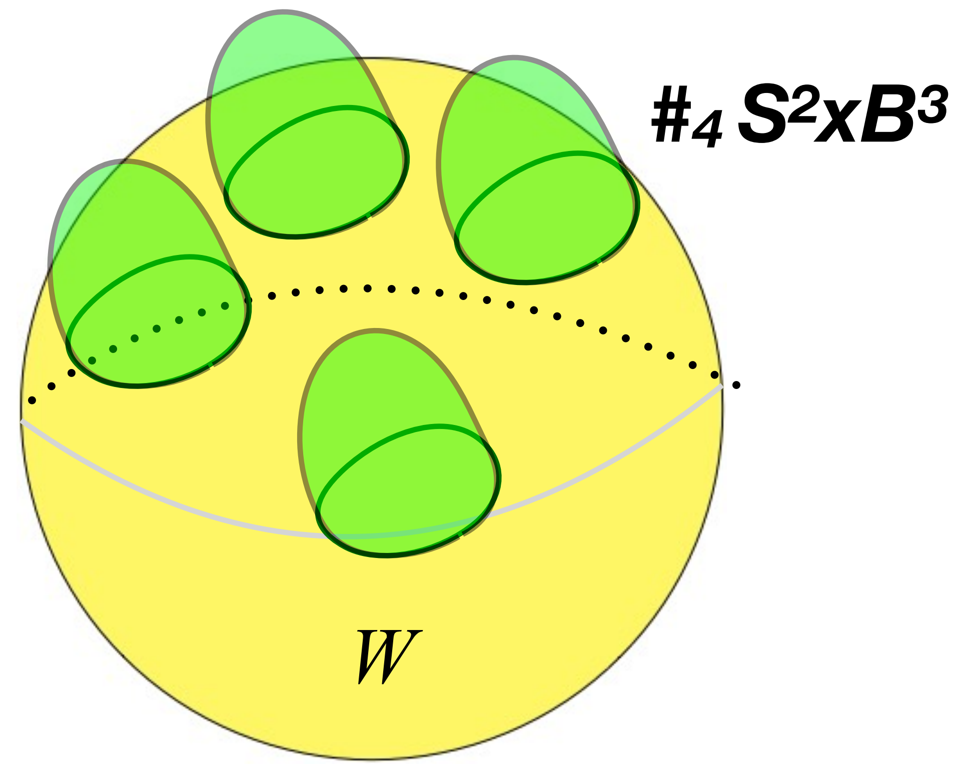
Important Example - $S^2 x S^2$

Start with a 5-dimensional ball $\mathbf{B}^5$. Its boundary is a 4-sphere $\mathbf{S}^4$.

Attaching n 2-handles to $\mathbf{B}^5$ gives a manifold W diffeomorphic to $\#_n\, S^2xB^3$.



one 2-handle



Four 2-handles

Attaching n trivial 2-handles to $\mathbf{B}^5$ gives a manifold *whose boundary* is diffeomorphic to $\#_n\, S^2xS^2$.

There is only one way to attach a 2-handle to $\mathbf{B}^5$, since all curves in S^4 are isotopic.

4-Manifold Recognition is Undecidable

Proof: We show that an algorithm to recognize 4-manifolds implies that there is an algorithm to solve the problem of whether a given presentation represents the trivial group, TRIVIALITY. But the latter is known to be undecidable.

We *reduce* TRIVIALITY to 4-MANIFOLD RECOGNITION.

1. Start with a presentation $G = \langle g_1, g_2, \ldots, g_m ; r_1, r_2, \ldots, r_n \rangle$

2. Construct a 5-dimensional manifold W^5 with $\pi_1(W) \approx G$ as follows:

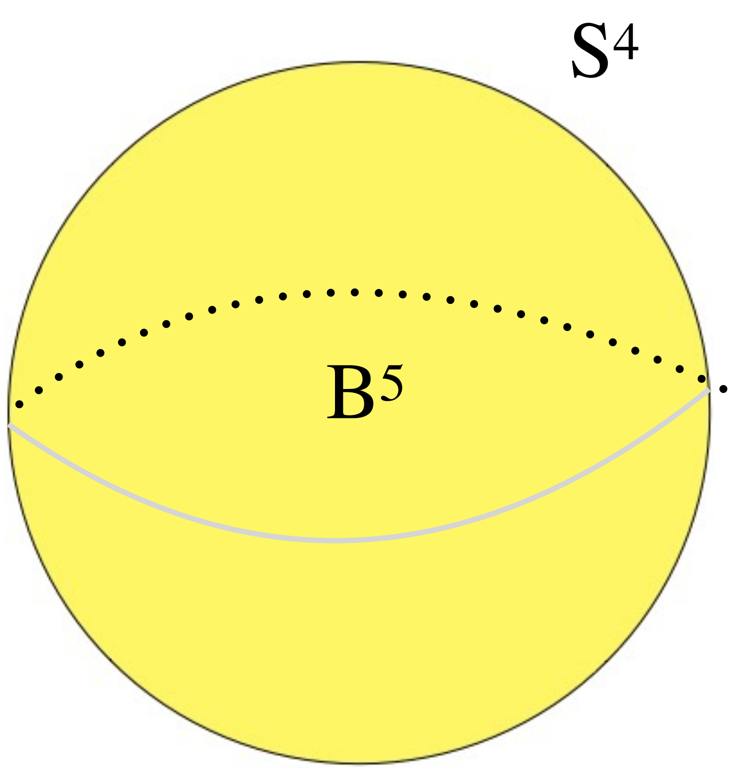
4-Manifold Recognition is Undecidable

Proof: Reduce TRIVIALITY to 4-MANIFOLD RECOGNITION.

1. Start with a presentation $G = \langle g_1, g_2, \ldots, g_m ; r_1, r_2, \ldots, r_n \rangle$

2. Construct a 5-dimensional manifold W^5 as follows:

a. Take the 5 ball B^5. Its boundary is S^4.

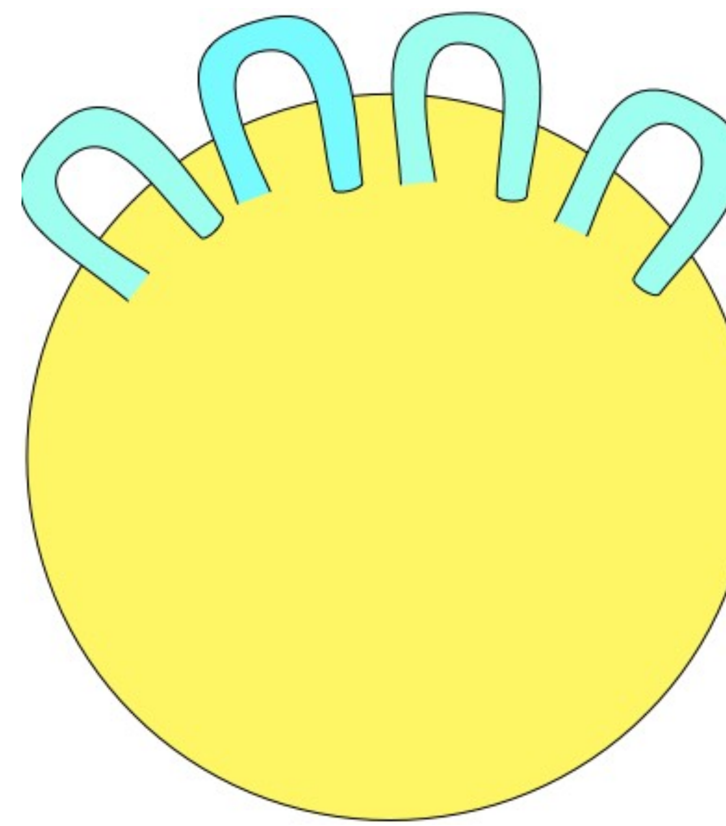


4-Manifold Recognition is Undecidable

Proof: Reduce TRIVIALITY to 4-MANIFOLD RECOGNITION.

1. Start with a presentation $G = <g_1, g_2, \ldots, g_m ; r_1, r_2, \ldots, r_n>$

2. Construct a 5-dimensional manifold W^5 as follows:



a. Take the 5 ball B^5. Its boundary is S^4.

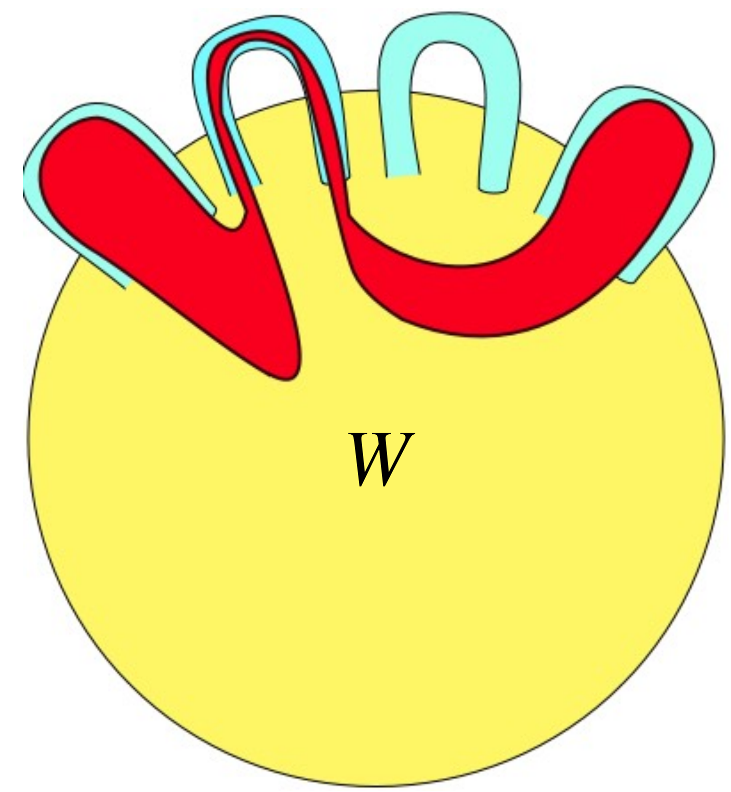
b. Add m 1-handles.
Boundary is now $\#_m S^2 x S^2$

4-Manifold Recognition is Undecidable

Proof: Reduce TRIVIALITY to 4-MANIFOLD RECOGNITION.

1. Start with a presentation $G = \langle g_1, g_2, \ldots, g_m ; r_1, r_2, \ldots, r_n \rangle$

2. Construct a 5-dimensional manifold W^5 as follows:



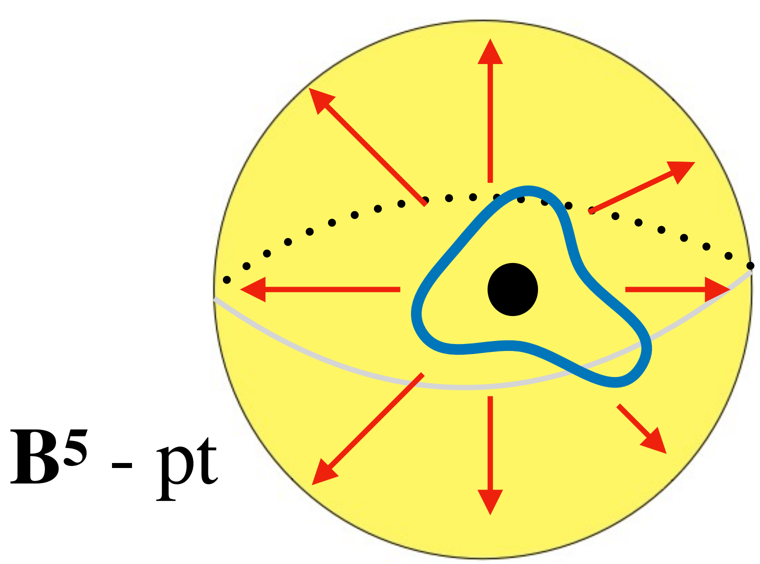
a. Take the 5 ball B^5. Its boundary is S^4.

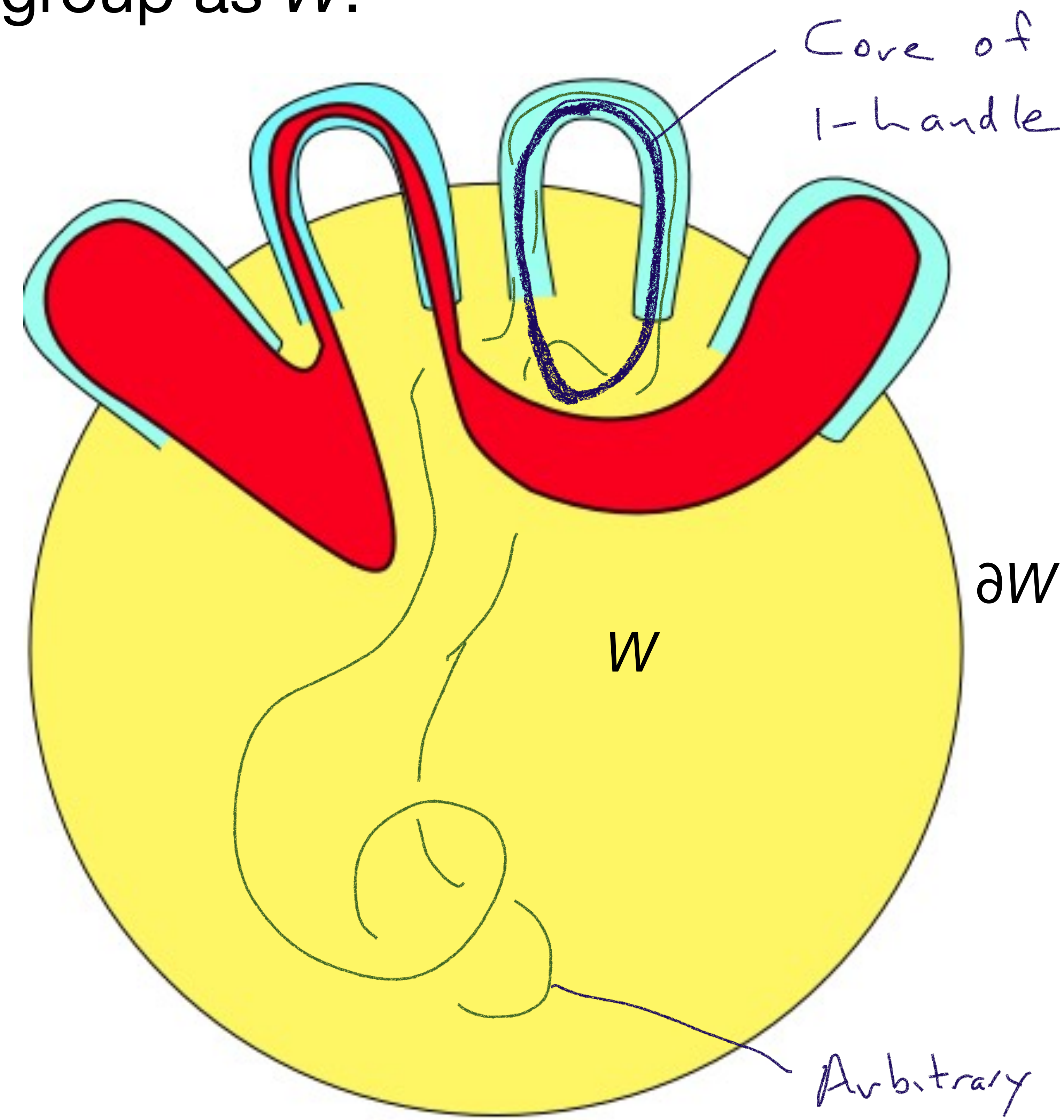
b. Add m 1-handles.
Boundary is now $\#_m S^2xS^2$

c. Add n 2-handles. Attach them so they follow the presentation given for G. Now have a 5-manifold W with fundamental group G. Its boundary is some complicated 4-manifold.

