Classical Link Recognition is in NP

(joint work with Kazuhiro Ichihara and Seiichi Tani)

Yuya Nishimura
Topology and Computer 2020
September 17th, 2020

Graduate School of Integrated Bases Science, Nihon University
A decision problem is in $\text{P} \iff$ The problem can be solved in polynomial time by a deterministic Turing machine.

A decision problem is in $\text{NP} \iff$ The problem can be solved in polynomial time by a non-deterministic Turing machine.
1. Virtual link
2. Preparation
3. Main result
1. Virtual link
Virtual link diagram

Definition 1 (Virtual link diagram)

A virtual link diagram is a 4-regular plane graph with over/under or virtual information at each vertex.
Definition 2 (Virtual link)

A virtual link is an equivalence class of virtual link diagrams under the $R$-moves and the virtual $R$-moves.
Example of virtual links

virtual trefoil  virtual Hopf link  trefoil

All classical links are also virtual links.
Oriented Gauss code

Encode a diagram with oriented Gauss code\(^1\).

1. Assign natural numbers to all real crossings.
2. For each component, choose an orientation and a starting point.
3. Choose a starting point, and travel along the diagram.

\[\begin{align*}
\rightarrow 1 & \quad +\rightarrow 2 \quad ; \\
+<1 & \quad -<2 \quad +<3 \\
+<4 & \quad \rightarrow 3 \quad \rightarrow 4
\end{align*}\]

\(^1\)http://www.javaview.de/services/knots/doc/description.html
4. Each time you come through a real crossing, write three symbols.
   - Write "+" if you go over the crossing, otherwise write "−".
   - Write ">" if you see the arc you intersect going from left to right, otherwise write "<".
   - Write index number of the crossing.

5. Write ";" as a components separator.

A virtual link diagram can be encoded with length $O(c)$, where $c$ is the real crossing number.
Problem 1 (Classical link recognition)

Let $L$ be a virtual link, and $D$ be its diagram.

**input:** virtual link diagram $D$

**output:**

\[
\begin{cases}
\text{yes} \cdots & \text{if } L \text{ is a classical link} \\
\text{no} \cdots & \text{otherwise}
\end{cases}
\]

Theorem 1 (Kauffman and Manturov\textsuperscript{2}, 2006)

Classical link recognition is computable.

\textsuperscript{2}Virtual Knots and Links, Proceedings of the Steklov Institute of Mathematics (2006), 252, pp. 104—121
Main result

Main result 1
Classical link recognition is in NP.

Main result 2
We give an algorithm to solve classical link recognition, and implement the algorithm using Regina.
2. Preparation
Canonical surface realization of diagram $D$

- Abstract link diagram of $D$
- Canonical surface realization of $D$
- Attach disks to $\partial N(\tilde{D})$
Canonical space realization of diagram $D$

canonical surface realization of $D$  
$(\tilde{D}, F)$

canonical space realization of $D$  
$(\hat{D}, F \times I)$

canonical complement of $D$  
$F \times I - N(\hat{D})$
Splitting and Destabilization

We define the following two operations on a canonical complement.

1. Split link components by $\mathbb{S}^2$, and attach 3-balls.

2. Destabilize by vertical ess. ann.
Minimal complement

\[ \text{canonical comp.} \quad M_0 \rightarrow \cdots \rightarrow M_i \rightarrow M_{i+1} \rightarrow \cdots \rightarrow M_k \]

\[ \text{splitting or destabilization} \]

\[ F_0 \times I - N(\hat{D}) \]

\[ F_k \times I - N(\hat{D}') \]

**Theorem 2 (Kuperberg\textsuperscript{3}, 2003)**

*Virtual link* $L$ *has a unique minimal complement (up to isotopy).*

**Lemma 1**

*Virtual link* $L$ *is classical* \iff \( g(F_k) = 0 \)

---

\textsuperscript{3}What is a virtual link?, Algebraic & Geometric Topology 3 (2003), pp.587—591
We perform the above operation on a triangulation.

1. Construct a triangulation of the canonical complement of \( D \).
2. Find an essential sphere or an essential annulus in the triangulation.
3. Perform the operations to construct the minimal complement.
# Triangulation of the canonical complement

1. Construct a triangulation of the canonical complement of $D$.

## Lemma 2

**Input:** virtual link diagram $D$ with $c$ real crossings

We can construct a triangulation of the canonical complement of $D$ with $O(c)$ tetrahedra in time $O(c)$. 

![Diagram](image)
2. Find an essential sphere or an essential annulus in the triangulation.

**Definition 3 (Normal surface)**

$\mathcal{T}$: 3-manifold triangulation with $n$ tetrahedra

A *normal surface* in $\mathcal{T}$ is a proper embedded surface meeting each tetrahedron in a collection of disjoint normal disks.

7 type normal disks in a tetrahedron
Normal surface

A normal surface \( S \) in \( \mathcal{T} \) can be represented as a vector in \( \mathbb{R}^{7n} \) by counting the number of disks of each type for each tetrahedron.

\[
x(S) = (x_{1,1}^\triangle, x_{1,2}^\triangle, x_{1,3}^\triangle, x_{1,4}^\triangle, x_{1,1}^\square, x_{1,2}^\square, x_{1,3}^\square x_{2,1}^\triangle, \cdots, x_{n,3}^\square) \in \mathbb{R}^{7n}
\]

\( S_1 + S_2 := \) the normal surface whose vector representation is

\[
x(S_1) + x(S_2)
\]

**Definition 4 (Vertex surface)**

A normal surface \( S \) is called a **vertex surface** if \( S \) is 2-sided & the equation \( kS = lS_1 + mS_2 \) cannot be satisfied for any positive integers \( k, l \) and \( m \) and any non-empty surfaces \( S_1 \) and \( S_2 \).
Find essential surfaces

**Theorem 3 (Jaco and Tollefson\textsuperscript{4}, 1995)**

$\mathcal{T}$: triangulation of a 3-manifold

If there is an essential sphere $S \subset \mathcal{T}$, then there is a vertex surface $F$ in $\mathcal{T}$ isotopic to $S$.

