Automated Reasoning for Experimental Mathematics Part I: (Un)knot Detection

Alexei Lisitsa[1]

[1] Department of Computer Science, The University of Liverpool

26 June 2019

Outline

- Part I: Automated Reasoning for Knots (computational topology)
- Part II: Solution for Erdos Discrepancy Problem, C=2 (combinatorial number theory)
- Part III: Exploration of the Andrews-Curtis Conjecture (computational combinatorial group theory)

Outline of Part I

- Preamble
- Unknot detection problem: short overview
- Involutory quandles as unknot detectors
- (Non)-trivilaity of quandles via theorem (dis-)proving
- Experimental Results
- From involutory quandles to quandles
- From theorem disproving to constraint satisfaction and SAT solving
- Very fast knot certification
- Efficiency vs Transparency

Part I is based on

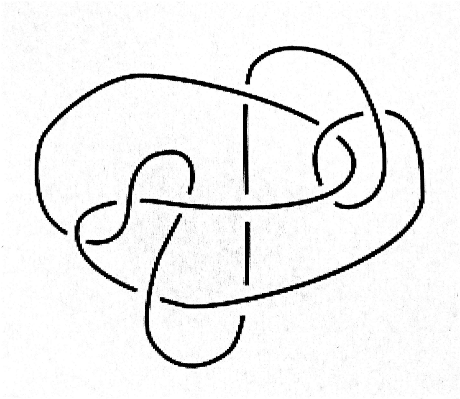
(FL 2014) Andrew Fish and Alexei Lisitsa. *Detecting unknots via equational reasoning, I: Exploration.* In International Conference on Intelligent Computer Mathematics, pages 76-91. Springer, 2014

(FLS 2015) Andrew Fish, Alexei Lisitsa, and David Stanovsky, *A combinatorial approach to knot recognition.* in Emerging Economies: First Workshop, EGC 2015, Almaty, Kazakhstan, 2015. Proceedings, pages 64-78. Springer International Publishing, 2015.

(FLSS 2016) Andrew Fish, Alexei Lisitsa, David Stanovsky, Sarah Swartwood: *Efficient Knot Discrimination via Quandle Coloring with SAT and sharp-SAT.* ICMS 2016: 51-58

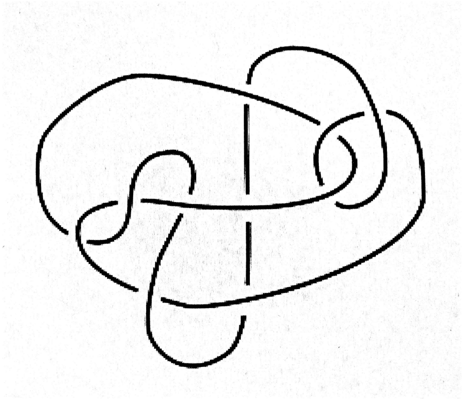
(FLV 2018) Andrew Fish, Alexei Lisitsa, and Alexei Vernitski *Visual Algebraic Proofs for Unknot Detection.* Diagrams 2018: 89-104, Springer, 2018

Question: is this a trivial knot?



Unknot detection

Question: is this a trivial knot?



Answer: Yes, it is so called *culprit* unknot

Culprit undone

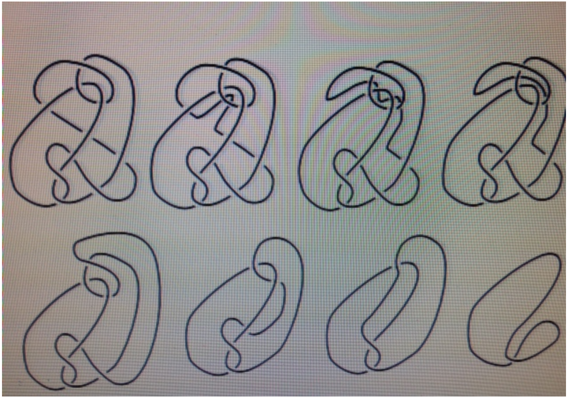
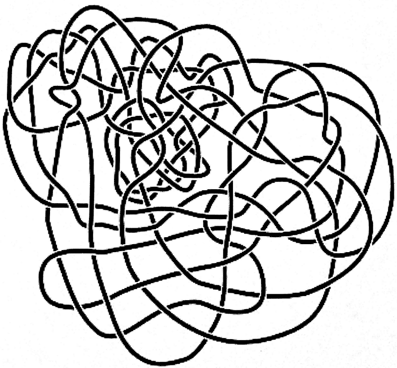


