Knot recognition by SAT and ATP

David Stanovský

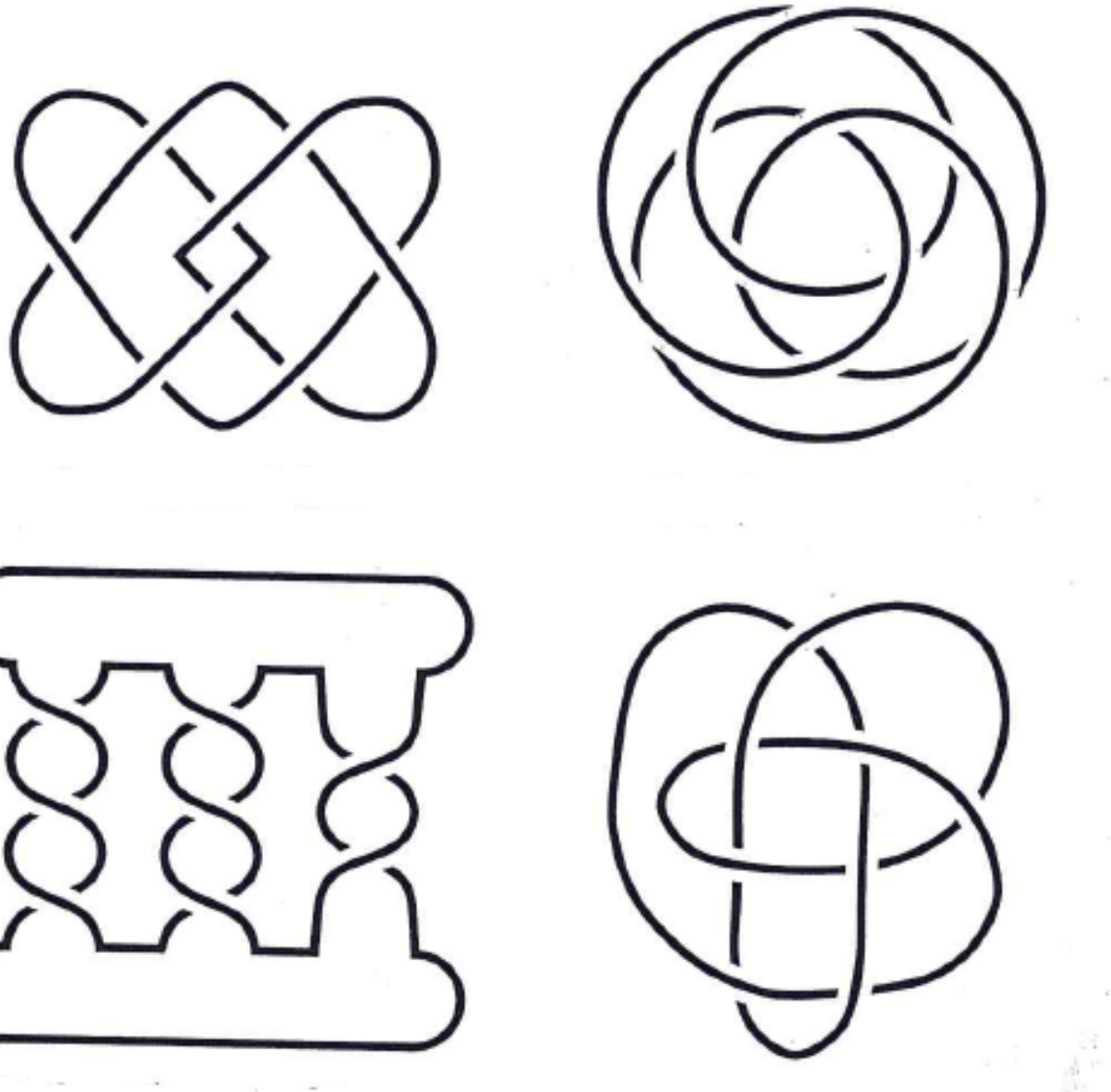
Charles University, Prague, Czech Republic

based on joint research with
Andrew Fish (Brighton) and Alexei Lisitsa (Liverpool)