**Theorem 4 (Tollefson\textsuperscript{5}, 1998)**

$\mathcal{T}$: triangulation of a irr. $\partial$-irr. 3-manifold

If there is a 2-sided imcomp. $\partial$-imcomp. surface $S \subset \mathcal{T}$, then there is a vertex surface $F$ isotopic to $S$.


\textsuperscript{5}NORMAL SURFACE Q-THEORY, pacific journal of mathematics \textbf{183}(1998), No. 2, pp.359—374
3. Perform the operations to construct the minimal complement.

$\mathcal{T}$: triangulation of the canonical complement with $n$ tetrahedra.

**Lemma 3 (Splitting)**

- *Splitting can be carried out in time $O(n)$.*
- *Splitting does not increase the number of tetrahedra.*

**Lemma 4 (Destabilization)**

- *Destabilization can be carried out in time $O(n^3)$.*
- *Destabilization increases the number of tetrahedra by $O(c)$.*
3. Main result
A decision problem is in NP.

\[ \iff \text{The problem can be solved in polynomial time by a non-deterministic Turing machine.} \]

\[ \iff \text{Let } D \text{ be an instance which answer is "yes".} \]

\[ \text{instance: } D \rightarrow \text{DTM} \rightarrow \text{Is the answer "yes"?} \]

\[ \text{witness: } w \]

\[ \text{in polynomial time} \]
Outline of proof

**Main theorem 1**

Classical link recognition is in NP.

$L$: classical link

$D$: virtual link diagram of $L$ with $c$ crossings

\[ \begin{align*}
\cdots & \rightarrow \mathcal{T}_0 \rightarrow \mathcal{T}_1 \rightarrow \mathcal{T}_{i+1} \rightarrow \cdots \rightarrow \mathcal{T}_{\mathcal{O}(c)} \\
\text{split or destabilize} & \\
F \times I - N(\hat{D}) & \rightarrow \cdots \\
\text{canonical comp.} & \rightarrow \ \\
S^2 \times I - N(\hat{D}') & \rightarrow \text{minimal comp.}
\end{align*} \]

We give the vertex surfaces as a witness.

It can be verified that $L$ is classical by splitting and destabilization $\mathcal{O}(c)$ times.
Outline of proof

Splitting and destabilization can be carried out in polynomial time. 
⇒ We can verify that $L$ is classical in polynomial time. 
⇒ The problem is in NP. □

A decision problem is in NP
⇒ There is an algorithm which runs in exponential time.

We give a concrete algorithm to solve classical link recognition.
Algorithm for classical link recognition

input: virtual link diagram $D$

Construct $F \times I - N(\hat{D})$

$g(F) = 0$?

"YES" \hspace{2cm} "NO"

Enumerate vertex surfaces

$\exists$ ess. sphere?

Split

$\exists$ vertical ess. annulus?

Destabilize

"NO"
Algorithm for classical link recognition

Theorem 5 (Burton\textsuperscript{6}, 2010)

\(\mathcal{T}:\) triangulation with \(n\) tetrahedra

Let \(\phi = \frac{1+\sqrt{5}}{2}\). The number of vertex surfaces in \(\mathcal{T}\) is bounded above by \(\mathcal{O}(\phi^{7n}) \simeq \mathcal{O}(29.03^n)\)

\(\Rightarrow\) It takes exponential time to execute the algorithm.

\textsuperscript{6}The complexity of the normal surface solution space, SCG ’10: Proceedings of the Twenty-Sixth Annual Symposium on Computational Geometry (2010), pp. 201—209.
Implementation

It takes exponential time to execute the algorithm in theory. How long does it take to execute the algorithm in practice?

⇒ Implement the algorithm, and measure the execution times.

Table 1: Environment

<table>
<thead>
<tr>
<th>Language</th>
<th>C++14</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Ubuntu 18.04.3 LTS</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel Core i5 1.60GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>4GB</td>
</tr>
</tbody>
</table>
We implement the algorithm using Regina\textsuperscript{7,8}. Regina is a software package developed by B. Burton, with a focus on triangulations, normal surfaces and angle structures.

\textsuperscript{7}https://regina-normal.github.io

\textsuperscript{8}Introducing Regina, the 3-manifold topology software, Experimental Mathematics \textbf{13} (2004), pp. 267–272.
We use a table of virtual knot\(^9\) made by J. Green as an input data.

Example of input

\[
+>1 \quad +>2 \quad -<1 \quad -<2
\]

\(^9\)https://www.math.toronto.edu/drorbn/Students/GreenJ/index.html
Experimental result

We measured the execution times of the algorithm. The size of triangulation is heuristically reduced after each operation.
Experimental result

The bottleneck is an enumeration of vertex surfaces.

\[ \#\text{vertex surface} \in O(29.03^n) \]

where \( n \) = triangulation size

The triangulation size is essentially involved in execution time.
Summary

- We showed that classical link recognition is in NP.
- We gave and implemented the algorithm.

Future works

- Is classical link recognition in co-NP?
- Can we construct a fast algorithm for the problem?