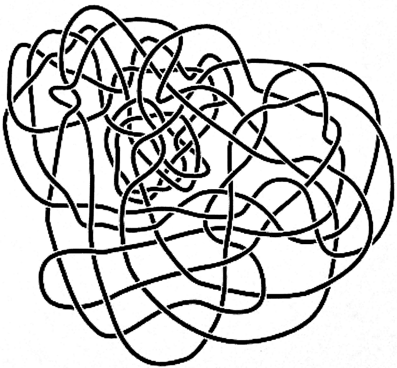
Figure: by L. H. Kauffman and S. Lambropoulou

Unknot detection

Question: is this a trivial knot?



Question: is this a trivial knot?



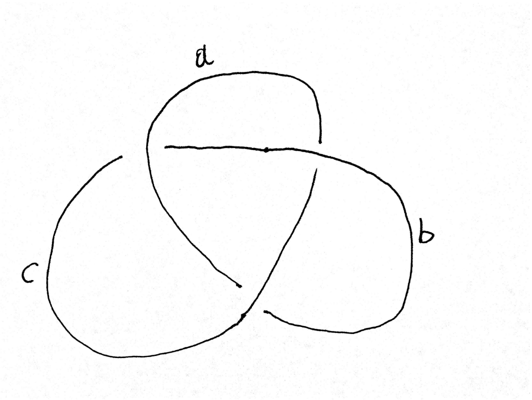
Answer: Yes, it is so called *Haken's Gordian* unknot

Haken's Gordian undone

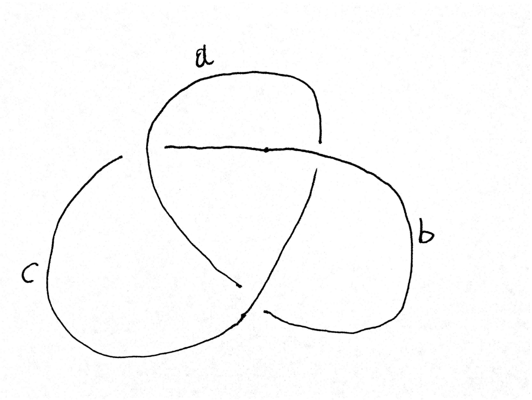
Don't even think to try it here!

Unknot detection

Question: is this a trivial knot?



Question: is this a trivial knot?



Answer: NO, it is so called *Treifoil Knot*

Unknot detection problem, I

Given: A *knot*, which is a closed loop without self-intersection embedded in 3-dimensional Euclidean space $\mathbb{R}^3$,
Question: Is it possible to deform $\mathbb{R}^3$ continuously such that the knot is transformed into a trivial unknotted circle without passing through itself?

Given: A projection of the knot on the plane,
Question: Is it possible to deform $\mathbb{R}^3$ continuously such that the knot is transformed into a trivial unknotted circle without passing through itself?

<u>**Given:**</u> A discrete code of the knot *diagram*,
<u>**Question:**</u> Is it possible to deform $\mathbb{R}^3$ continuously such that the knot is transformed into a trivial unknotted circle without passing through itself?

Given: A discrete code[1] of the knot diagram,
Question: Is it possible to deform $\mathbb{R}^3$ continuously such that the knot is transformed into a trivial unknotted circle without passing through itself?

[1] such as Gauss Code

Unknot detection: short profile

- It is decidable W. Haken (1961)

Unknot detection: short profile

- It is decidable W. Haken (1961)
- It is in coNP J. Hass,J.C. Lagarias, N. Pippenger (1999)

Unknot detection: short profile

- It is decidable W. Haken (1961)
- It is in coNP J. Hass,J.C. Lagarias, N. Pippenger (1999)
- It is in NP G. Kuperberg (2011)

Unknot detection: short profile

- It is decidable W. Haken (1961)
- It is in coNP J. Hass,J.C. Lagarias, N. Pippenger (1999)
- It is in NP G. Kuperberg (2011) (modulo GRH)

Unknot detection: short profile

- It is decidable W. Haken (1961)
- It is in coNP J. Hass,J.C. Lagarias, N. Pippenger (1999)
- It is in NP G. Kuperberg (2011) (modulo GRH)
- **Main open question:** Is it in PTIME?