Claim. The boundary of W, ∂W, is a 4-manifold with the same fundamental group as W.

Proof. There is enough room in five dimensions to push curves and curve homotopies to ∂W. A homotopy of a curve is 2-dimensional. In 5 dimensions the homotopy can be made to miss the 1-dimensional core of a 1-handle and the 2-dimensional core of a 2-handle. Thus curves and curve homotopies can be pushed out to ∂W.





In codimension 3, curves and surfaces can be pushed to ∂W

4-Manifold Recognition is Undecidable

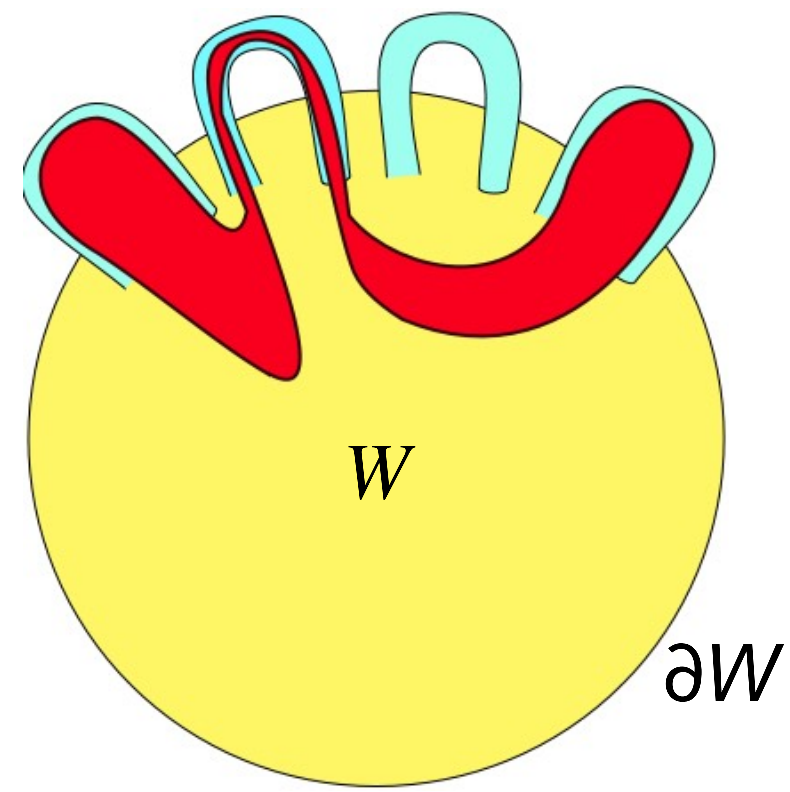
Proof: Reduce TRIVIALITY to 4-MANIFOLD RECOGNITION.

1. Start with a presentation $G = \langle g_1, g_2, \ldots, g_m ; r_1, r_2, \ldots, r_n \rangle$

2. Construct a 5-dimensional manifold W^5 as follows:

a. Start with the 5 ball B^5. Its boundary is S^4.

b. Add m 1-handles.



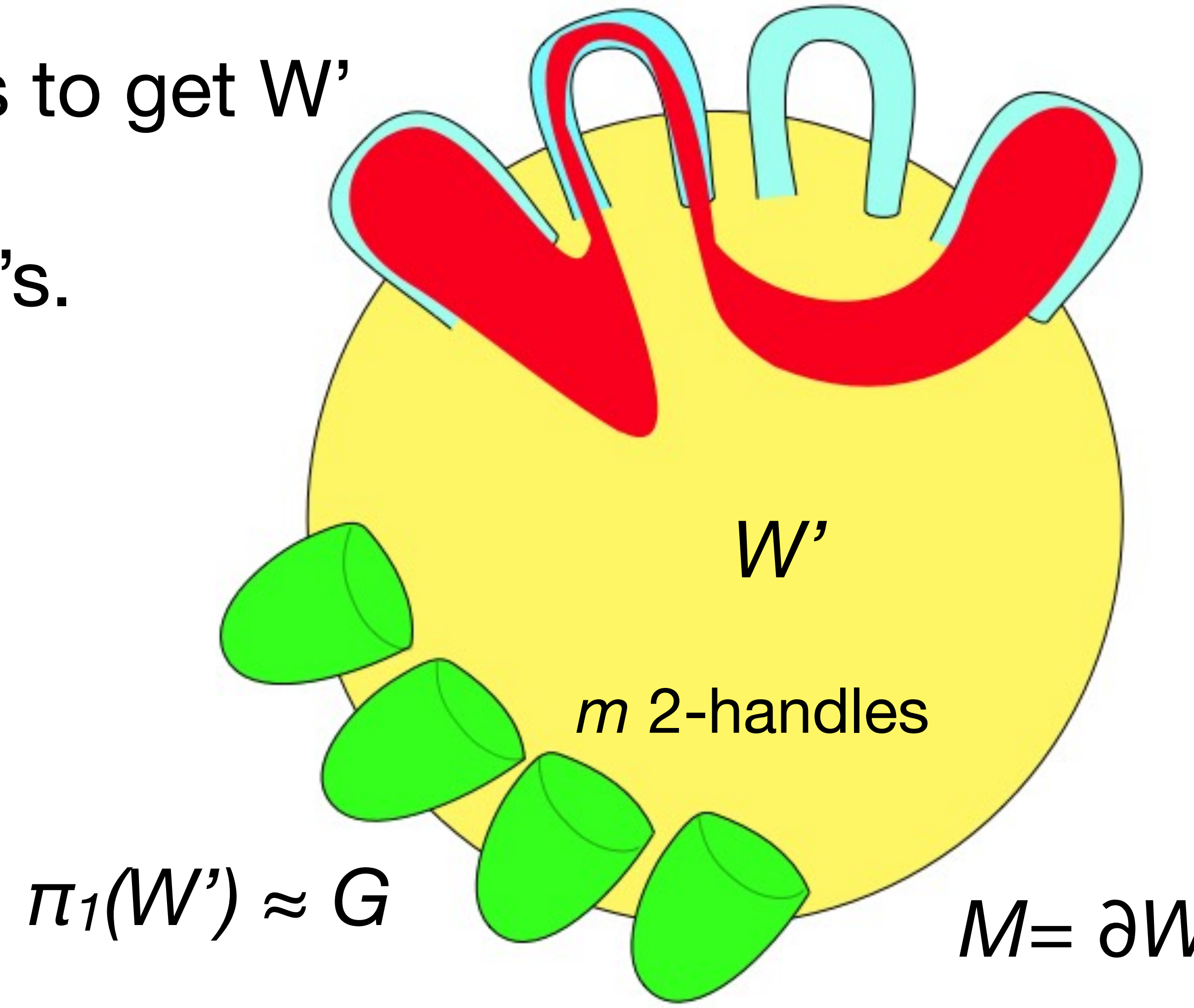
c. Add n 2-handles. Now have a 5-manifold W with fundamental group G.

d. The boundary of W is a 4-manifold with the same fundamental group as W.

e. Add an additional *m* trivial 2-handles to get W’

W’ is the connect sum of W with $S^2 \times B^3$’s.

$\pi_1(W) \approx \pi_1(W') \approx G$, does not change.



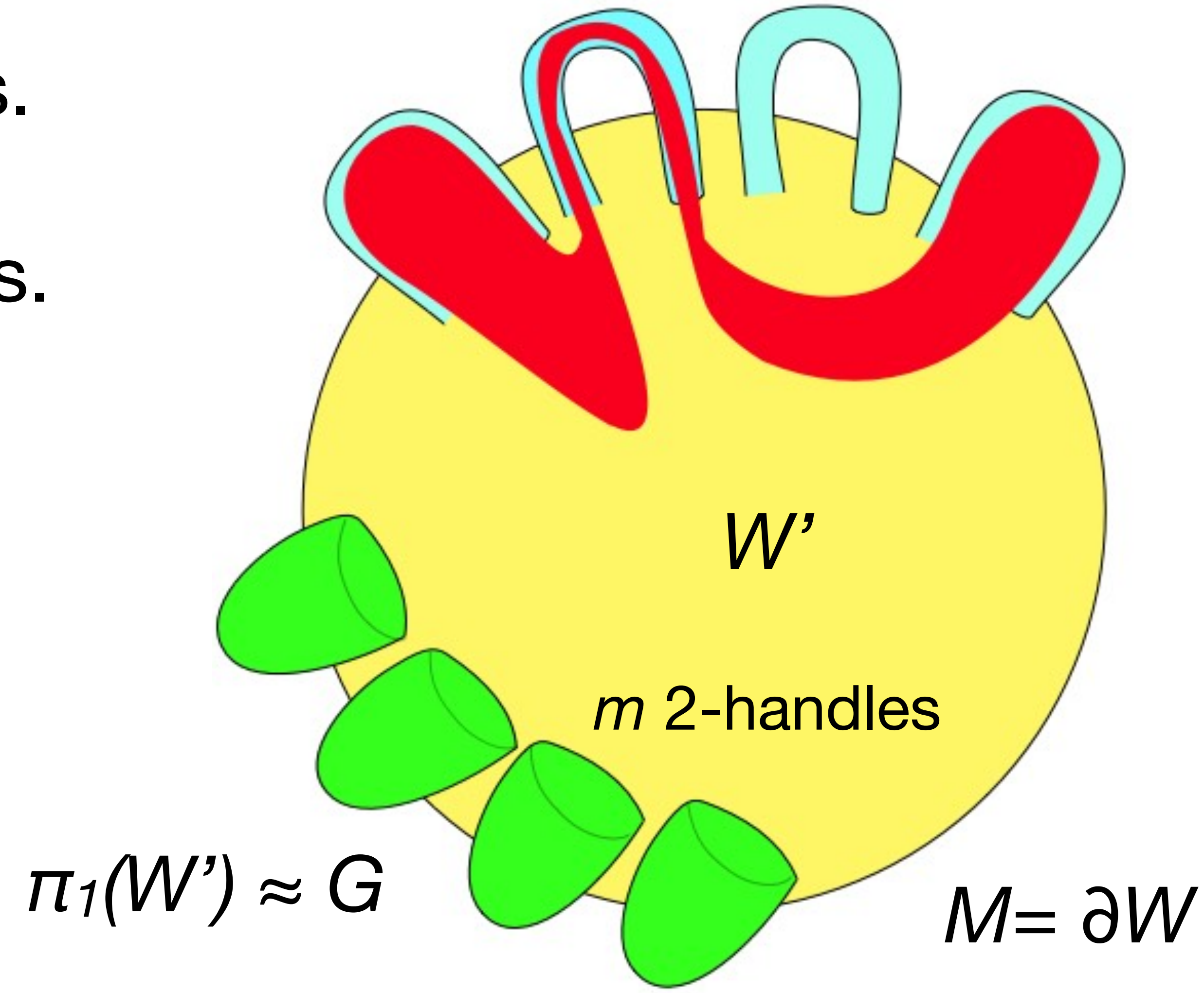
Let *M* be the boundary of this 5-manifold.
$\pi_1(M) \approx \pi_1(W') \approx G$.

Theorem (Markov, 1958)
M is diffeomorphic to $\#_n\, S^2 \times S^2 \Leftrightarrow$ *G* is the trivial group

e. Add an additional *m* trivial 2-handles.

W is the connect sum of W with $S^2 \times B^3$'s.

$\pi_1(W) \approx \pi_1(W') \approx G$, does not change.



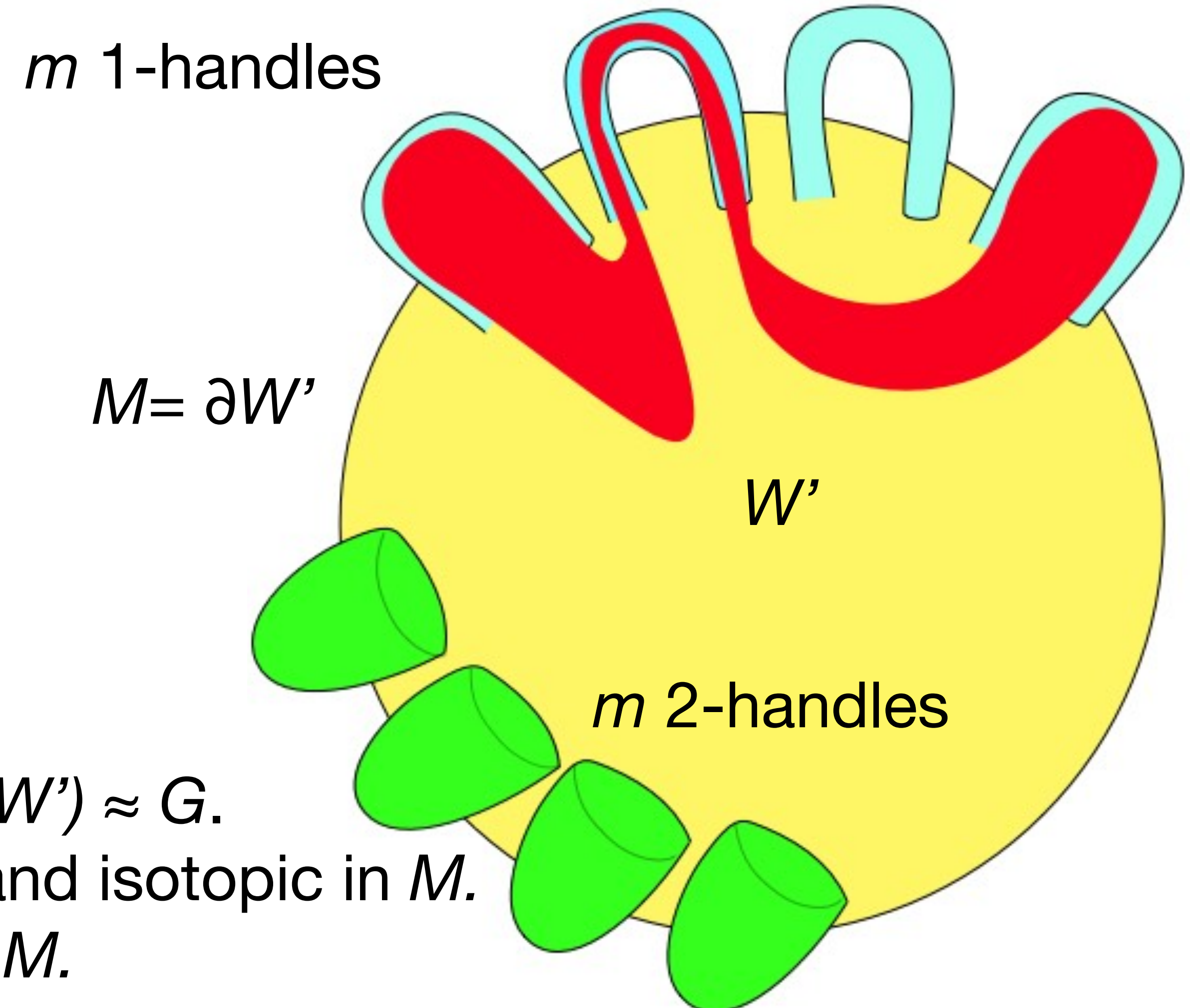
Let M be the boundary of this 5-manifold.
$\pi_1(M) \approx \pi_1(W') \approx G$.