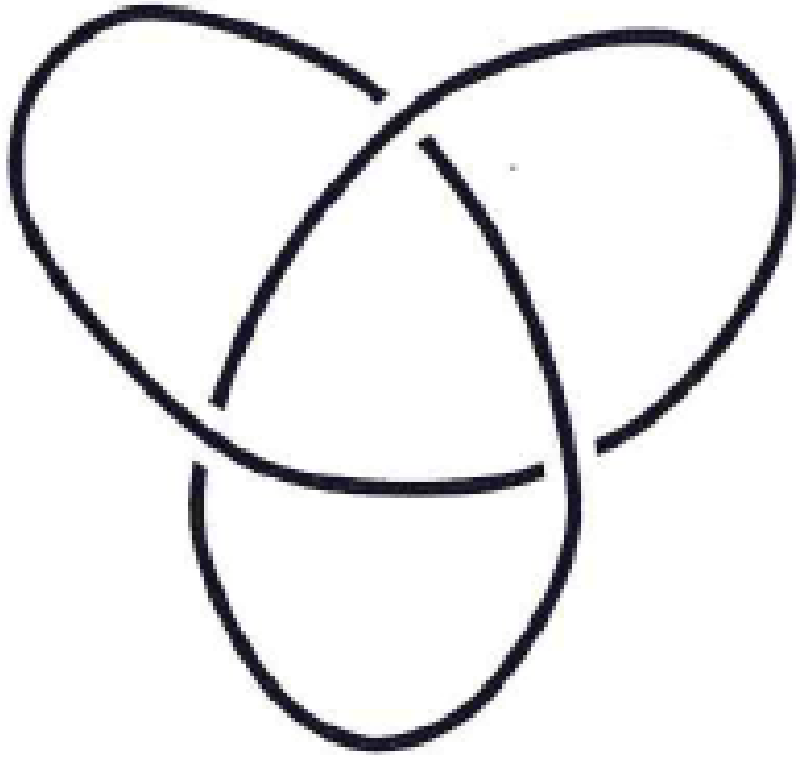
May 30, 2016

Is it really knotted?

Four pictures, one knot



Is it really knotted?



If you think it cannot be untangled, PROVE IT!

Knot recognition

Knot equivalence = a continuous deformation of space that transforms one knot into the other.

Fundamental Problem

Given two knots (or knot diagrams), are they equivalent?

Is it (algorithmically) decidable?

If so, what is the complexity?

Knot recognition

Knot equivalence = a continuous deformation of space that transforms one knot into the other.

Fundamental Problem

Given two knots (or knot diagrams), are they equivalent?

Is it (algorithmically) decidable?

Yes, very hard to prove. (Haken, 1962)

If so, what is the complexity?

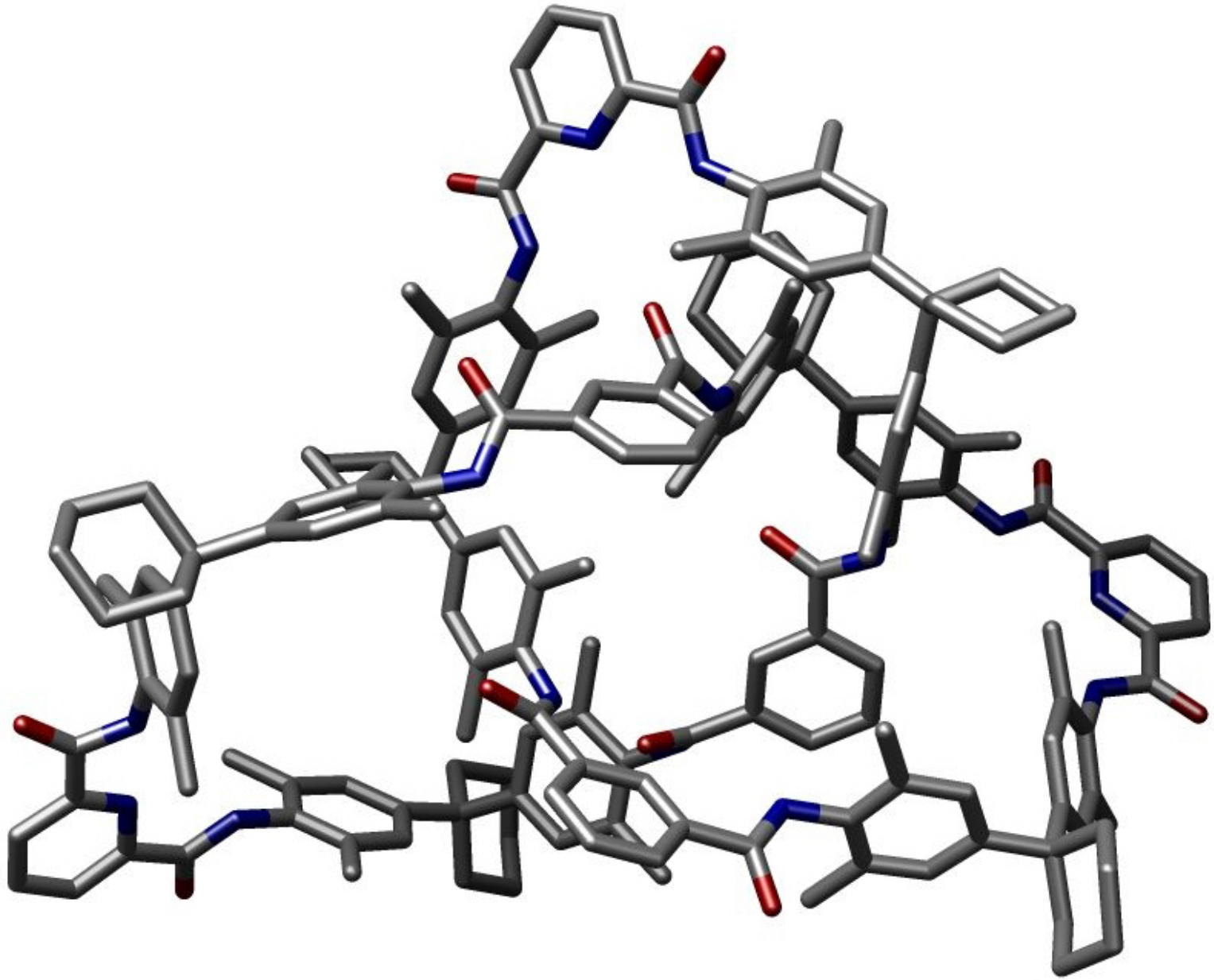
Nobody knows. No efficient algorithm known.