Unknot detection: short profile

- It is decidable W. Haken (1961)
- It is in coNP J. Hass,J.C. Lagarias, N. Pippenger (1999)
- It is in NP G. Kuperberg (2011) (modulo GRH)
- **Main open question:** Is it in PTIME?
- We are not aiming to resolve this question

- It is decidable W. Haken (1961)
- It is in coNP J. Hass,J.C. Lagarias, N. Pippenger (1999)
- It is in NP G. Kuperberg (2011) (modulo GRH)
- **Main open question:** Is it in PTIME?
- We are not aiming to resolve this question (as yet);

Unknot detection: short profile

- It is decidable W. Haken (1961)
- It is in coNP J. Hass,J.C. Lagarias, N. Pippenger (1999)
- It is in NP G. Kuperberg (2011) (modulo GRH)
- **Main open question:** Is it in PTIME?

- We are not aiming to resolve this question (as yet);
- We are rather looking for *practically* efficient procedures.

Some algorithms for unknot detection

- An early algorithm, presented by W. Haken (1961) was deemed to be impractical due to being too complex to attempt to implement it;
- The algorithms based on *monotone simplifications* (I. Dynnikov et al, circa 2000) provide practically fast recognition of unknots but do not necessarily yield a decision procedure.
- The algorithms based on *normal surface theory*, implemented in Regina system (Burton at al, 2012) provide efficient recognition of non-trivial knots:
  - every non-trivial knot with crossing number ≤ 12 is recognized as such in under 5 minutes.

Efficiency problems

There still are efficiency problems with the existing algorithms:

- they in the worst case are exponential, and it appears that
- establishing that a particular diagram with a few hundred (or even dozens of) crossings represents a non-trivial knot may well be out of reach of the available procedures;
- Thus the exploration of alternative procedures for unknot detection is an interesting and well-justified task.

Our approach

- The unknotedness property can be faithfully characterized by the properties of algebraic invariants associated with knot projections;
- We attempt to establish the properties of concrete invariants by using methods and procedures developed in the *automated reasoning* area;
- **A key observation:** the task of unknot detection can be reduced to the task of (dis)proving a first-order formulae, and for this there are efficient generic automated procedures

Involutory quandle

Let Q be a set equipped with a binary operation $\triangleright$ (product) such that the following hold:

Q1 $x \triangleright x = x$ for all $x \in Q$.

Q2 $(x \triangleright y) \triangleright y = x$ for all $x, y \in Q$.

Q3 For all $x, y, z \in Q$, we have
$(x \triangleright y) \triangleright z = (x \triangleright z) \triangleright (y \triangleright z)$.

Then Q is called a *involutory quandle*

- The three equalities $Q1$, $Q2$ and $Q3$ form an equational theory of involutory quandles, which we denote by E_{iq}.

Involutory quandle of knot diagram

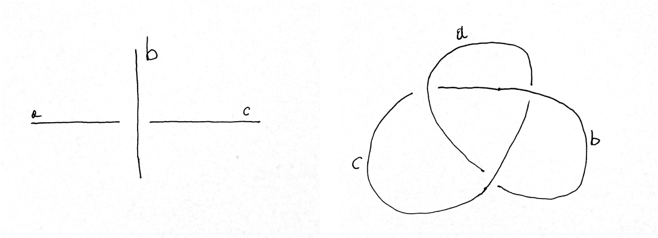
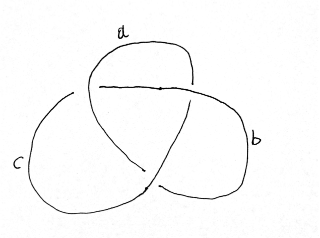


Figure: (a) Left: A labelled crossing and its corresponding relation $a \triangleright b = c$; here a and c are the labels of the underarcs at this crossing, whilst b is the label of the overarc, and we often identify the arcs with their labels to simplify language in discussions.
(b) Right: The trefoil knot diagram, with solid arcs a,b,c.

Involutory quandle of knot diagram (cont.)

Let D_{tr} be the diagram of the trefoil knot K shown below



.

The involutory quandle of D_{tr} is defined by the presentation
$IQ(D_{tr}) = \langle a, b, c \mid a \triangleright b = c, b \triangleright c = a, c \triangleright a = b \rangle$

Why IQ is good?