Theorem (Markov, 1958)
M is diffeomorphic to $\#_n\, S^2 \times S^2 \;\Leftrightarrow\; G$ is the trivial group

Proof: $\Rightarrow$ Immediate, since $\pi_1(\#_n\, S^2 \times S^2) \approx 1$.

Theorem
M is diffeomorphic to $\#_n S^2xS^2$
$\Leftrightarrow$ G is the trivial group

Proof:



$\Leftarrow$

We know from the construction that $\pi_1(M) \approx \pi_1(W') \approx G$.
So $\pi_1(M) \approx 1$, and so all curves are homotopic and isotopic in M.
So all 2-handle attaching curves are isotopic in M.
So the m (green) 2-handles can be isotoped to run exactly once over, and so cancel, the m (blue) 1-handles.
Then the n (red) 2-handles are being attached to an S^4 and are isotopic to n trivial 2-handles.
Attaching n trivial 2-handles (red) to S^4 gives a manifold M that is diffeomorphic to $\#_n S^2xS^2$.

Corollary (Markov) 4-Manifold Recognition is Undecidable

Proof. Since we can't algorithmically decide if G is the trivial group, no algorithm will tell us if M is diffeomorphic to $\#_n S^2xS^2$.

Corollary (Markov) 4-Manifold Recognition is Undecidable

What Other Problems are Undecidable?

Perhaps most topology problems? Your thesis problem?

Corollary (Markov) 4-Manifold Recognition is Undecidable

What Other Problems are Undecidable?

Perhaps most topology problems? Your thesis problem?

Haken's results come right on the tail of these undecidability results and indicate that three manifold questions seem to have algorithmic solutions. This leads to a general idea that algorithm obstacles are closely related to dimension.

Dimension 1 and 2: Fast algorithms generally exist.
Dimension 3: Algorithms generally exist, but could be exponential.
Dimension 4 and above: Many problems are undecidable.

Now to Dimension Three

So far, Haken's approach seems to have the most widespread applicability to 3-manifolds problems.

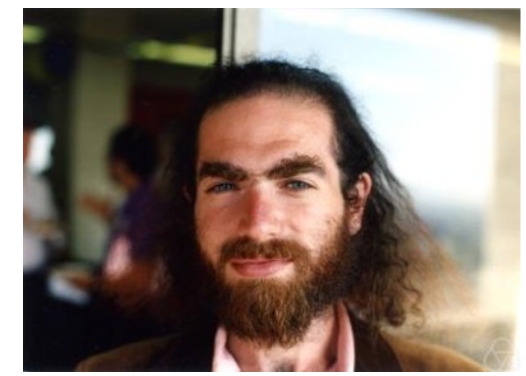
But there are many appealing approaches that seem to be plausible for UNKNOTTING.

Other Approaches to Unknotting

1. Geometric structures

Thurston (1978): Knot complements have geometric structures.

Perelman (2003) 3-manifolds have geometric structures.



Sela (1995): There is an algorithm to determine if two geometric 3-manifolds are homeomorphic.

Sela's approach reduces to determining if two hyperbolic groups are isomorphic. (More general than determining if two 3-manifold groups are isomorphic.)

The running time of such algorithms seems to involve towers of exponentials.

Approach to Unknotting 2: Knot invariants

Alexander Polynomial (1920)
There are non-trivial knots with trivial Alexander Polynomial.
Jones Polynomial (1984)
It is not known if $V(K) = 1$ only when K is unknotted.
There are distinct knots with the same Jones polynomial.

Approach to Unknotting 2: Knot invariants

Alexander Polynomial (1920)
There are non-trivial knots with trivial Alexander Polynomial.
Jones Polynomial (1984)
It is not known if $V(K) = 1$ only when K is unknotted.
There are distinct knots with the same Jones polynomial.

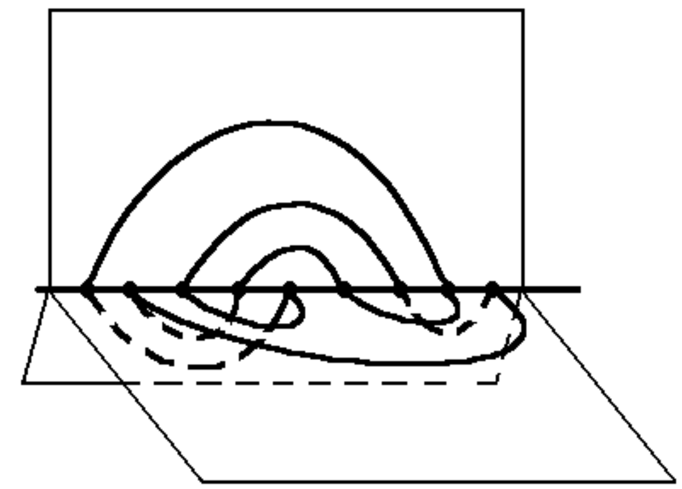
Other knot invariants can distinguish the unknot:
1. Knot Floer Homology (2004, Ozsváth and Szabó)
2. A-polynomial (2004, Boyer and Zhang, Dunfield and Garoufalidis)
3. Khovanov Homology (2011, Kronheimer and Mrowka)

But -
1. Can be hard to compute
2. Relation to 3-manifold theory is unclear.

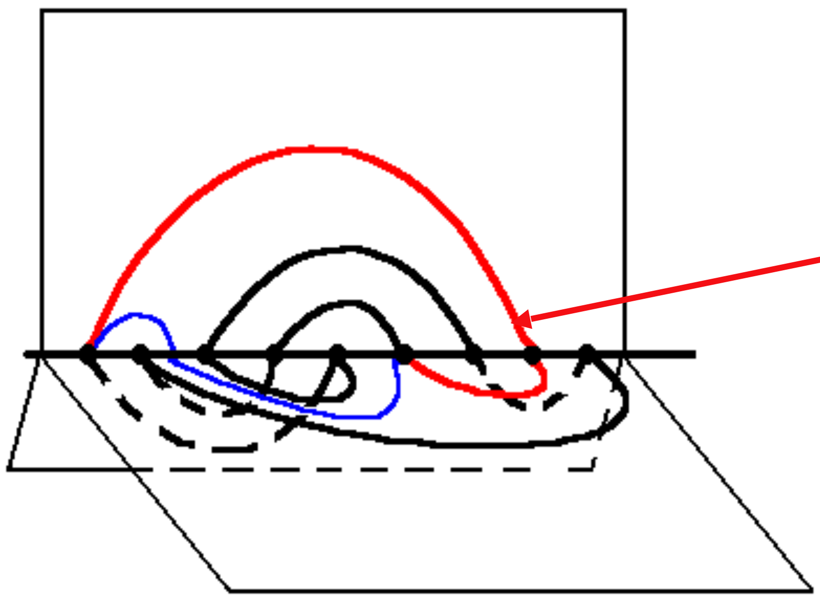
The Normal Surface approach we will look at to detect unknots appears to be
a. Easier to compute.
b. Can recognize *all* knots - gives a classification of knots.
c. Widely applicable for many 3-manifold problems.

Approach 3: 3-page books

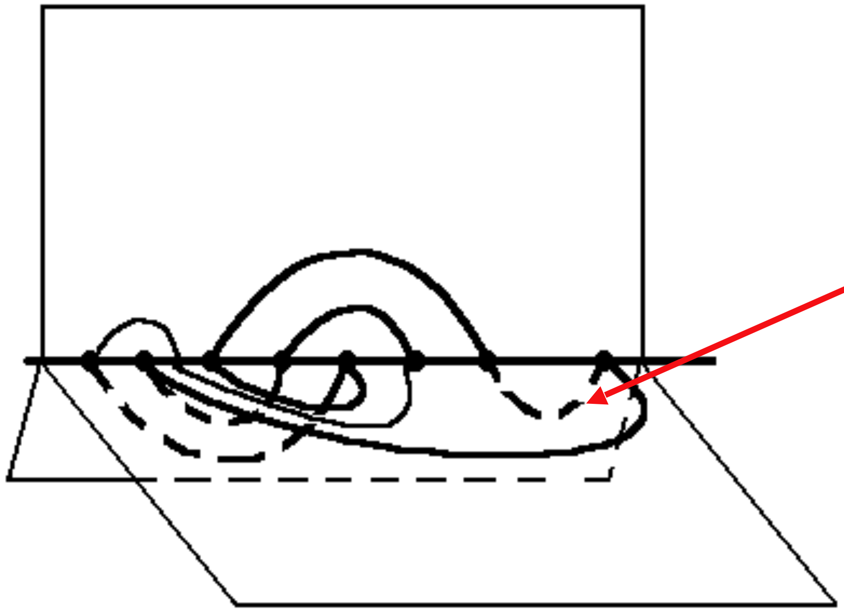
(I. Dynnikov) Knots are represented as curves on a book with three pages.



Some Moves:

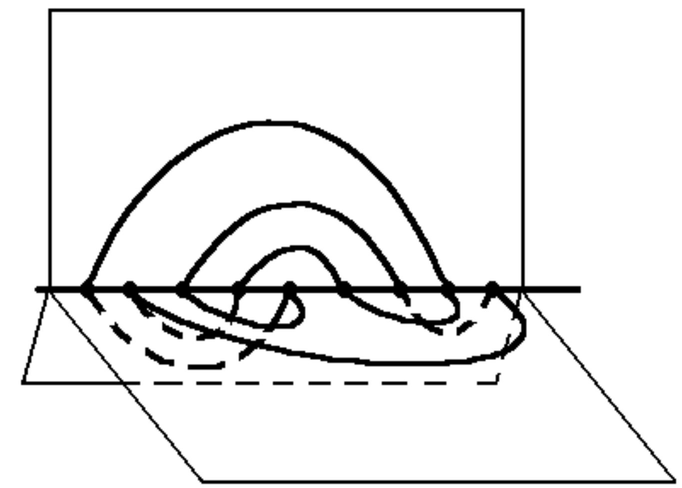


Replace the red pair of arcs with the blue pair.



Simplify:
Contract the arc connecting the rightmost pair of points.
Two less points on central line.

Approach 3: 3-page books



Knots are represented as curves on a book with three pages.

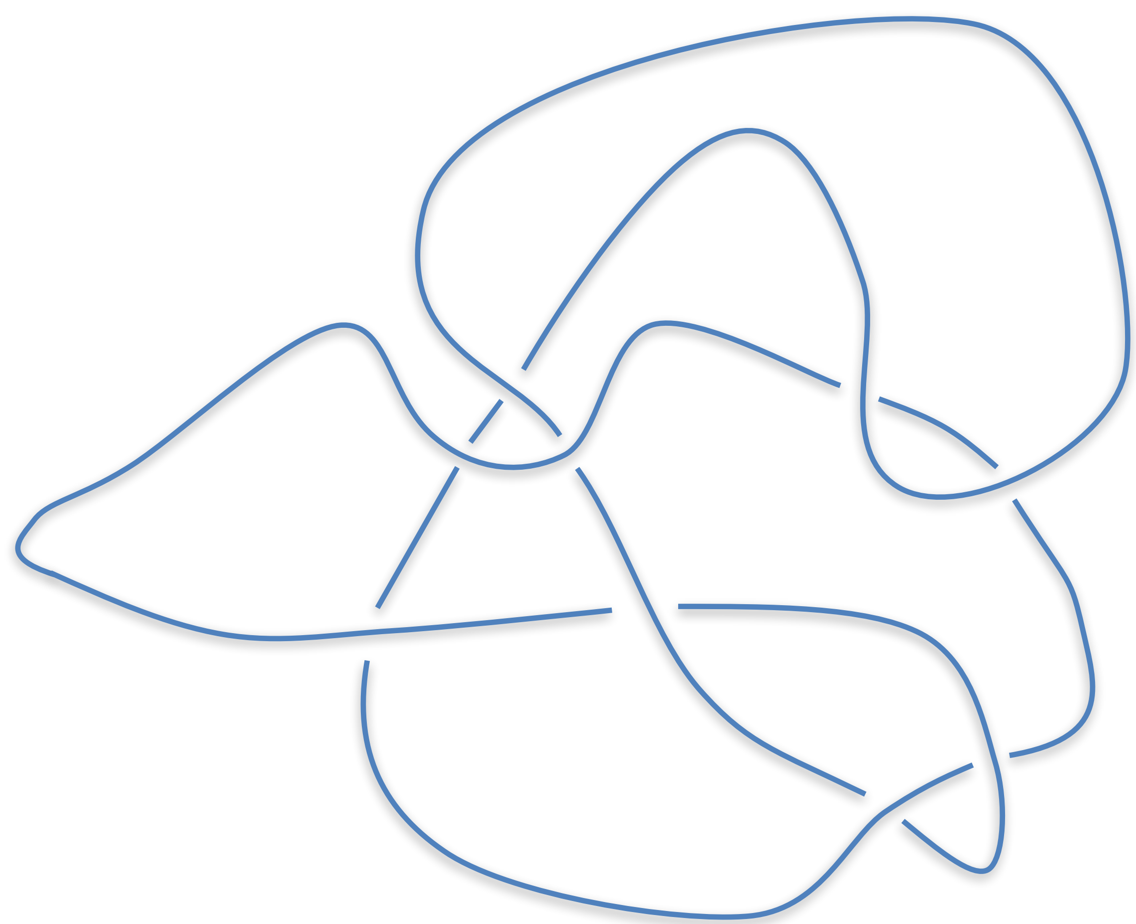
Theorem (Dynnikov 1999)
This gives an unknotting algorithm.

This is one of several approaches implemented in software.