What is it good for?

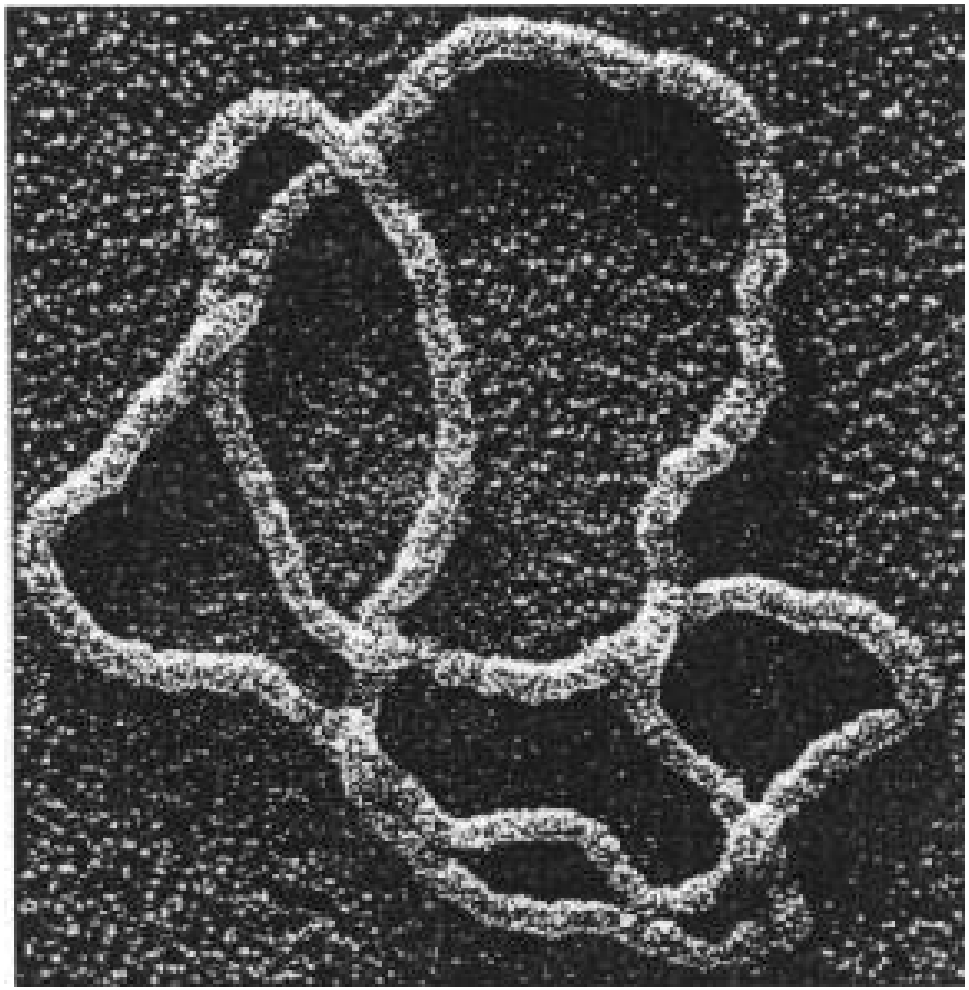
What is it good for?

I don't care. I am a mathematician.

Knots are in chemistry



Knots are in biology



... with applications towards antibiotics production (believe or not)

Knots are everywhere

... with applications towards black magic (believe or not)