The importance of involutory quandles, in the context of unknot detection, relies on the following properties (Joyce1982),(Winker 1984):

- Involutory quandle is a knot invariant, i.e. it does not depend on the choice of diagram;
- Involutory quandle $IQ(K)$ of a knot K is trivial (i.e. it contains a single element e with $e * e = e$) if and only if K is the *unknot*.

IQ as unknot detector

These properties suggest the following approach to unknot detection.

- Given a knot diagram,

IQ as unknot detector

These properties suggest the following approach to unknot detection.

- Given a knot diagram, one can try to decide whether its associated involutory quandle is trivial.
- Non-trivial task:

IQ as unknot detector

These properties suggest the following approach to unknot detection.

- Given a knot diagram, one can try to decide whether its associated involutory quandle is trivial.
- Non-trivial task: an involutory quandle of a knot can be an infinite (Winker 1984).

IQ as unknot detector

These properties suggest the following approach to unknot detection.

- Given a knot diagram, one can try to decide whether its associated involutory quandle is trivial.
- Non-trivial task: an involutory quandle of a knot can be an infinite (Winker 1984).
- Not much progress has been made towards the development of specific decision procedures for such a problem, apart of that presented in the thesis of S. Winker;

IQ as unknot detector

These properties suggest the following approach to unknot detection.

- Given a knot diagram, one can try to decide whether its associated involutory quandle is trivial.
- Non-trivial task: an involutory quandle of a knot can be an infinite (Winker 1984).
- Not much progress has been made towards the development of specific decision procedures for such a problem, apart of that presented in the thesis of S. Winker;
- The diagrammatic method presented there, together with details and explanations, allows one to construct the involutory quandles for many knot diagrams,.

IQ as unknot detector via automated reasoning

In (Fish, Lisitsa 2014), we take an alternative route and propose to tackle unknot detection as follows:

- Given a knot diagram, compute its involutary quandle presentation;
- Convert the task of involutary quandle triviality detection into the task of proving a first-order equational formula;
- Concurrently, apply generic automated reasoning tools for first-order equational logic to tackle the (dis)proving task

Unknot detection by equational reasoning

- Given a knot diagram D, with n arcs, consider its involutory quandle representation $IQ(D) = \langle G_D \mid R_D \rangle$ with $G_D = \{a_1, \ldots, a_n\}$

Unknot detection by equational reasoning

- Given a knot diagram D, with n arcs, consider its involutory quandle representation $IQ(D) = \langle G_D \mid R_D \rangle$ with $G_D = \{a_1, \ldots, a_n\}$
- Denote by $E_{iq}(D)$ an equational theory of $IQ(D)$, i.e. $E_{iq}(D) = E_{iq} \cup R_D$.

Unknot detection by equational reasoning

- Given a knot diagram D, with n arcs, consider its involutory quandle representation $IQ(D) = \langle G_D \mid R_D \rangle$ with $G_D = \{a_1, \ldots, a_n\}$
- Denote by $E_{iq}(D)$ an equational theory of $IQ(D)$, i.e. $E_{iq}(D) = E_{iq} \cup R_D$.

Proposition

A knot diagram D is a diagram of the unknot if and only if $E_{iq}(D) \vdash \wedge_{i=1\ldots n-1}(a_i = a_{i+1})$, where $\vdash$ denotes derivability in the equational logic (or, equivalently in the first-order logic with equality).

Unknot detection by equational reasoning (cont.)

So, the unknot detection procedure P which we propose here consists of the parallel composition of

- automated proving $E_{iq}(D) \to \wedge_{i=1\dots n-1}(a_i = a_{i+1})$, and
- automated disproving $E_{iq}(D) \to \wedge_{i=1\dots n-1}(a_i = a_{i+1})$ by a finite model finder.

It is obvious that the parallel composition above provides with *at least* a semi-decision algorithm for unknotedeness.

Unknot detection by equational reasoning (cont.)

So, the unknot detection procedure P which we propose here consists of the parallel composition of

- automated proving $E_{iq}(D) \rightarrow \wedge_{i=1\ldots n-1}(a_i = a_{i+1})$, and
- automated disproving $E_{iq}(D) \rightarrow \wedge_{i=1\ldots n-1}(a_i = a_{i+1})$ by a finite model finder.

It is obvious that the parallel composition above provides with *at least* a semi-decision algorithm for unknotedeness.
Is it decision procedure?

Unknot detection by equational reasoning (cont.)