Book Knot Simplifier
Andreeva, Dynnikov, Koval, Polthier, Taimanov

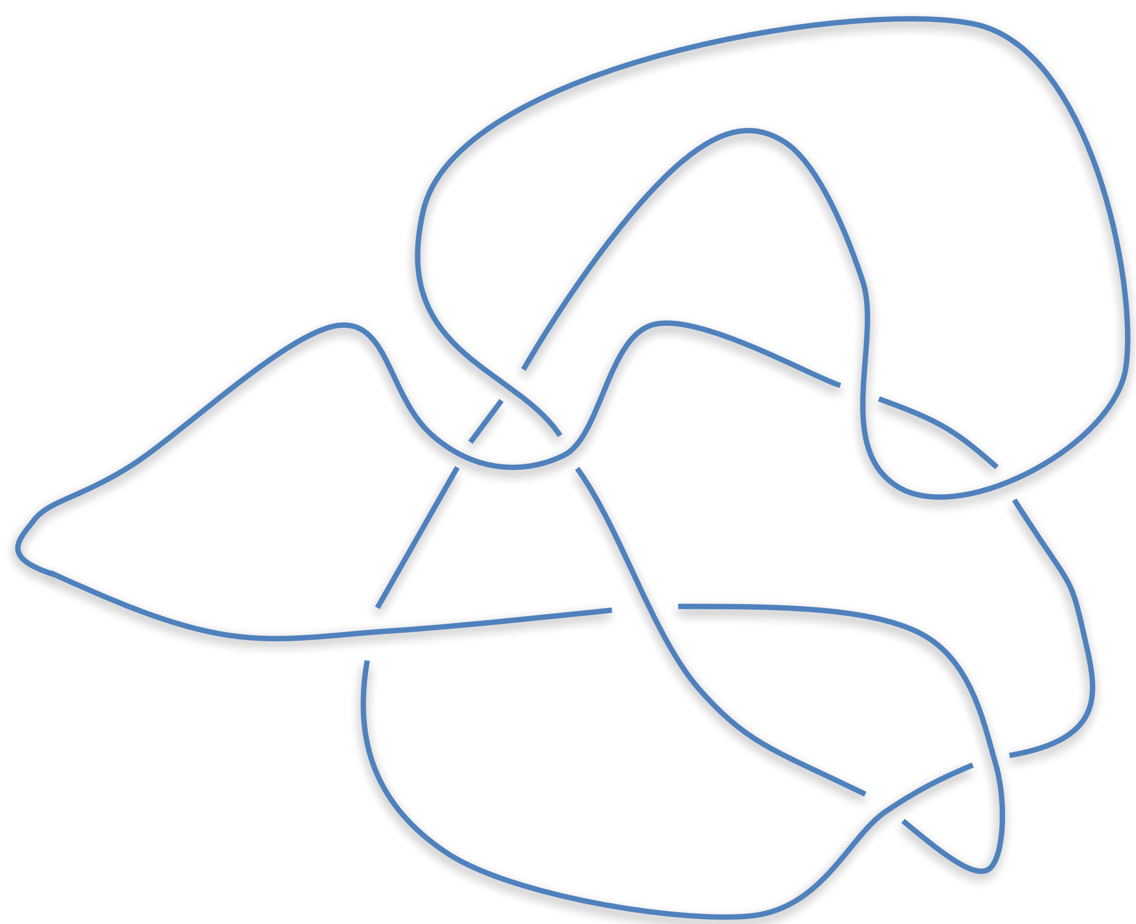
Knot Simplifier web service
http://www.javaview.de/services/knots

Approach 4: Diagrams



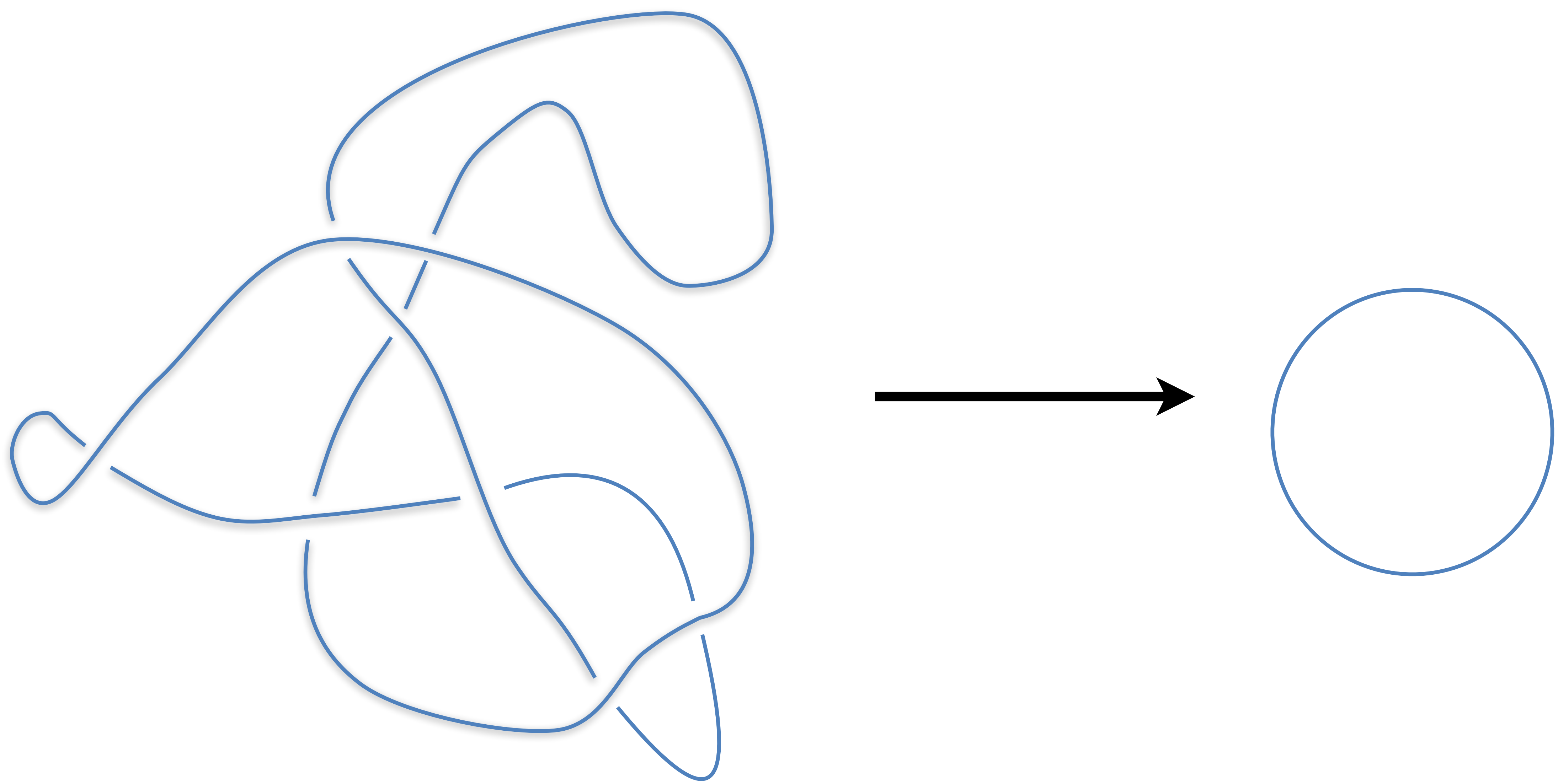
The study of knot diagrams - planar curves with choices of over and under-crossings, is an interesting subject of its own. Can we work directly with diagrams, manipulating them to simplify unnecessary crossings?

Knot and Link Diagrams



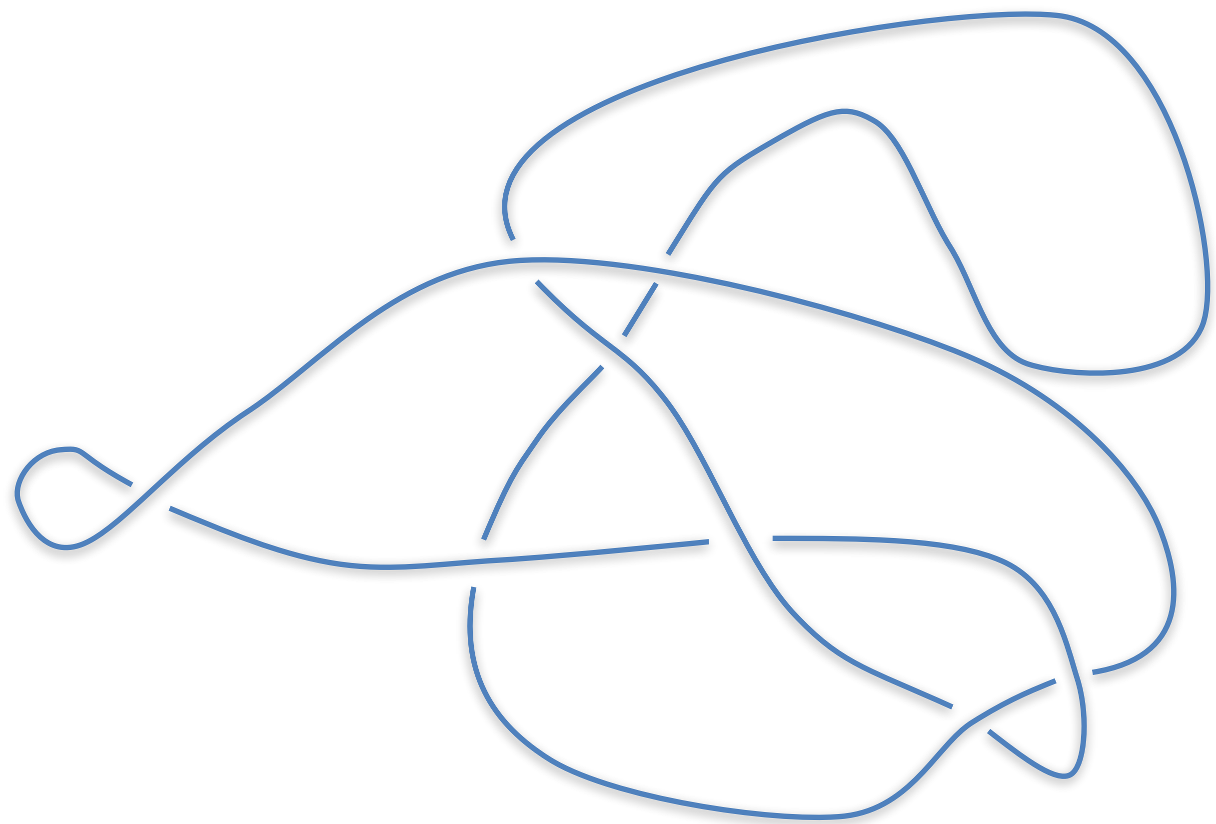
The study of knot diagrams - planar curves with choices of over and under-crossings, is an interesting subject of its own.

Direct approach to Unknotting: Simplify Knot Diagrams



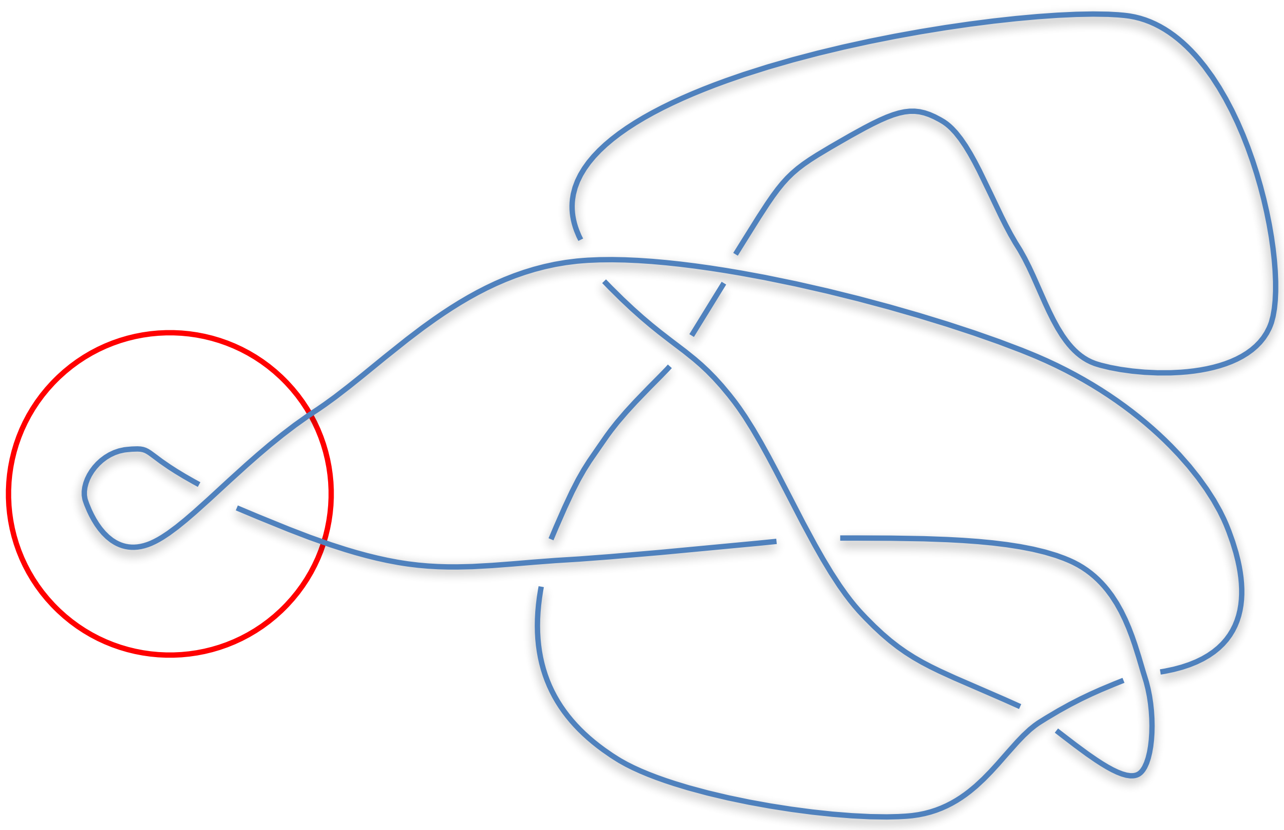
Question:
Can we find a way to change an *n*-crossing unknot diagram to a trivial diagram?

Diagrams can be changed using Reidemeister moves, without changing the knot they represent.