Classical approach to knot recognition

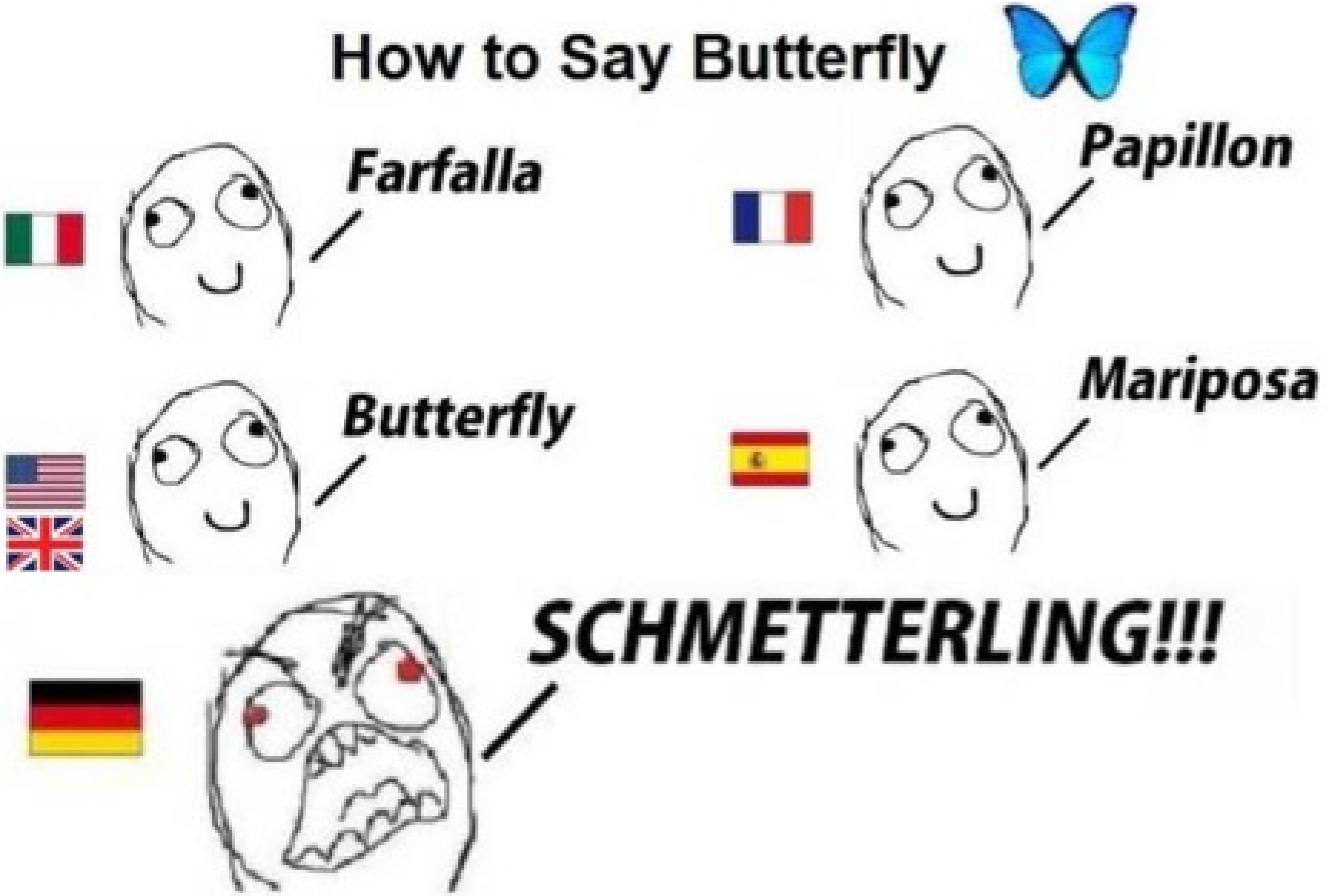
Develop *invariants*, properties shared by equivalent knots.

$$K_1 \sim K_2 \quad \text{implies} \quad P(K_1) = P(K_2)$$

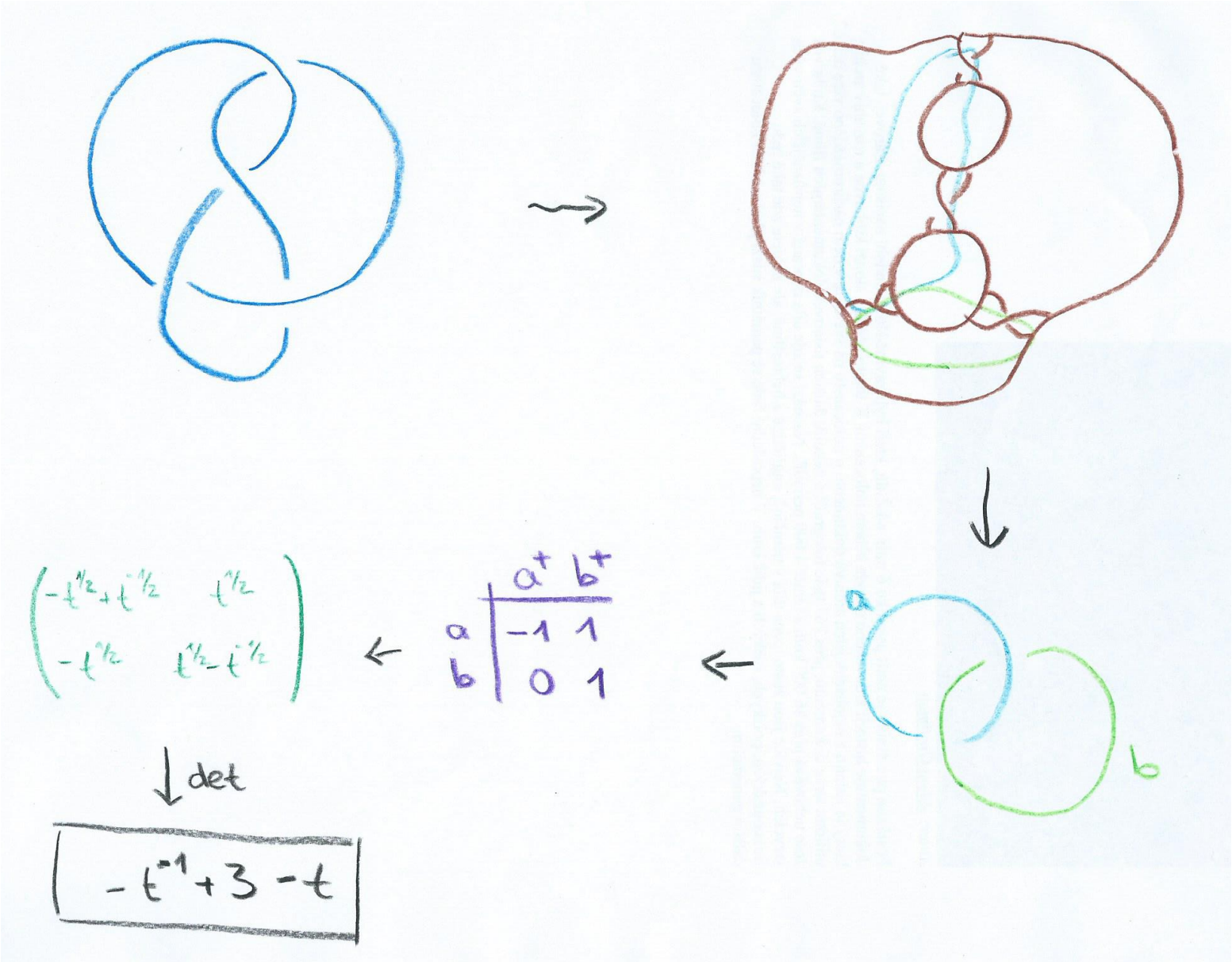
Classical invariants use various algebraic constructions to code some of the topological properties of a knot.