So, the unknot detection procedure P which we propose here consists of the parallel composition of

- automated proving $E_{iq}(D) \rightarrow \wedge_{i=1...n-1}(a_i = a_{i+1})$, and
- automated disproving $E_{iq}(D) \rightarrow \wedge_{i=1...n-1}(a_i = a_{i+1})$ by a finite model finder.

It is obvious that the parallel composition above provides with *at least* a semi-decision algorithm for unknotedeness.
Is it decision procedure?
We don't know . . . It would be if the following conjecture holds

Finite residuality conjecture

Conjecture (Involutory quandles are finitely residual)

For any knot diagram D, if $IQ(D)$ is not trivial (i.e. consists of more than 1 element), then there is a finite non-trivial involutory quandle Q which is a homomorphic image of $IQ(D)$.

Example, I

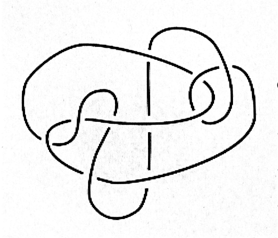


Figure: Culprit Unknot

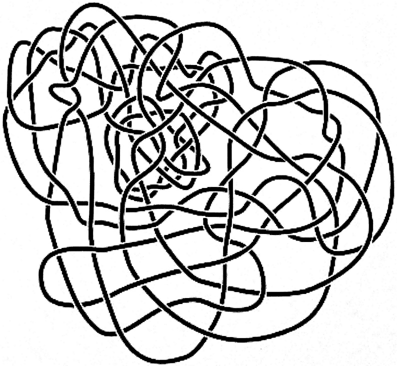
Assumptions:

```
%Involutory quandle axioms
x * x = x.
(x * y) * y = x.
(x * z) * (y * z) = (x * y) * z.
%Culprit unknot
a1 = a9 * a7.
a3 = a1 * a2.
a2 = a3 * a4.
a5 = a2 * a10.
a6 = a5 * a4.
a7 = a6 * a1.
a8 = a7 * a4.
a10 = a8 * a9.
a4 = a10 * a3.
a9 = a4 * a8.
```

Goals:

```
(a1 = a2) & (a2 = a3)
(a3 = a4) & (a4 = a5)
(a5 = a6) & (a6 = a7)
(a7 = a8) & (a8 = a9)
(a9 = a10).
```

Example, II: Haken's unknot



(See demonstration)

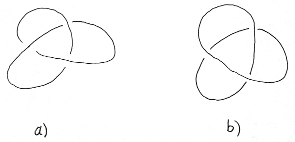
Some comparisons on unknot certification

- The only alternative approach capable of detecting unknotedness of Haken's Gordian Unknot in practice, that we are aware of, is Dynnikov's algorithm based on *monotone simplifications* (under a second);
- We have experimented also with the detection of other well-known hard unknots, such as
  - Goerlitz unknot,
  - Thistlethwaite unknot,
  - Friedman's Twisted unknot, etc;

Some comparisons on unknot certification

- The only alternative approach capable of detecting unknotedness of Haken's Gordian Unknot in practice, that we are aware of, is Dynnikov's algorithm based on *monotone simplifications* (under a second);
- We have experimented also with the detection of other well-known hard unknots, such as
  - Goerlitz unknot,
  - Thistlethwaite unknot,
  - Friedman's Twisted unknot, etc;
  - All detected in a less than a second
- The largest tried unknot we can detect had 339 crossings (Dynnikov's example, in 40s)
- The smallest tried unknot we can not detect had 407 crosssings (Dynnikov' example, $\geq 40000s$)

Example, III: Trefoil knot a)



The countermodel found by Mace4 is:

```
interpretation( 3, [number=1, seconds=0], [
        function(a1, [ 0 ]),
        function(a2, [ 1 ]),
        function(a3, [ 2 ]),
        function(*(_,_), [
   0, 2, 1,
   2, 1, 0,
   1, 0, 2 ])
]).
```

Detecting non-trivial knots

- We have experimented with the detection of all prime knots up to 10 crossings using Mace4 model finder;

Detecting non-trivial knots

- We have experimented with the detection of all prime knots up to 10 crossings using Mace4 model finder;
- The data presented in a table (separate document) include the standard code of the knot, size of minimal countermodel found and time taken;

Detecting non-trivial knots

- We have experimented with the detection of all prime knots up to 10 crossings using Mace4 model finder;
- The data presented in a table (separate document) include the standard code of the knot, size of minimal countermodel found and time taken;
- For the five special cases $10_{83}, 10_{91}, 10_{92}, 10_{117}, 10_{119}$ our approach did not terminate in a reasonable time ($\geq 20000s$);

Comparison with Regina tool algorithm

- Regina tool algorithm (Burton et al 2012) is the most efficient algorithm for non-triviality of knots certification (all knots up to 12 crossings can be certified in under 5min each, up to 10 crossings in under 3min each);
- How do we fare against Regina (up to 10 crossings)?