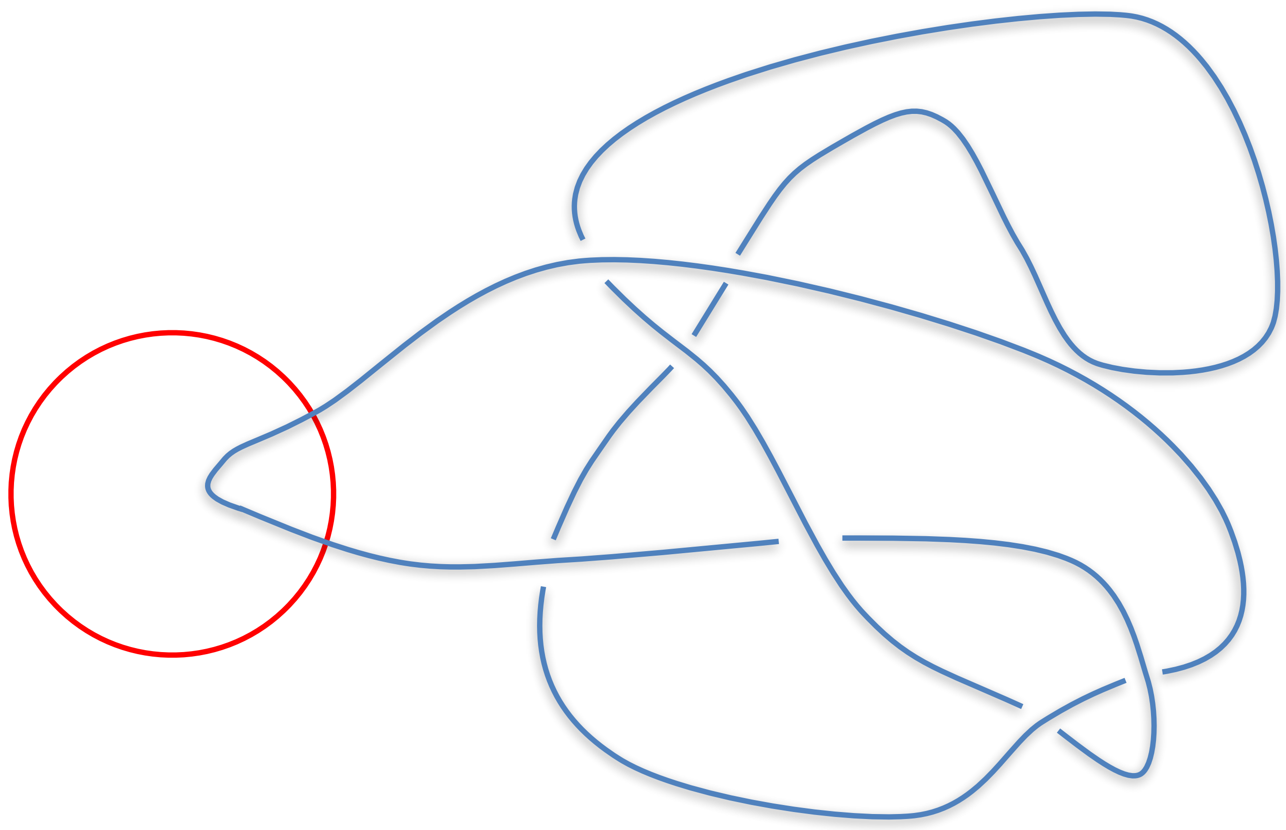
Reidemeister moves

R1

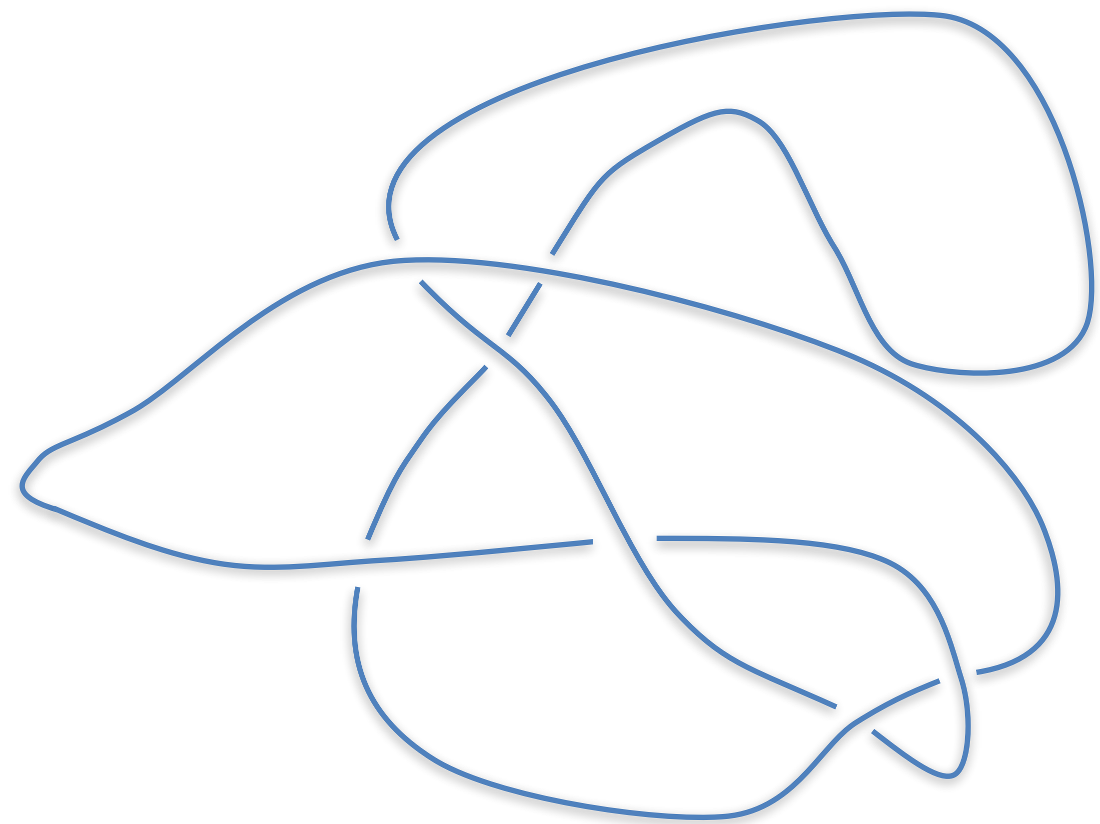


Reidemeister moves

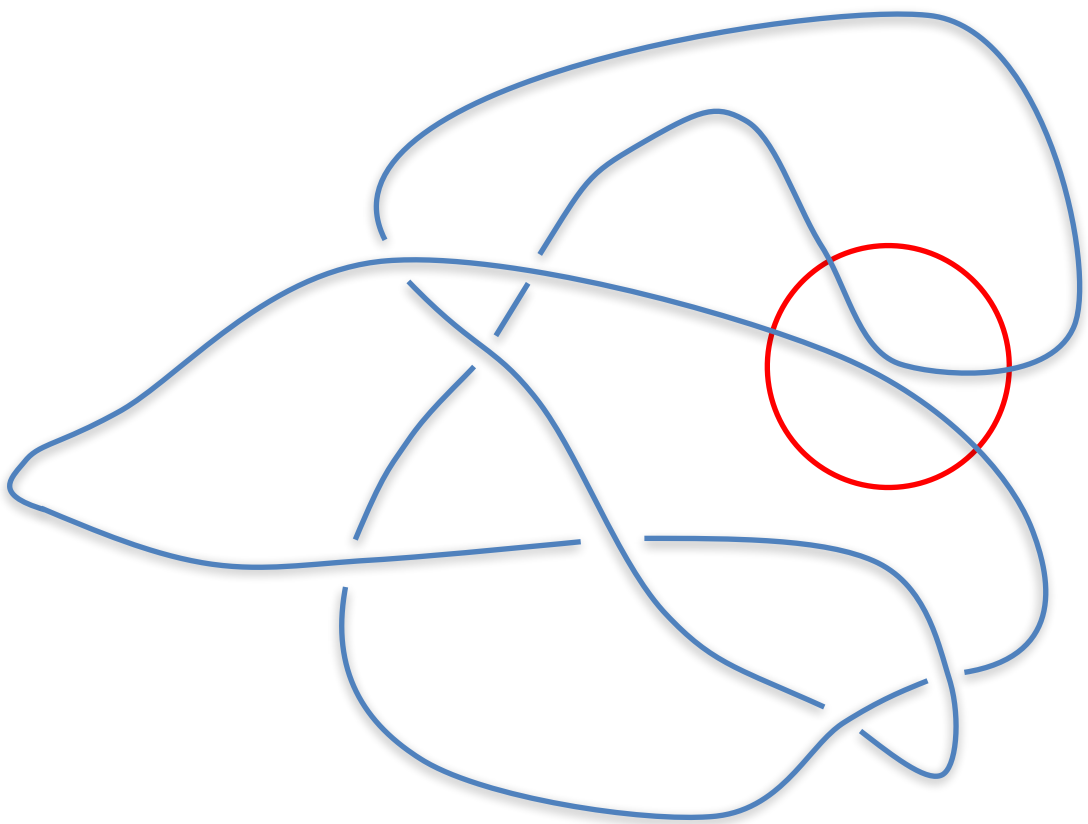
R1



Reidemeister moves

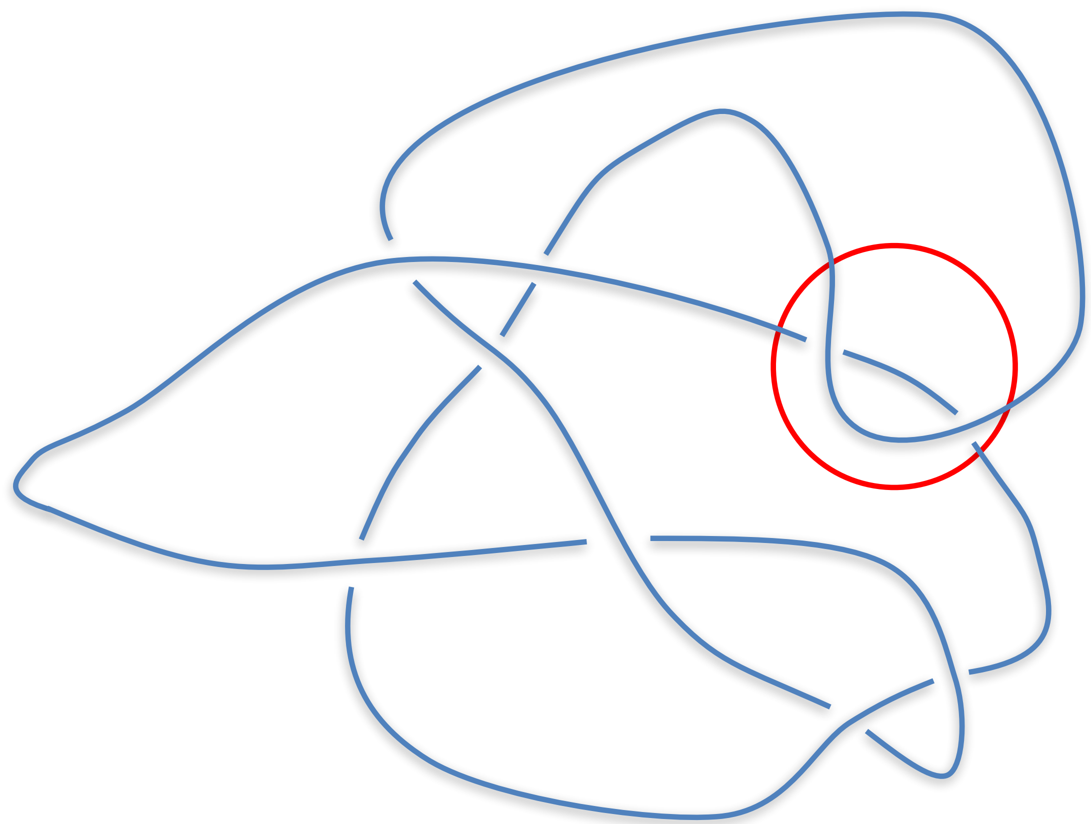


Reidemeister moves



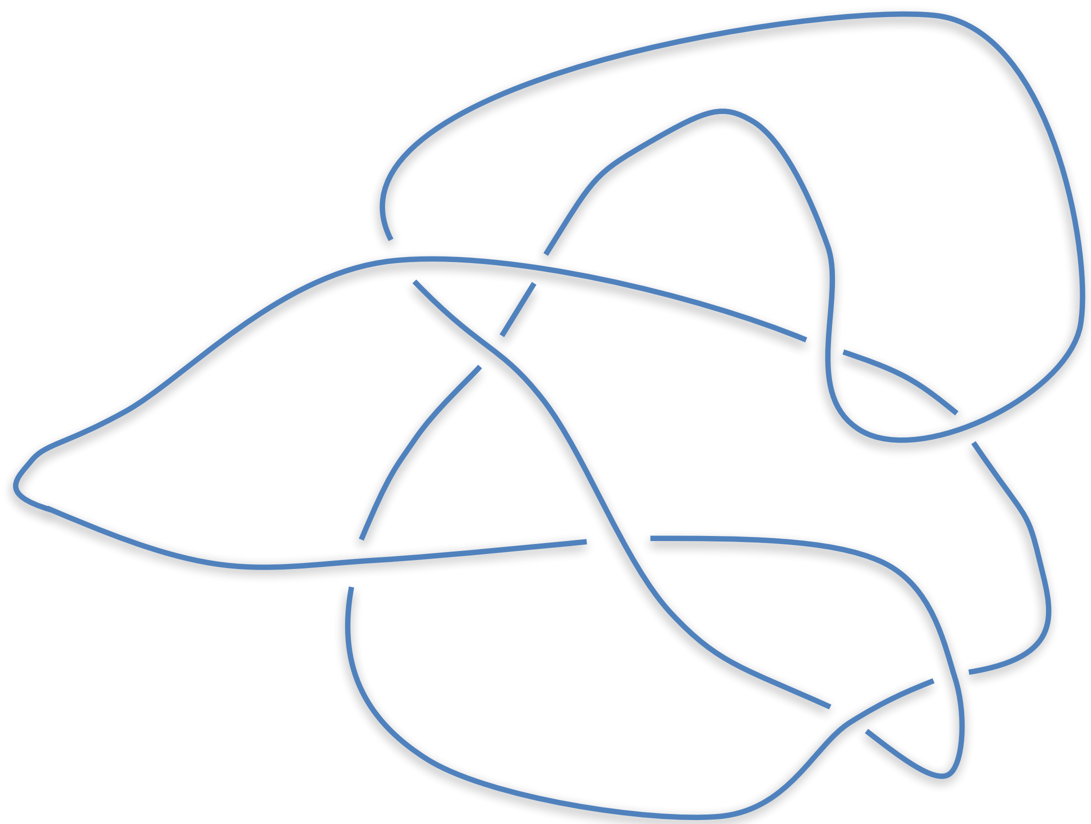
R2

Reidemeister moves



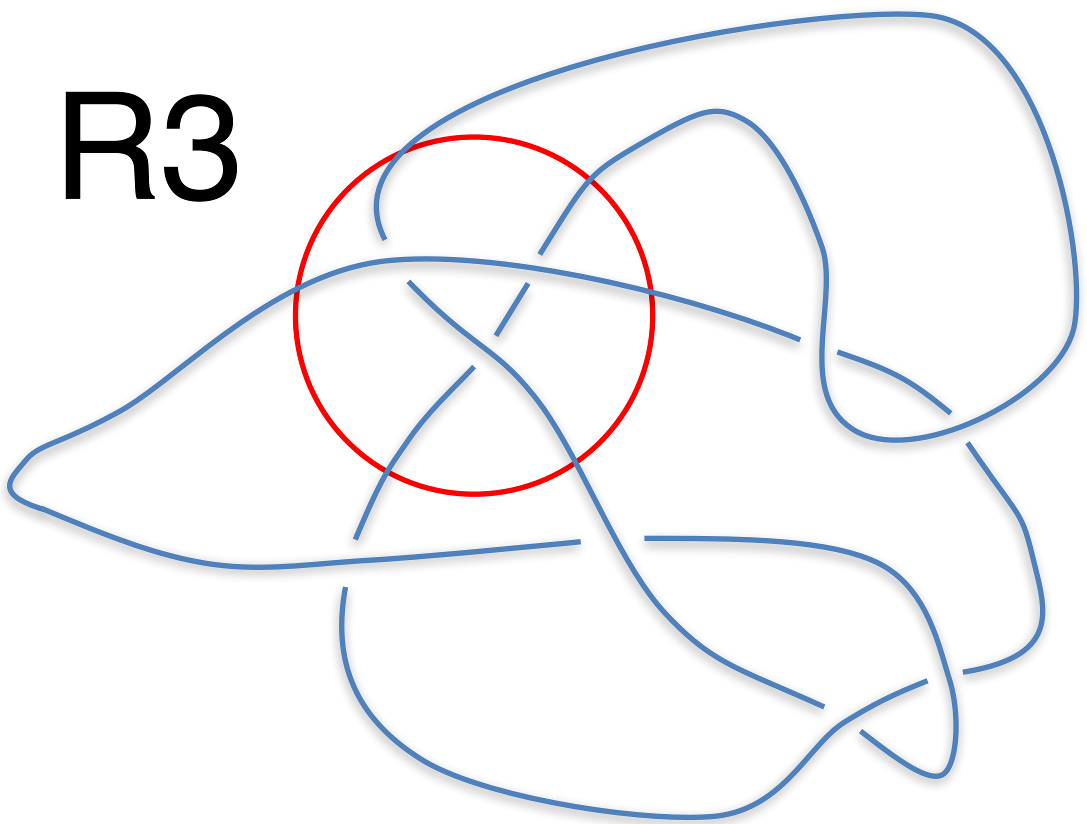
R2

Reidemeister moves



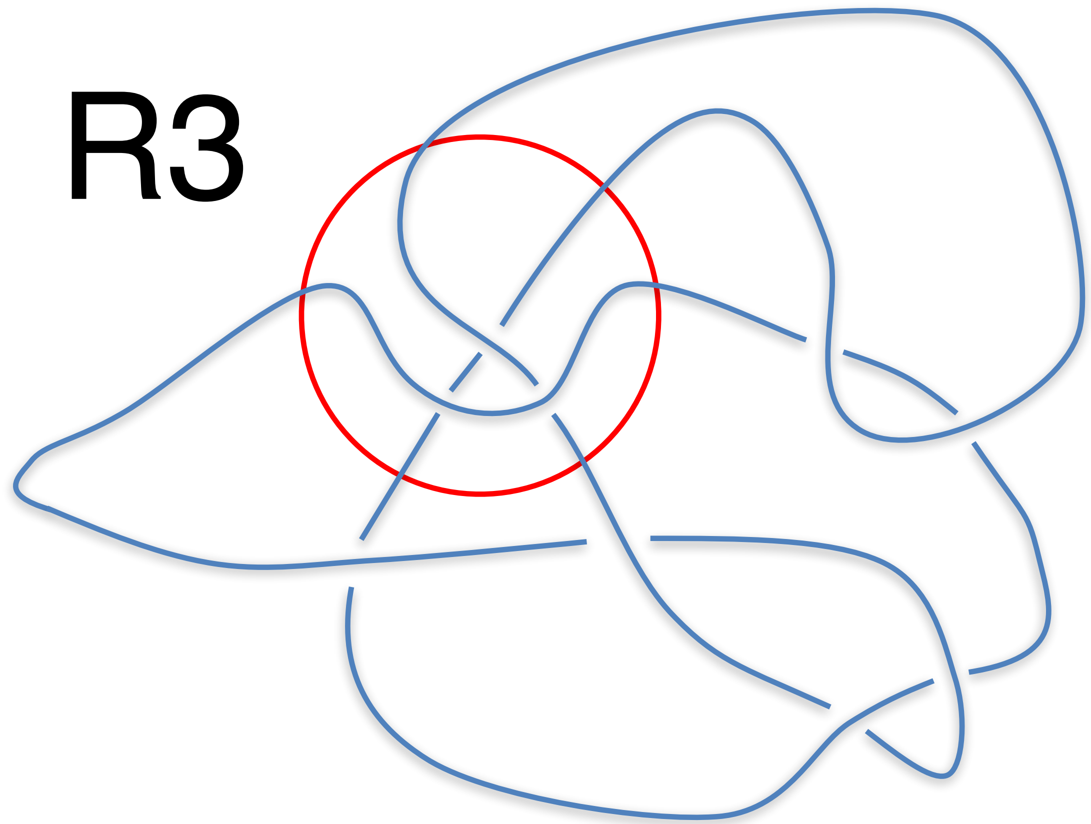
Reidemeister moves

R3

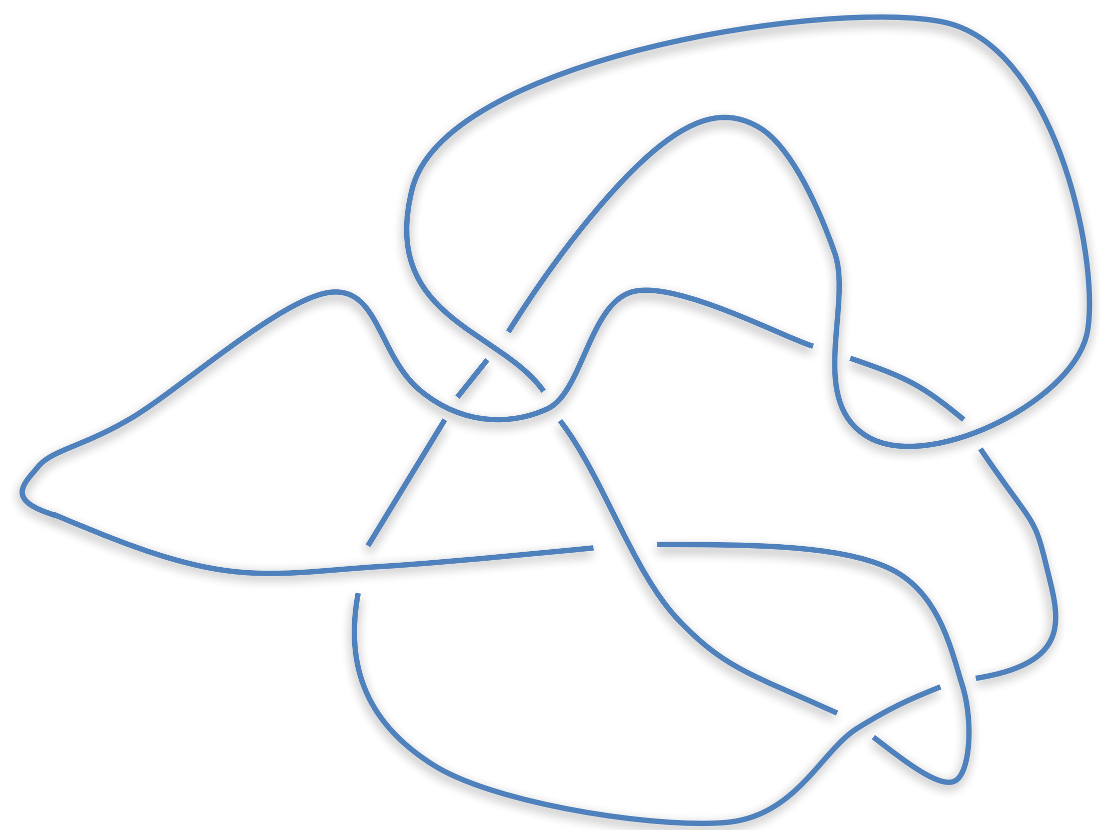


Reidemeister moves

R3

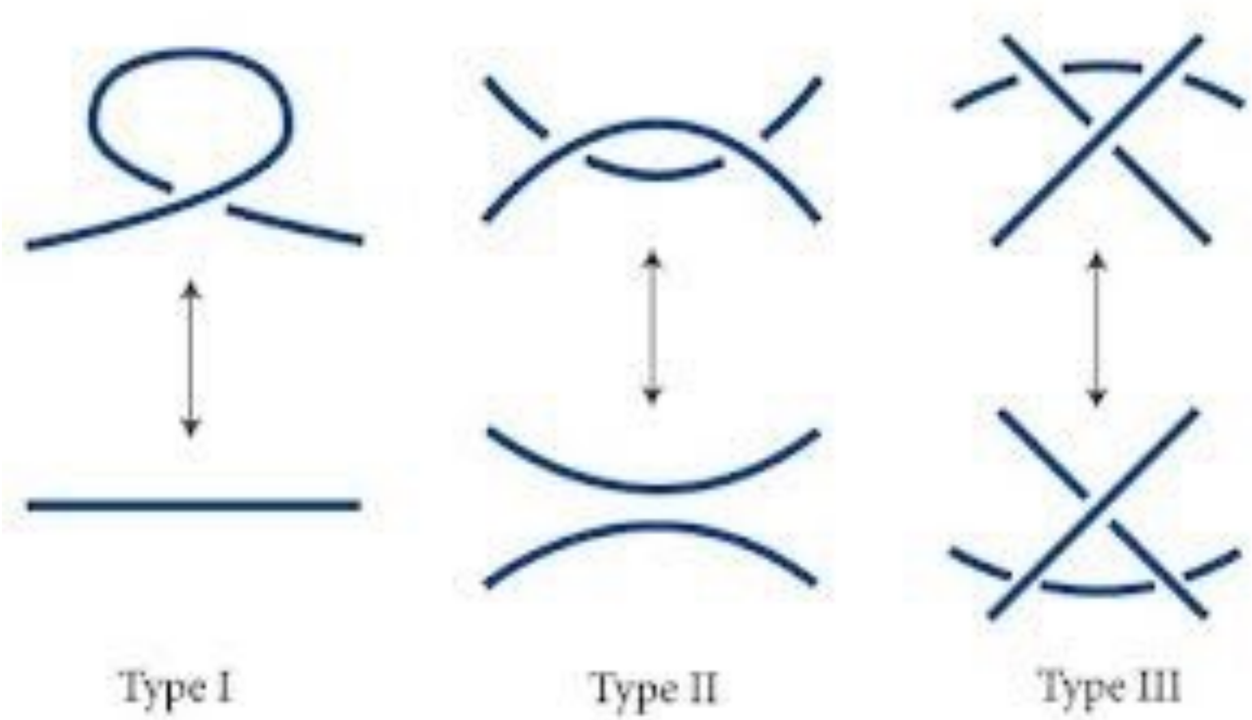