- the Alexander, Jones and other polynomials
- the fundamental group of the knot complement
- Khovanov homology, Heegaard-Floer homology, ...

Trade-off between complexity and ability to recognize knots.


How to Say Butterfly
Farfalla
Papillon
Butterfly
Mariposa
SCHMETTERLING!!!

Alexander polynomial



Reidemester moves

Knots are usually displayed by a *regular* projection into a plane.

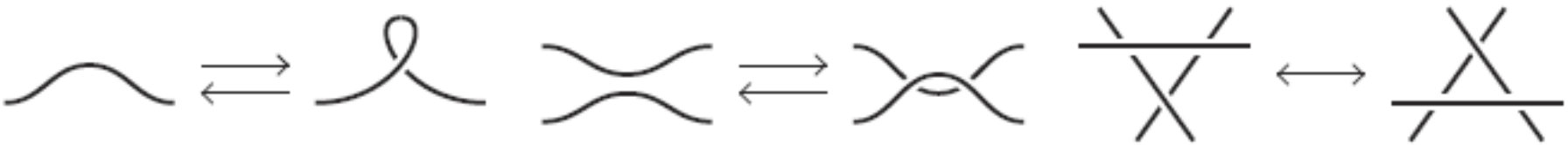
Theorem (Reidemeister 1926, Alexander-Brigs 1927)

$K_1 \sim K_2$ if and only if they are related by a finite sequence of Reidemeister moves:

I. *twist/untwist a loop;*

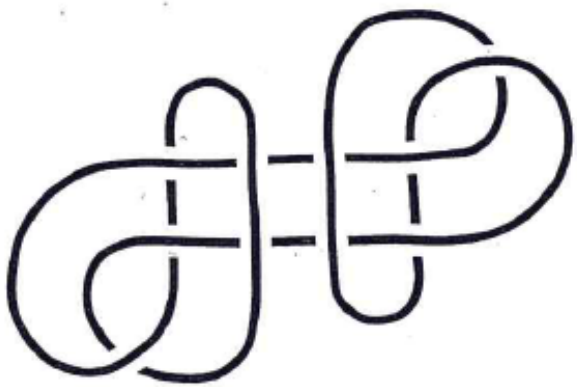
II. *move a string over/under another;*

III. *move a string over/under a crossing.*



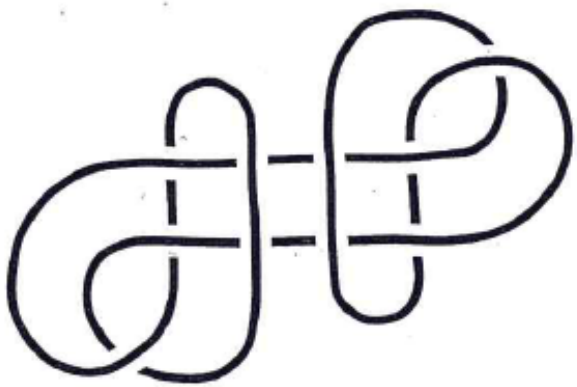
Reidemeister moves, where is the problem?

Bad news: When unknotting, $\mathrm{cross}(K)$ may increase



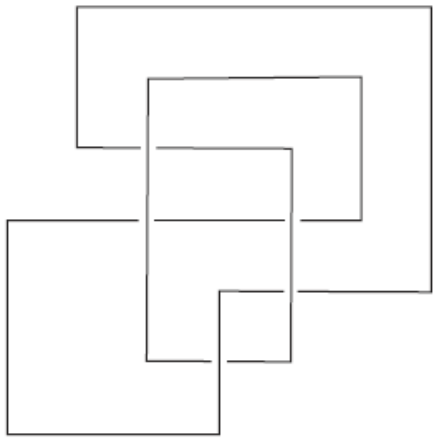
Reidemeister moves, where is the problem?