Comparison with Regina tool algorithm

- Regina tool algorithm (Burton et al 2012) is the most efficient algorithm for non-triviality of knots certification (all knots up to 12 crossings can be certified in under 5min each, up to 10 crossings in under 3min each);
- How do we fare against Regina (up to 10 crossings)?
  - Average time Regina: 47s
  - Average time Mace4: 1230s (ignoring 5 failed cases);

Comparison with Regina tool algorithm

- Regina tool algorithm (Burton et al 2012) is the most efficient algorithm for non-triviality of knots certification (all knots up to 12 crossings can be certified in under 5min each, up to 10 crossings in under 3min each);
- How do we fare against Regina (up to 10 crossings)?
  - Average time Regina: 47s
  - Average time Mace4: 1230s (ignoring 5 failed cases);

but

- In general our approach demonstrates much higher discrepancy in timing data:
  - For countermodels sizes up to 15-17 the detection time is under a second – that holds in more than 70% of instances, where our approach outperforms Regina's algorithm;
  - In a few cases with large countermodels (e.g $10_{88}, 10_{94}, 10_{115}$) it takes 40000-80000s to complete the search.

Let us try to do some exercises

See provided files with the encoding of some non-trivial knots.
Also have a look at *https://www.indiana.edu/ knotinfo/*

Further developments

- Assymetric approach: *prove for involutory quandles, disprove for quandles* (disproving using quandles is much faster than using inv. quandles)
- Quandle coloring, constraint satisfaction and SAT-solving

Quandles

Definition

A set Q equipped with a binary operation $\triangleright$ is called a *quandle* if the following conditions hold:

- Q1 $x \triangleright x = x$ for all $x \in Q$.
- Q2 For all $x, y \in Q$, there is a unique $z \in Q$ such that $x = z \triangleright y$.
- Q2' $(x \triangleright y) \triangleright y = x$ for all $x, y \in Q$.
- Q3 For all $x, y, z \in Q$, we have $(x \triangleright y) \triangleright z = (x \triangleright z) \triangleright (y \triangleright z)$.

Then Q is called a *quandle* if Q satisfies $Q1$, $Q2$ and $Q3$, and an *involutory quandle* if Q satisfies $Q1$, $Q2'$ and $Q3$.

Knot coloring by quandles

- Let D be a knot diagram and Q a quandle. A *coloring* is a mapping c assigning to every arc a color from Q in a way that for every crossing with arcs labeled α, β, γ as below, $c(\gamma) = c(\beta) \triangleright c(\alpha)$ holds.
- Let $col_Q(D)$ denote the number of non-trivial (more than one color) colorings of D by Q, then $col_Q(D)$ is knot invariant

Main theorem

Theorem

The following are equivalent for a knot K:

(*) *K is knotted (i.e., not equivalent to the unknot).*

(U1) *$Q(K)$ is non-trivial.*

(U2) *$IQ(K)$ is non-trivial.*

(K1) *There is a finite quandle Q such that $col_Q(K) > 0$.*

(K2) *There is a finite simple quandle Q such that $col_Q(K) > 0$.*

(K3) *There is a conjugation quandle Q over the group $SL(2, p)$, for some prime p, such that $col_Q(K) > 0$.*

By combination Joyce 1982, Matveev 1984, Winker 1984, Clark, Satio and Vendramin, 2014 , Kuperberg 2014

Semi-algorithm for knot detection

- Given a knot K and some (pre-computed) family of quandles $\mathcal{Q}$. Iterate over $\mathcal{Q}$ and check whether K is colorable by some quandle from $\mathcal{Q}$.