Reidemeister moves



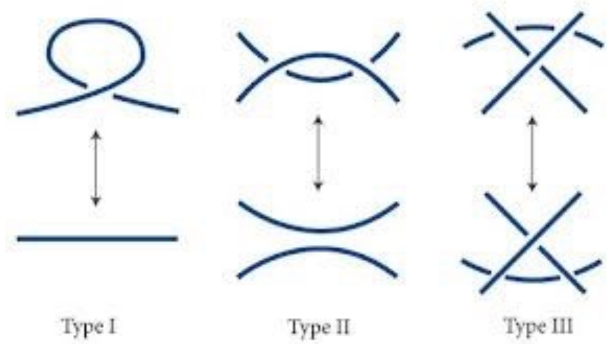
Reidemeister Moves

Theorem (Reidemeister, Alexander-Briggs, 1926)
Two diagrams representing the same knot are connected by a sequence of these three moves.

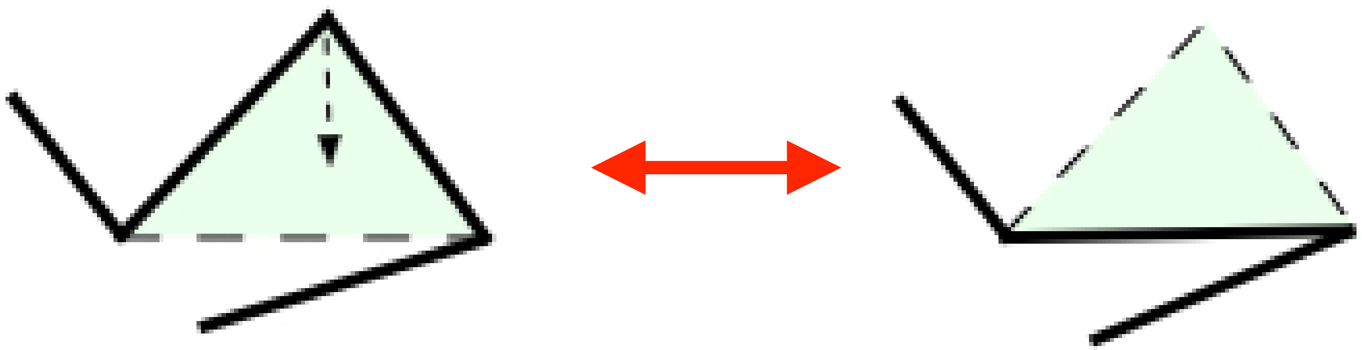


Reidemeister Moves

Theorem (Reidemeister, Alexander-Briggs, 1926)
Two diagrams representing the same knot are connected by a sequence of these three moves.

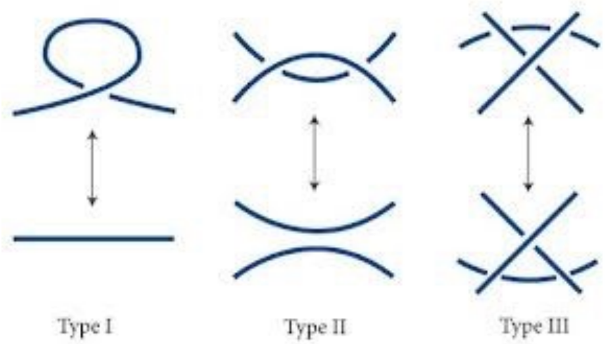


Proof: PL Knot equivalence consists of a sequence of elementary moves.