Bad news: When unknotting, $\mathrm{cross}(K)$ may increase



Good news: Lackenby (2013): not too much

Lackenby's idea: a special type of diagrams and moves (Dynnikov's theory)



Reidemeister moves, algorithmically?

Fact

Assume $K_1 \sim K_2$ *iff related by a sequence of at most* $f(\mathrm{cross}(K_1) + \mathrm{cross}(K_2))$ *Reidemeister moves, where* f *is computable. Then there is an algorithm to decide* $K_1 \sim K_2$*. (Very impractical one.)*

Finding such f is very difficult, even for $K_2 = \bigcirc$:

- Haas-Lagarias (2001): f exponential, $f(n) = 2^{10^{11} n}$
  - hence, $\sim \bigcirc$ is decidable
- Lackenby (2013): f polynomial, $f(n) = (231n)^{11}$
  - hence, $\sim \bigcirc$ is an NP problem

NP = there is a polynomial size certificate that can be checked in polynomial time

- Hass-Novik (2010): quadratic lower bound for unknot diagrams ($\exists K^{(n)} \sim \bigcirc$, $n = \mathrm{cross}(K^{(n)})$, with at least $n^2/25$ moves)

Recognizing knots, summary

Fundamental Problem

Given K_1, K_2, are they equivalent?
Given K, is $K \sim \bigcirc$?

Is it (algorithmically) decidable?

If so, what is the complexity?

- Haken (1961): $\sim \bigcirc$ is decidable (in EXP-time)
- Haken (1962): $\sim$ is decidable (in EXP-time)
- Haas-Lagarias-Pippinger (1999): $\sim \bigcirc$ is in NP
- Lackenby (2013): $\sim \bigcirc$ is in NP by bounding Reidemeister
- Lackenby (2013): $\sim$ is decideable by bounding Reidemeister

Recognizing knots, summary

Fundamental Problem

Given K_1, K_2, are they equivalent?
Given K, is $K \sim \bigcirc$?

Is it (algorithmically) decidable?

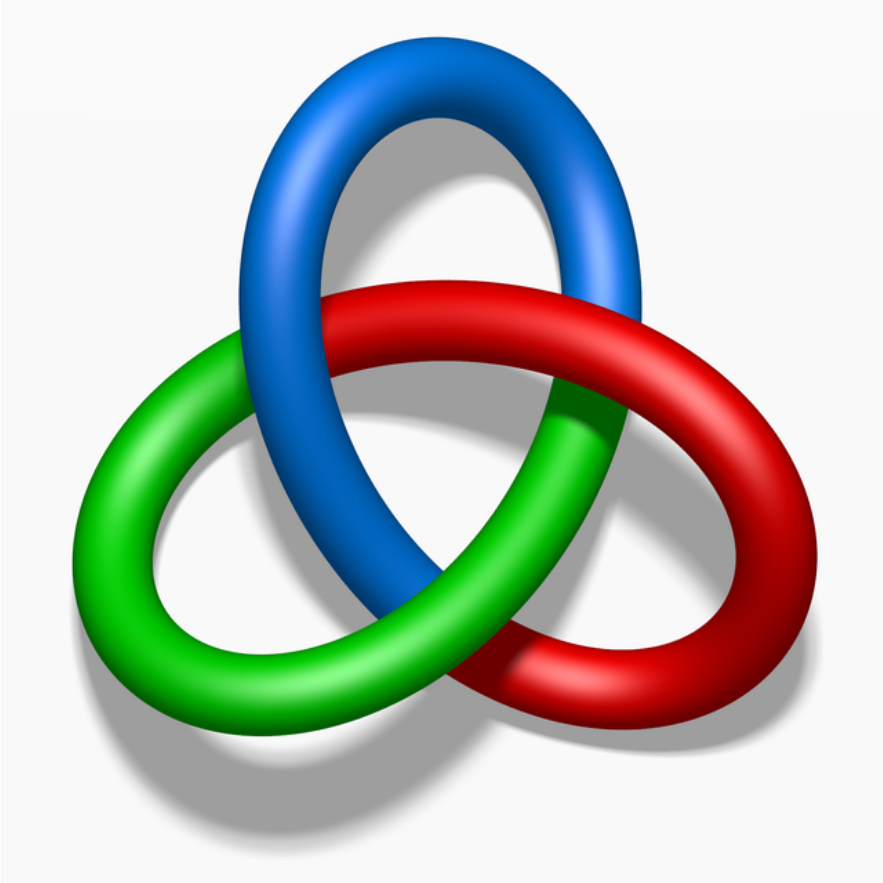
If so, what is the complexity?

- Haken (1961): $\sim \bigcirc$ is decidable (in EXP-time)
- Haken (1962): $\sim$ is decidable (in EXP-time)
- Haas-Lagarias-Pippinger (1999): $\sim \bigcirc$ is in NP
- Lackenby (2013): $\sim \bigcirc$ is in NP by bounding Reidemeister
- Lackenby (2013): $\sim$ is decideable by bounding Reidemeister

- Agol (2002, not published): $\not\sim \bigcirc$ is in NP assuming GRH
- Kuperberg (2011): $\not\sim \bigcirc$ is in NP assuming GRH

WHAT COMES NEXT?