Semi-algorithm for knot detection

- Given a knot K and some (pre-computed) family of quandles $\mathcal{Q}$. Iterate over $\mathcal{Q}$ and check whether K is colorable by some quandle from $\mathcal{Q}$.
- The check can be reduced to a task which can be handled by automated reasoning methods: *constraint satisfacton*, or *SAT-solving*

Semi-algorithm for knot detection

- Given a knot K and some (pre-computed) family of quandles $\mathcal{Q}$. Iterate over $\mathcal{Q}$ and check whether K is colorable by some quandle from $\mathcal{Q}$.
- The check can be reduced to a task which can be handled by automated reasoning methods: *constraint satisfacton*, or *SAT-solving*
- We have experimented with various variants, including serial and parallel constraint solving, Prolog search mechanisms, SAT-solving, etc

Semi-algorithm for knot detection

- Given a knot K and some (pre-computed) family of quandles $\mathcal{Q}$. Iterate over $\mathcal{Q}$ and check whether K is colorable by some quandle from $\mathcal{Q}$.
- The check can be reduced to a task which can be handled by automated reasoning methods: *constraint satisfacton*, or *SAT-solving*
- We have experimented with various variants, including serial and parallel constraint solving, Prolog search mechanisms, SAT-solving, etc
- SAT solving is an absolute winner!

Computational Problems

Given a quandle Q, and a knot diagram D, one formulates the following problems:

Q-colorability. Is ${}_Q(D) > 0$, i.e., is there a non-trivial Q-coloring of D?

Q-coloring number. Compute ${}_Q(D)$, the number of non-trivial Q-colorings of D.

SAT vs #SAT problems

- **SAT:** Given a propositional formula, is there a satisfying assignment?
- **#SAT:** Given a propositional formula, find a number of satisfying assignments.

SAT vs #SAT problems

- **SAT:** Given a propositional formula, is there a satisfying assignment?
- **#SAT:** Given a propositional formula, find a number of satisfying assignments.

For both problems there are efficient solvers.

Quandle colorability via SAT (FLS2015)

Fix a quandle $Q = (\{1, \ldots, q\}, \triangleright)$ and a knot diagram D with $|D| = n$, with arcs numbered $\alpha_1, \ldots, \alpha_n$.
We consider boolean variables $v_{i,c}$ that determine whether the arc α_i has the color c.
We need to satisfy the following constraints:

- Every arc has a unique color: the obvious description uses the clauses

  $$v_{i,1} \vee \ldots \vee v_{i,q} \quad \text{and} \quad \neg v_{i,c} \vee \neg v_{i,d}$$

  for every $i = 1, \ldots, n$ and $c = 1, \ldots, q$, $d = c + 1, \ldots, q$.
- Not all arcs have the same color: the obvious description uses, for every $c = 1, \ldots, q$, the clause

  $$\neg v_{1,c} \vee \ldots \vee \neg v_{n,c}.$$

- For every crossing we use formulas of the form

  $$(v_{i,c} \wedge v_{j,d}) \to v_{k,d \triangleright c}.$$

Experimental Setup

For our experiments, the following families of quandles and knots were used:

- **SQ**. all 354 simple quandles of size ≤ 47, indexed in accordance to size.
- **CQ**. 26 quandles (each of size ≤ 182).
- **Q1-Q3**. small sets of quandles used for knot recognition (with #-SAT).
- **K10-K13** all 249, 801, 2977 and 12965 prime knots (up to reverse and mirror image) with crossing numbers not exceeding 10,11,12 and 13 respectively.
- **T3**. $(3, n)$-torus knots with $n = 6k + 2$.
- **R**. 52 randomly generated large knots
- **A13**. all 34659 alternating minimal projections of prime knots with crossing numbers not exceeding 13.

Software

- MiniSat 2.2.0
- #-SAT 12.08
- Perl/Prolog scripts
- Debian Linux VM, hosted on Windows 7 system

SAT solving for fast knot detection

- Given a PD code of a knot, a procedure
  - iterates over all quandles from **SQ**,
  - converts quandle colorability task into SAT instance,
  - check satsifiability with MiniSat,
  - proceeds until the first satsifiable case is found.

- Given a PD code of a knot, a procedure
  - iterates over all quandles from **SQ**,
  - converts quandle colorability task into SAT instance,
  - check satsifiability with MiniSat,
  - proceeds until the first satsifiable case is found.
- When a satsifiable case is found, this is a solution to the Q-colorability problem, giving witness to the non-triviality of the knot.

SAT solving for fast knot detection

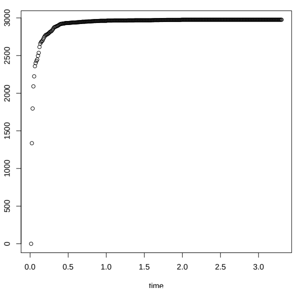


Figure: Cumulative frequency of running times (s) for the **K12** family.