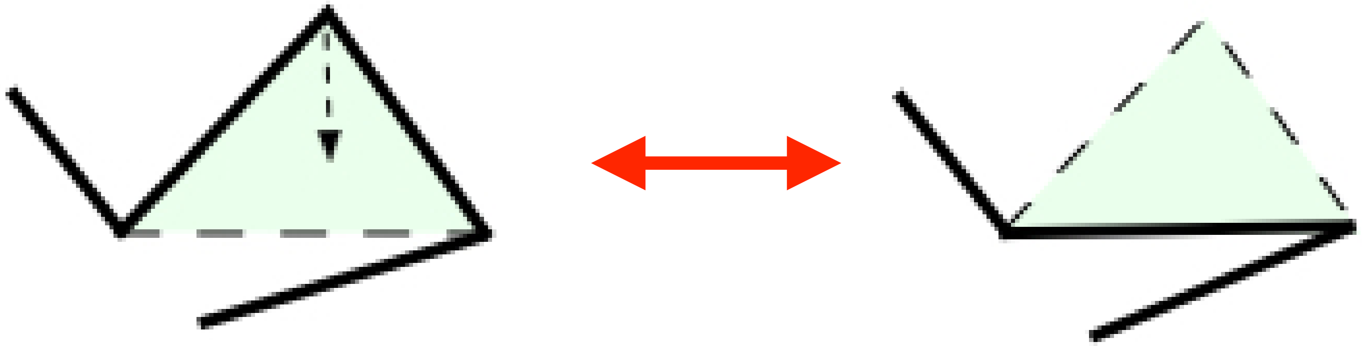


Reidemeister Moves

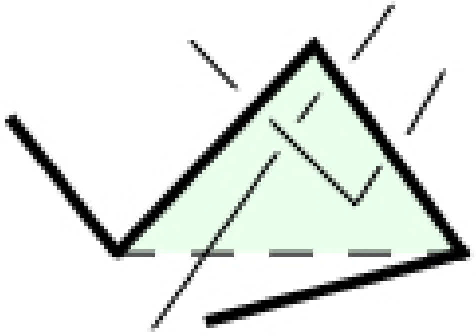
Theorem (Reidemeister, Alexander-Briggs, 1926)
Two diagrams representing the same knot are connected by a sequence of these three moves.



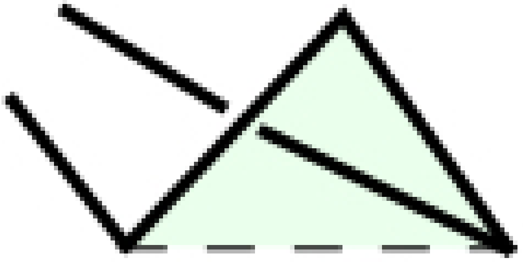
Proof: PL Knot equivalence consists of a sequence of elementary moves.



Each elementary move gives rise to finitely many Reidemeister moves.



Type II and III



Type I

Using Reidemeister moves for UNKNOTTING

Given an unknot diagram with n crossings, can we get an upper bound on how many Reidemeister moves are needed to trivialize it?

Using Reidemeister moves for UNKNOTTING

Given an unknot diagram with n crossings, how many Reidemeister moves are needed to trivialize it?

Can we find a function $U(n)$ such that any unknot diagram with n crossings can be transformed to the trivial diagram by at most $U(n)$ Reidemeister moves?

If yes, we have an algorithm. Just try all possible sequences of up to $U(n)$ Reidemeister moves and see if any give a trivial diagram.

Upper Bounds

What can we find on the internet?

From jewelerysecrets.com:

“It’s Not Difficult to get Rid of a Knot!
It’s just time consuming …”

Upper Bounds

To get an upper bound, unknot with Reidemester moves and count.

From jewelerysecrets.com:

“It’s Not Difficult to get Rid of a Knot!
It’s just time consuming …”

1) Lay the Chain Flat
2) Use Two Sewing Pins
3) Wiggle the Knot out

Upper Bounds

To get an upper bound, unknot with Reidemester moves and count.

From jewelerysecrets.com:

“It’s Not Difficult to get Rid of a Knot!
It’s just time consuming …”

1) Lay the Chain Flat (project the knot to the plane)
2) Use Two Sewing Pins (make the projection regular)
3) Wiggle the Knot out (use Reidemeister moves to unknot)

Upper Bounds

To get an upper bound, unknot with Reidemester moves and count.

From jewelerysecrets.com:

“It’s Not Difficult to get Rid of a Knot!
It’s just time consuming …”

1) Lay the Chain Flat (project the knot to the plane)
2) Use Two Sewing Pins (make the projection regular)
3) Wiggle the Knot out (use Reidemeister moves to unknot)

We will follow this advice.

Using Reidemeister moves for UNKNOTTING

Given an unknot diagram with n crossings, how many Reidemeister moves are needed to trivialize it?

Can we find a function $U(n)$ such that any unknot diagram with n crossings can be transformed to the trivial diagram by at most $U(n)$ Reidemeister moves?

If yes, we have an algorithm. Just try all possible sequences of up to $U(n)$ Reidemeister moves and see if any give a trivial diagram.

How many do we need?

Using Reidemeister moves for UNKNOTTING

Given an unknot diagram with n crossings, how many Reidemeister moves are needed to trivialize it?

Can we find a function $U(n)$ such that any unknot diagram with n crossings can be transformed to the trivial diagram by at most $U(n)$ Reidemeister moves?

If yes, we have an algorithm. Just try all possible sequences of up to $U(n)$ Reidemeister moves and see if any give a trivial diagram.

First bound: Need at least $n/2$ since each move reduces crossings by at most 2.

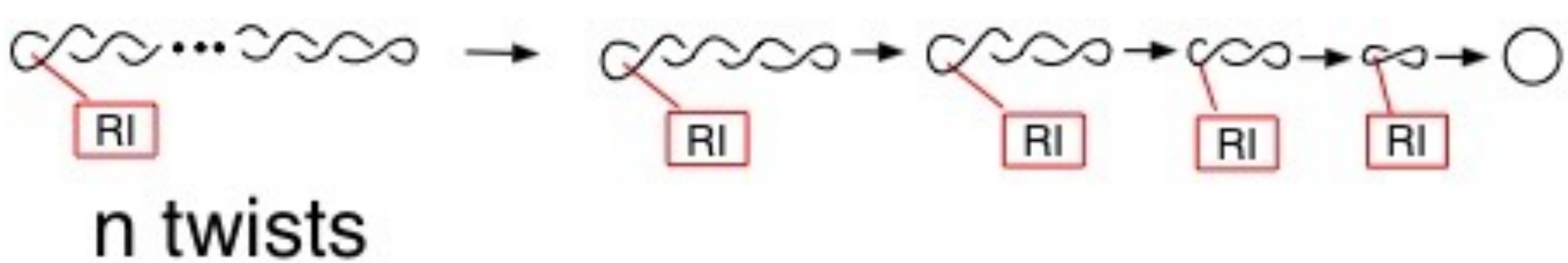
Using Reidemeister moves for UNKNOTTING

Given an unknot diagram with n crossings, how many Reidemeister moves are needed to trivialize it?

Can we find a function $U(n)$ such that any unknot diagram with n crossings can be transformed to the trivial diagram by at most $U(n)$ Reidemeister moves?

If yes, we have an algorithm. Just try all possible sequences of up to $U(n)$ Reidemeister moves and see if any give a trivial diagram.

First bound: Need at least $n/2$ since each move reduces crossings by at most 2.
Some unknots with n crossings require n Reidemeister moves to trivialize:



Using Reidemeister moves for UNKNOTTING

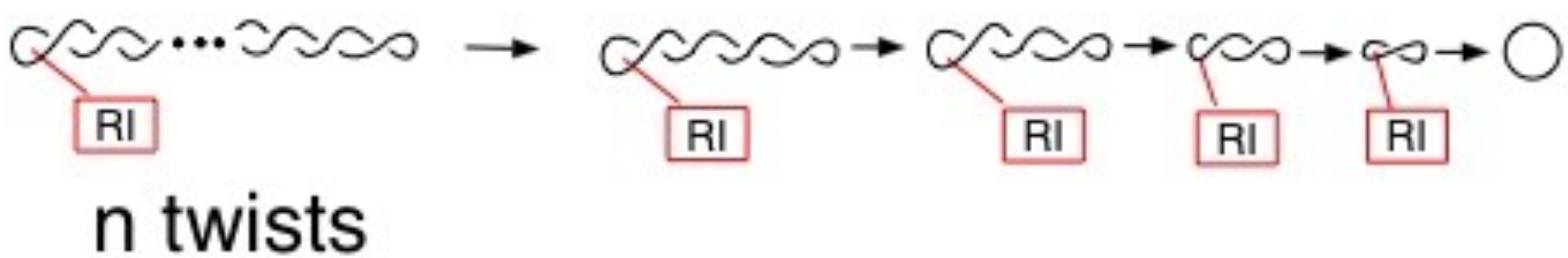
Given an unknot diagram with n crossings, how many Reidemeister moves are needed to trivialize it?

Can we find a function $U(n)$ such that any unknot diagram with n crossings can be transformed to the trivial diagram by at most $U(n)$ Reidemeister moves?

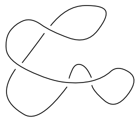
If yes, we have an algorithm. Just try all possible sequences of up to $U(n)$ Reidemeister moves and see if any give a trivial diagram.

First bound: Need at least $n/2$ since each move reduces crossings by at most 2.

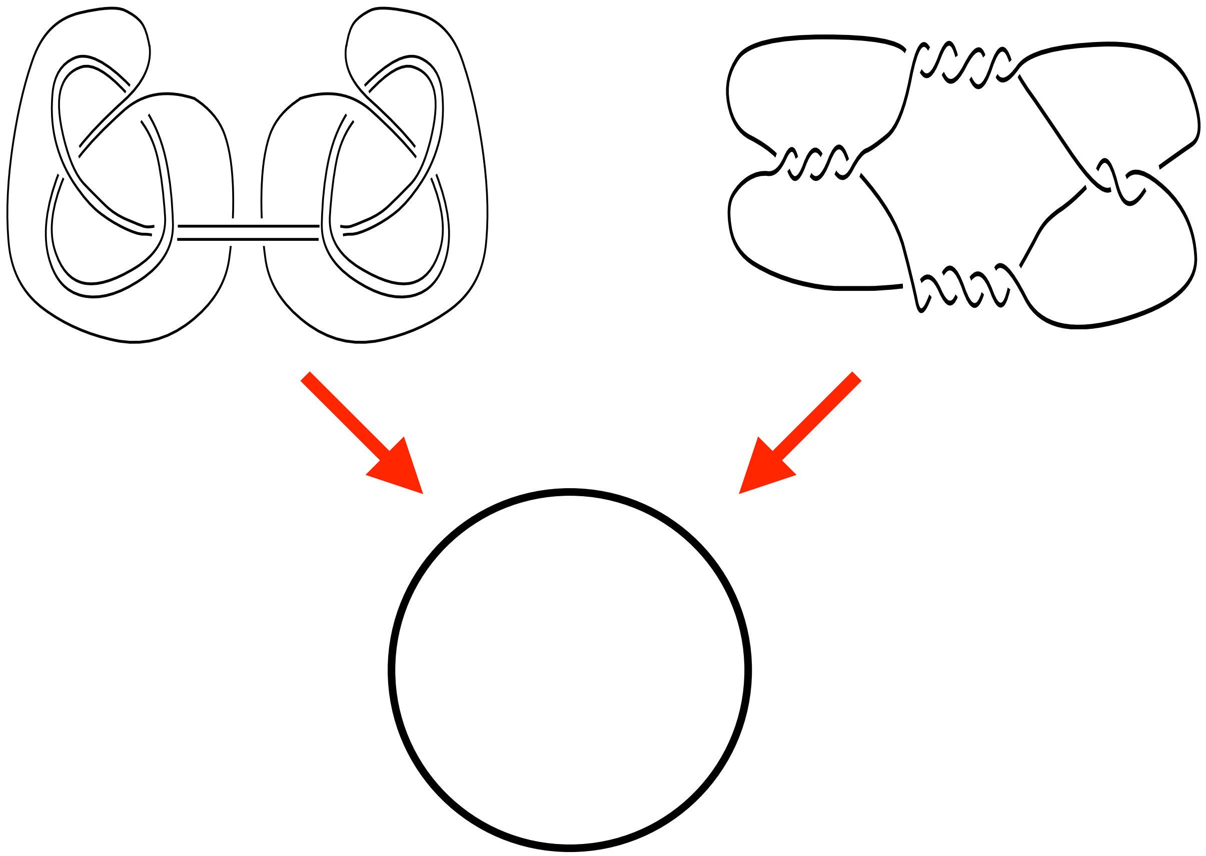
Some unknots with n crossings require n Reidemeister moves to trivialize:



This unknotting sequence monotonically reduces crossing number.

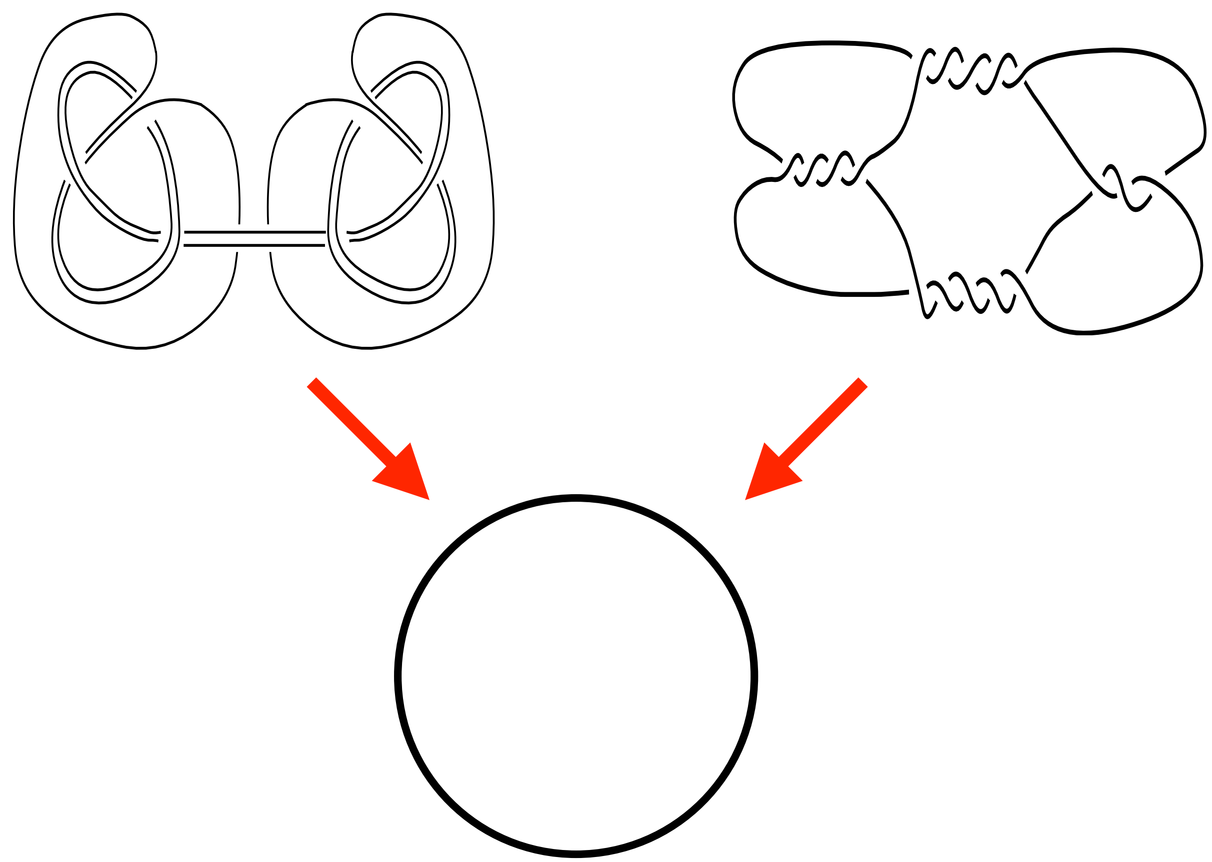
Using Reidemeister moves for UNKNOTTING 

Some unknot diagrams require that the crossing number *increase* as they are transformed to the trivial diagram by Reidemeister moves.