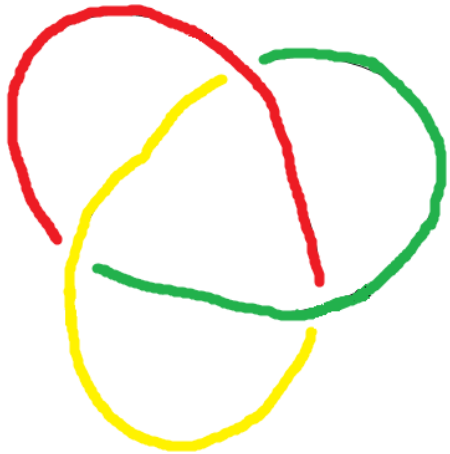
- a whole new *combinatorial* approach to the knot recognition problem
- (re)explanation of Kuperberg's certificate
- a practical tool for knot recognition problem (via ATP and SAT)




SCORE: 75
HI-SCORE:1201


PARATROOPER
by
Greg Kuperberg
PRESS 'I' FOR INSTRUCTIONS
PRESS space bar FOR KEYBOARD PLAY
OR joystick button FOR JOYSTICK PLAY
OR ctrl-J FOR JOYSTICK adjustment
(C)1982 ORION SOFTWARE, INC.

Combinatorial approach: 3-coloring

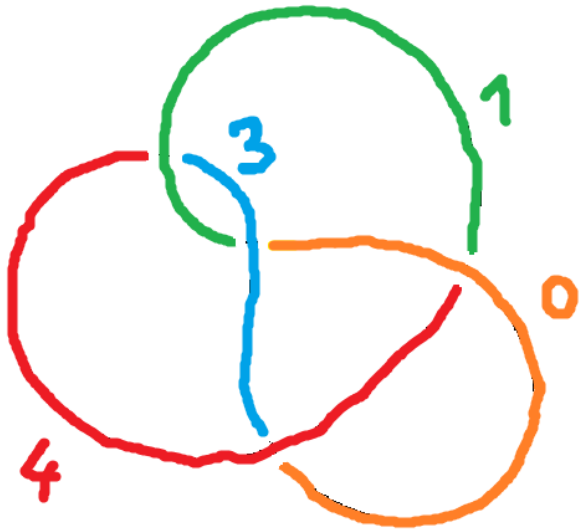


To every arc, assign one of three colors in a way that

every crossing has one or three colors.

Invariant: count non-trivial (non-monochromatic) colorings.

Combinatorial approach: n-coloring

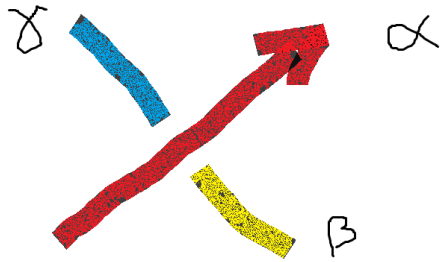


To every arc, assign one of n colors, $0, \ldots, n-1$, in a way that

at every crossing, $2 \cdot$ bridge $=$ left $+$ right, modulo n

Invariant: count non-trivial colorings.

Combinatorial approach: quandle coloring

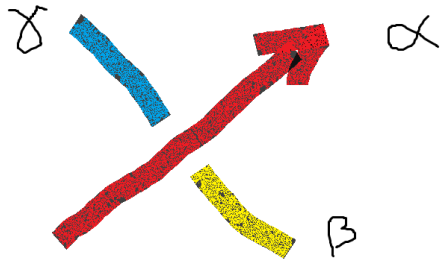


To every arc, assign one of the colors from a set C in a way that

$$(c(\alpha), c(\beta), c(\gamma)) \in T$$

Invariant: count non-trivial colorings, $col_Q(K)$. Really?

Combinatorial approach: quandle coloring



To every arc, assign one of the colors from a set C in a way that

$$(c(\alpha), c(\beta), c(\gamma)) \in T$$

Invariant: count non-trivial colorings, $col_Q(K)$. Really?

Fact (implicitly Joyce, Matveev ('82), explicitly Fenn-Rourke ('92))

Coloring by (C, T) is an invariant if T is a graph of an operation $*$ such that for every x, y, z

- $x * x = x$
- *there is a unique u such that $x * u = y$*
- $x * (y * z) = (x * y) * (x * z)$

Quandles

Such algebraic objects are called quandles.

- $x * x = x$
- there is a unique u such that $x * u = y$
- $x * (y * z) = (x * y) * (x * z)$

Fact (implicitly Joyce, Matveev ('82), explicitly Fenn-Rourke ('92))

Coloring by (C, T) is an invariant if T is a graph of a quandle.

With more care, one can formulate the fact with "if and only if".

Quandles

Such algebraic objects are called quandles.

- $x * x = x$
- there is a unique u such that $x * u = y$
- $x * (y * z) = (x * y) * (x * z)$

Fact (implicitly Joyce, Matveev ('82), explicitly Fenn-Rourke ('92))

Coloring by (C, T) is an invariant if T is a graph of a quandle.