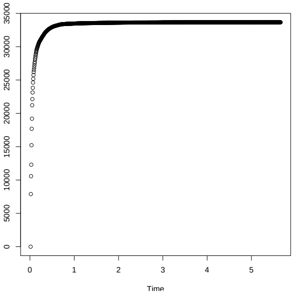


Figure: Cumulative frequency of running times (s) for the **A13** family.

- **K12** family, SAT solving: the detection time for each case is in the interval 0.013–3.31s
- **K12** family, Regina's algorithm B. Burton & M. Özlen,(2012): the detection time for each case is in under 5 minutes.

Visual Proofs based on tangles

Tangles are essentially knots but with free ends possible

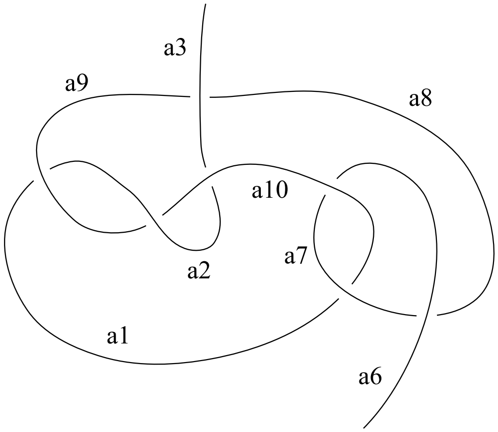


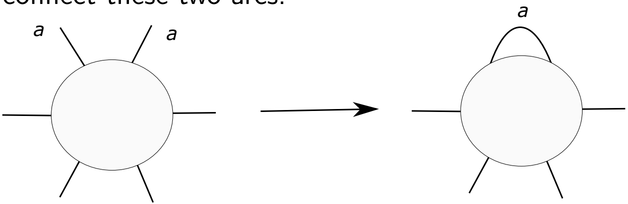
Figure: A tangle (disconnected Culprit)

Theorem (FLV18)

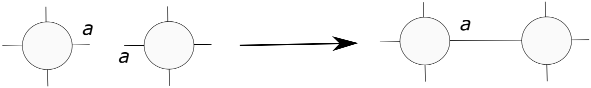
A knot diagram D (with unique labels) represents the unknot if and only if for each pair of its labels a, b a labeled tangle diagram T which has exactly 2 free-end arcs labelled a and b, can be build from the elementary tangles (corresponding to the original crossings) using the tangle building rules.

Tangle building rules (FLV18)

1. Given a labelled tangle diagram which has, amongst its end arcs, two adjacent end arcs labelled with the same letter, connect these two arcs.



2. Given two labelled tangle diagrams T and U such that both T and U have an end arc labelled with the same letter, connect these two arcs.



Now the tangle building procedure can be delegated again to the automated theorem proving procedure, giving yet another way for proving unknotedness.

Theorem Proving for Tangle Building

Now the tangle building procedure can be delegated again to the automated theorem proving procedure, giving yet another way for proving unknotedness.

Try practical example *culprit-tangle*

Conclusion

- Unknot detection via theorem (dis)proving is viable and interesting;

Conclusion

- Unknot detection via theorem (dis)proving is viable and interesting;
- There is a variety ways in which this can be done;

Conclusion

- Unknot detection via theorem (dis)proving is viable and interesting;
- There is a variety ways in which this can be done;
- Knot detection via SAT solving is practically fastest known procedure;

Conclusion

- Unknot detection via theorem (dis)proving is viable and interesting;
- There is a variety ways in which this can be done;
- Knot detection via SAT solving is practically fastest known procedure;
- Further analysis, both empirical and theoretical is required;

Conclusion

- Unknot detection via theorem (dis)proving is viable and interesting;
- There is a variety ways in which this can be done;
- Knot detection via SAT solving is practically fastest known procedure;
- Further analysis, both empirical and theoretical is required;
- Applications to biology: detection of knotted fragments of DNA and proteins

Conclusion

- Unknot detection via theorem (dis)proving is viable and interesting;
- There is a variety ways in which this can be done;
- Knot detection via SAT solving is practically fastest known procedure;
- Further analysis, both empirical and theoretical is required;
- Applications to biology: detection of knotted fragments of DNA and proteins

Thank you for you attention!