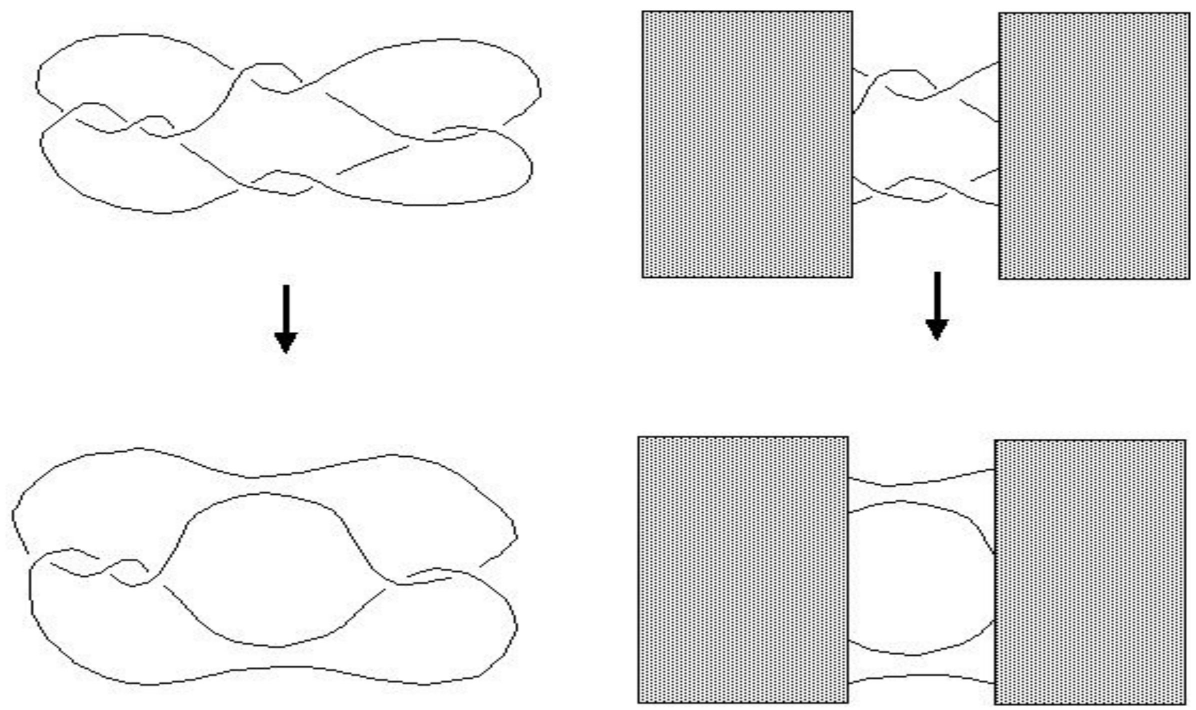
Using Reidemeister moves for UNKNOTTING

Some unknot diagrams require that the crossing number *increase* as they are transformed to the trivial diagram by Reidemeister moves.



Question Can we enlarge the set of moves to allow for monotone descent in crossing number?

A possible extra move: Flype



A flype

Can we extend Reidemeister moves of types 1, 2 and 3, adding moves of type 4, 5, … N so that together with Reidemeister moves we get monotone descent for the number of crossings?

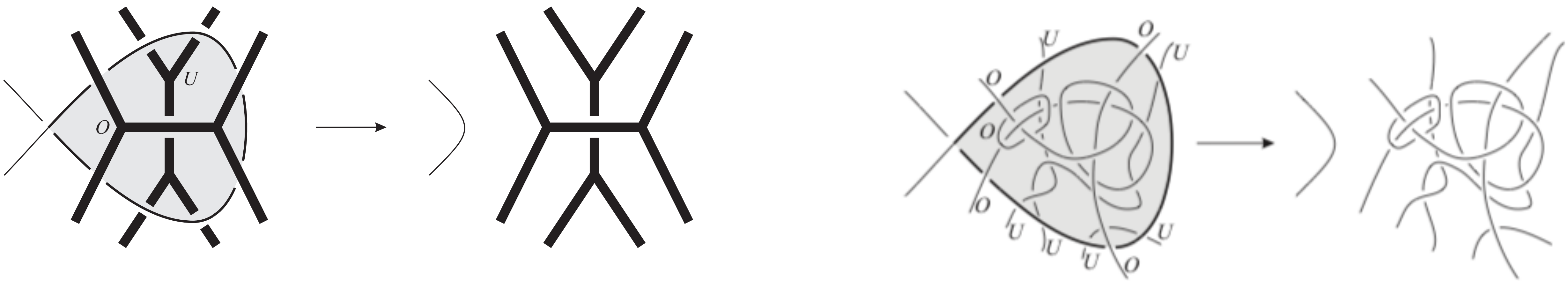
Not known if we can do this. If yes, we would have a fast algorithm.

Possible new moves: Flype Moves

Flypes are good at messing up puppets.

Possible new moves

Petronio-Zanellati 2016: Suggested 4 new moves.



Move Z_1

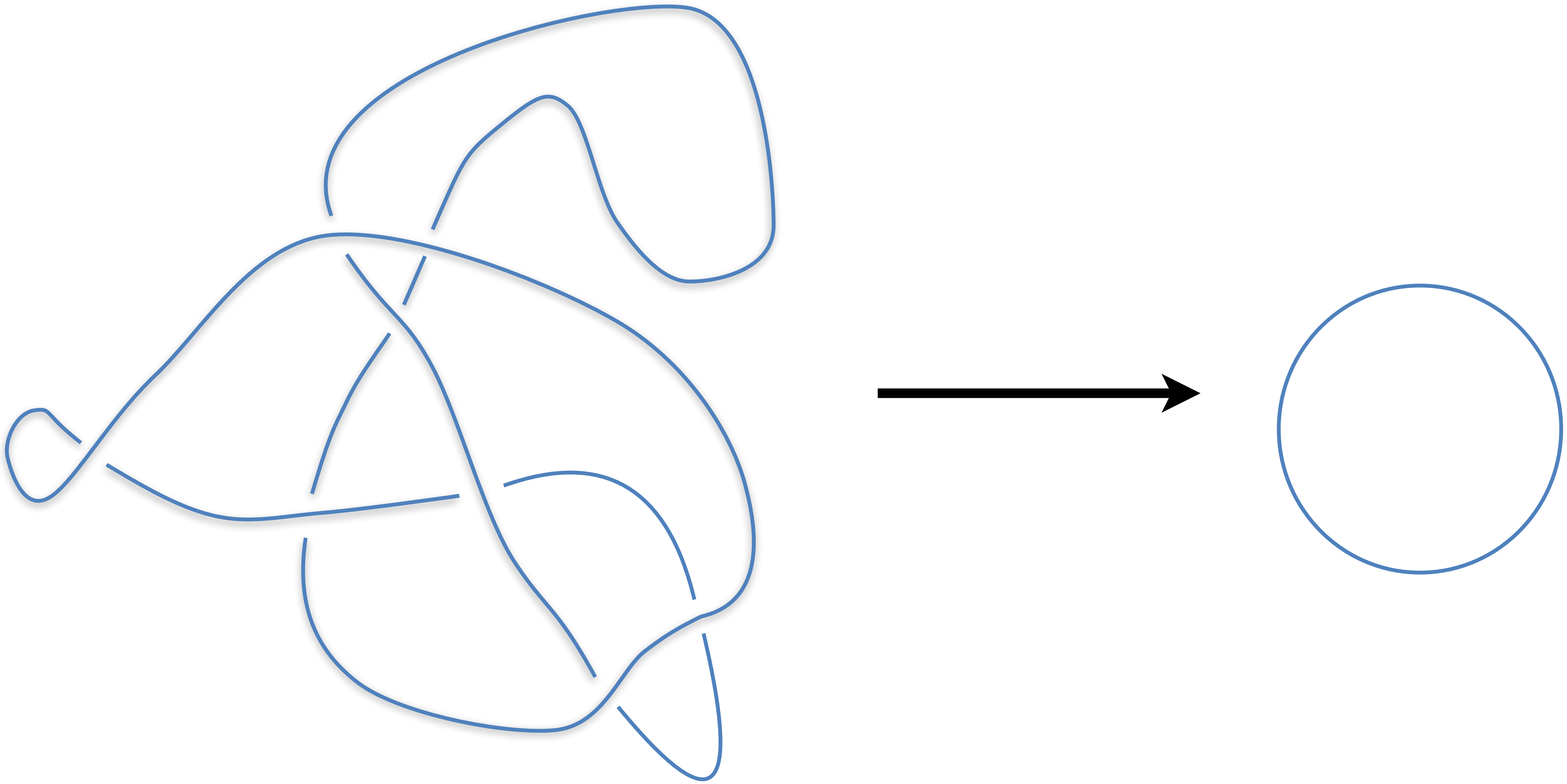
Windows software at http://www.zanellati.it/knot/index.htm

There is no proof that this method always works.

In practice, this and other methods (Regina (Burton), Snappea (Weeks, Culler, Dunfield), …) work surprisingly well. They can handle many diagrams with several hundred crossings.

Question. Is there a simple method for unknotting that we have missed?

Bounding the number of required Reidemeister moves



Question:

How many Reidemeister moves are needed to change an *n*-crossing unknot diagram to a trivial diagram?

Upper Bounds

Find $U(n)$ so that: The number of Reidemeister moves needed to change an n-crossing unknot diagram to a trivial diagram is at most $U(n)$.

A bound on $U(n)$ implies an unknotting algorithm.
Namely try all sequences of up to $U(n)$ Reidemeister moves.

Haken's method implies bounds

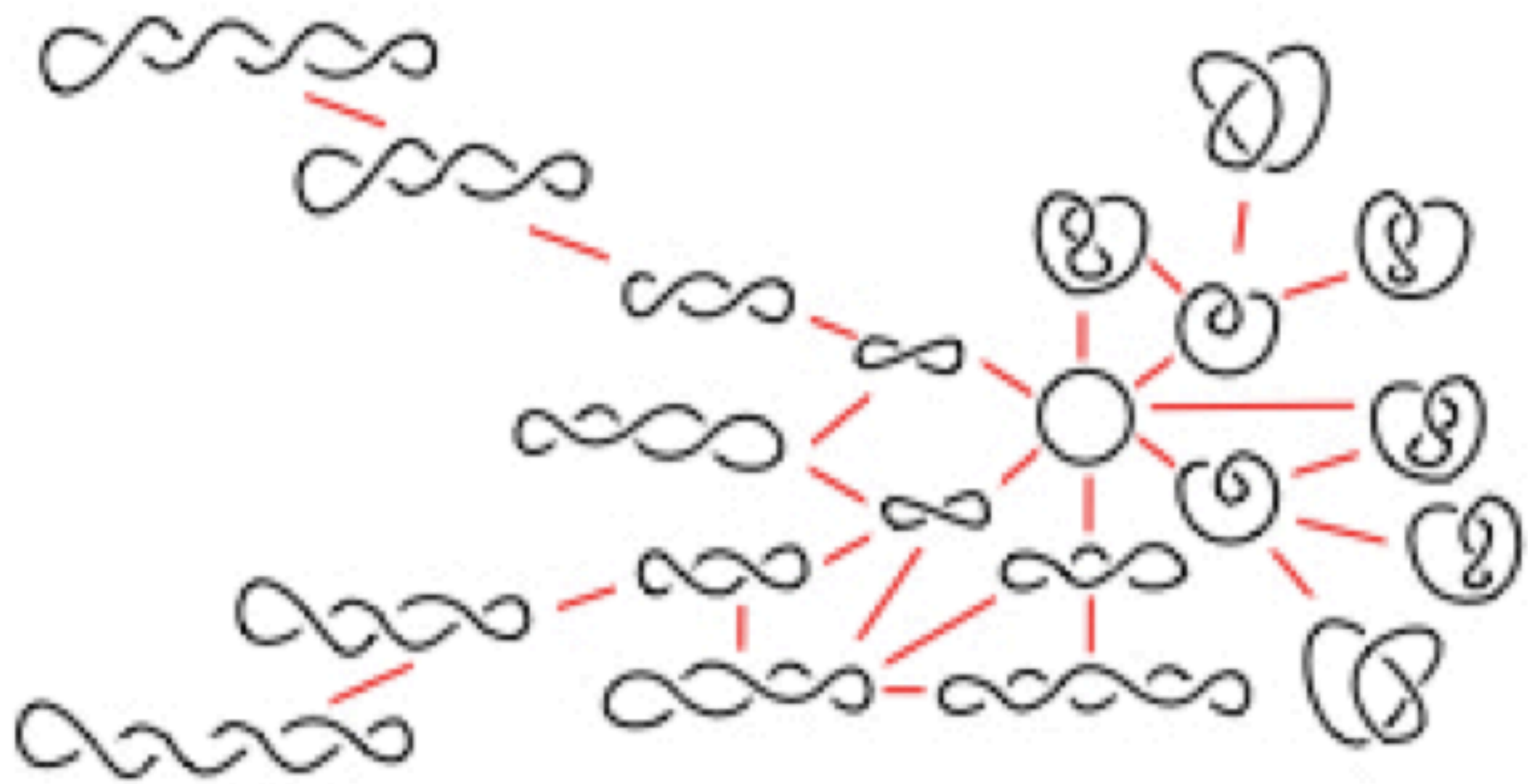
Theorem H-Lagarias, 2001
$U(n) \leq c^n$ $\qquad c = 2^{10^{11}}$

Recently there was a major improvement. The bound was improved from exponential to polynomial.

Theorem Lackenby 2012
$U(n) \leq (231n)^{11}$

Upper Bounds

The complex of Unknot Diagrams



What properties does this graph have?

$U(n) =$ maximal distance of an n-crossing diagram to O.