With more care, one can formulate the fact with "if and only if".

Theorem (dtto)

$K \not\sim \bigcirc$ if and only if there is a quandle Q with $col_Q(K) > 0$.

Quandles

Theorem (Hulpke, S., Vojtěchovský; and other alternative approaches)

We know fairly well what quandles are.

Essentially, connected quandles correspond uniquely to a choice of a transitive group G and a fixed $\zeta \in Z(G_e)$.
You can ignore the non-connected ones for coloring purposes.

Knot recognition algorithm

Parameter: a (potentially infinite) set of quandles $\mathcal{Q}$

IN: two knots K_1, K_2

run over $Q \in \mathcal{Q}$

    if $col_Q(K_1) \neq col_Q(K_2)$, then return "they are different"

return "I have no idea"

Knot recognition algorithm

Parameter: a (potentially infinite) set of quandles $\mathcal{Q}$

IN: two knots K_1, K_2

run over $Q \in \mathcal{Q}$
    if $col_Q(K_1) \neq col_Q(K_2)$, then return "they are different"
return "I have no idea"

Semidecision procedure: either stops with a certificate of inequivalence, or fails to say anything valuable

Not yet clear how to make it a decision procedure.

Works well for small knots.
(experiments by Clark, Elhamdadi, Saito for knots with ≤ 13 crossings)

Unknot detection

Special case: $K_2 = \bigcirc$.

Theorem (Joyce, ..., Kuperberg)

The following are equivalent for a knot K:

(1) *K is knotted.*

(2) *There is a quandle Q such that $col_Q(K) > 0$.*

(3) *There is a finite quandle Q such that $col_Q(K) > 0$.*

(4) *There is a finite simple quandle Q such that $col_Q(K) > 0$.*

(5) *There is a conjugation quandle Q over the group $SL(2, p)$, for a prime p, such that $col_Q(K) > 0$.*

... the theorem suggests a choice of the $\mathcal{Q}$ family

... colorability of a knot is a first order problem (see blackboard)

Unknot detection algorithm

Parameter: a (potentially infinite) set of quandles $\mathcal{Q}$

IN: a knot K

two algorithms running in parallel:

run over $Q \in \mathcal{Q}$

if $col_Q(K) > 0$, then return "the knot is non-trivial"

use an automated theorem prover to prove $col_Q(K) = 0$ for every Q

Unknot detection algorithm

Parameter: a (potentially infinite) set of quandles $\mathcal{Q}$

IN: a knot K

two algorithms running in parallel:

run over $Q \in \mathcal{Q}$

if $col_Q(K) > 0$, then return "the knot is non-trivial"

use an automated theorem prover to prove $col_Q(K) = 0$ for every Q

Decision procedure: either stops with a certificate of non-triviality, or a proof of triviality is found.

Works well for moderately sized knots.
(experiments by Fish, Lisitsa, S., knots up to 100 crossings)

Unknot detection algorithm

Parameter: a (potentially infinite) set of quandles $\mathcal{Q}$

run over $Q \in \mathcal{Q}$

if $col_Q(K) > 0$, then return "the knot is non-trivial"

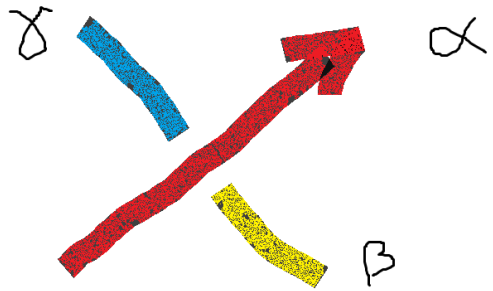
How big quandle do you need to find a coloring?

1. for random knots: usually very small ($\mathcal{Q}$ =simple quandles)
2. for specially designed knots (e.g. 2-torus): can be rather large
3. Kuperberg's certificate: under GRH, there is a "small" simple quandle Q such that $col_Q(K) > 0$ (we only knot that $|Q| = poly(|K|)$, GRH gives no concrete bound)

Crucial step: find a coloring

IN: knot K, quandle Q

OUT: $col_Q(K) > 0$?



Remember: For every crossing, we have an equation

$$c(\alpha) * c(\beta) = c(\gamma)$$

So, we are solving a system of equations over a connected quandle $(Q, *)$.

By results of Zádori, equation solving is

- P-TIME for affine connected quandles
- NP-complete otherwise

But we have a special type of equations, so ??? (in practice: fast)

Crucial step: find a coloring

IN: knot K, quandle Q

OUT: $col_Q(K) > 0$?

Brute Force: $|Q|^{|K|}$ options

Our implementation: translate to SAT, run a SAT-solver (or #SAT-solver) (see blackboard)

With an old version of MiniSat, my experimental running times were

- $\sim |Q|^3$ for 12-crossing knots
- $\sim |K|$ for torus knots and a fixed Q
- e.g., for $|Q| = 47$, $|K| = 12$, the average running time is cca 0.05 s
- we can certify the whole library of 12-crossing knots in about 5 minutes

We are quite happy with that.