木幅を固定した
結び目の自明性判定問題

西村 勇哉

卒業研究・卒業演習成果発表会
2019 年 1 月 31 日

日本大学文理学部情報科学科　谷研究室

目次

結び目理論

結び目とは

定義 1.1 (結び目)

３次元空間の中で自己交差しないようにひもを絡めてそのひもの両端をつないだもの

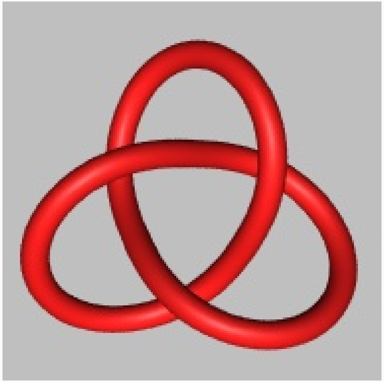


Figure 1: 三葉結び目

KnotPlot(`https://knotplot.com/`) より引用

結び目の例

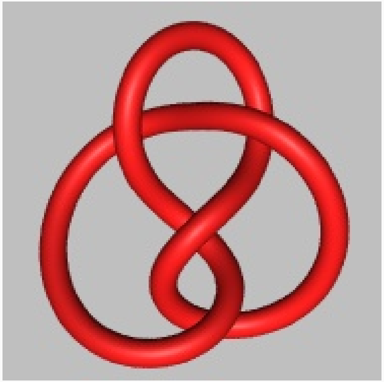


Figure 2: 8 の字結び目

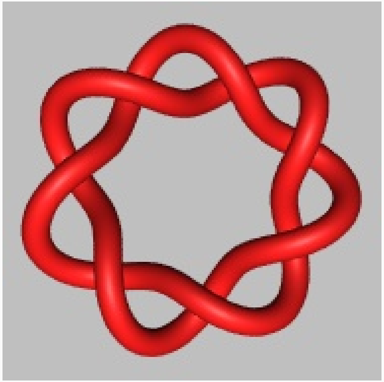


Figure 3: 7_1

KnotPlot(https://knotplot.com/) より引用

結び目の応用

例えば，DNA は右図のように結び目をなすことがあり，その解析には結び目理論が応用されている

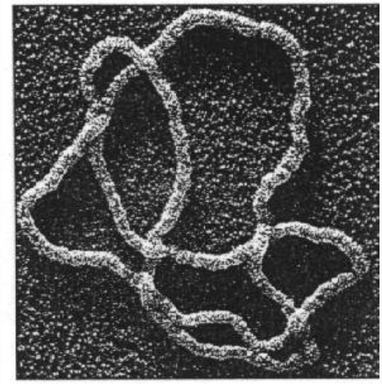
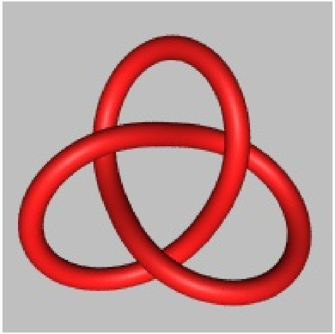


Figure 4: 結ばれた環状 DNA[9]

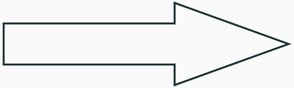
出典:De Witt Sumners,*Lifting the curtain: using topology to probe the hidden action of enzymes*, Notices of the American Mathematical Society vol.42(5),(1995),528-537.

結び目の図式

結び目を $\mathbb{R}^2$(または S^2) に射影して，上下の情報を持たせたものを図式という



射影+上下の情報

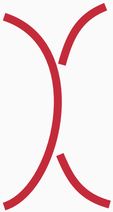


結び目の図式

図式の交点は横断的にのみ交わる二重点になるように射影

OK

NG(横断的でない)



NG(三重点)

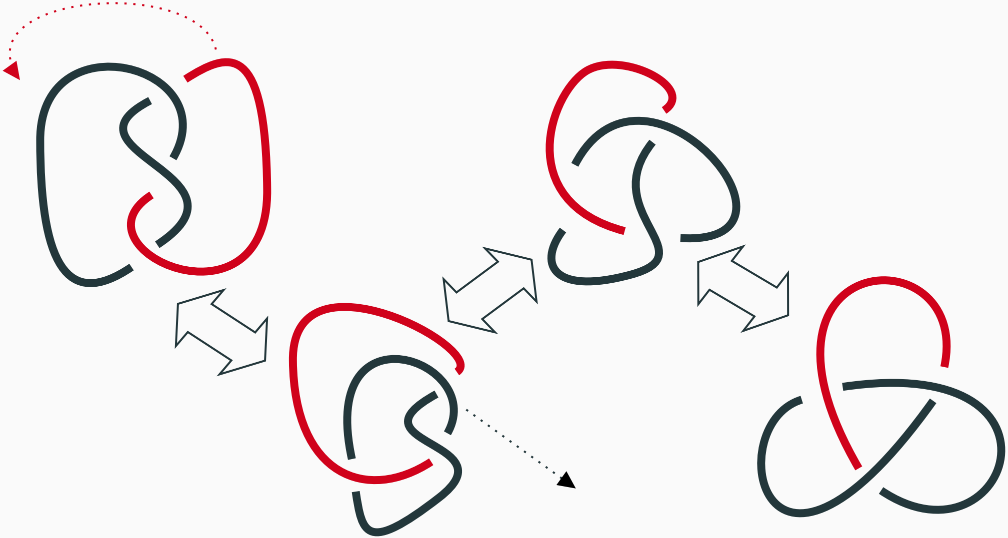
結び目の同値性

定義 1.2 (同型な結び目)

K, K' を結び目とする．K, K' を切ったり，貼り付けたり，自己交差することなく変形させて互いに同じ形にできる時 K, K' を同型な結び目といい $K \simeq K'$ と表す

(厳密にはアンビエントイソトピーという連続写像を使って定義される)

同型な結び目 (具体例)



自明結び目

定義 1.3 (自明結び目)

交点のないループを図式としてもつ結び目と同型な結び目を自明な結び目という

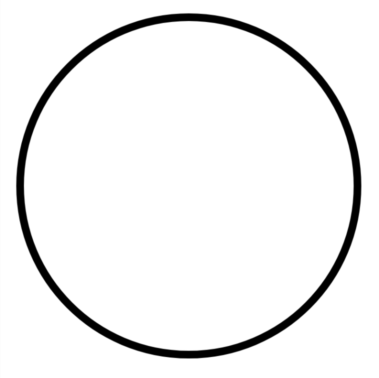
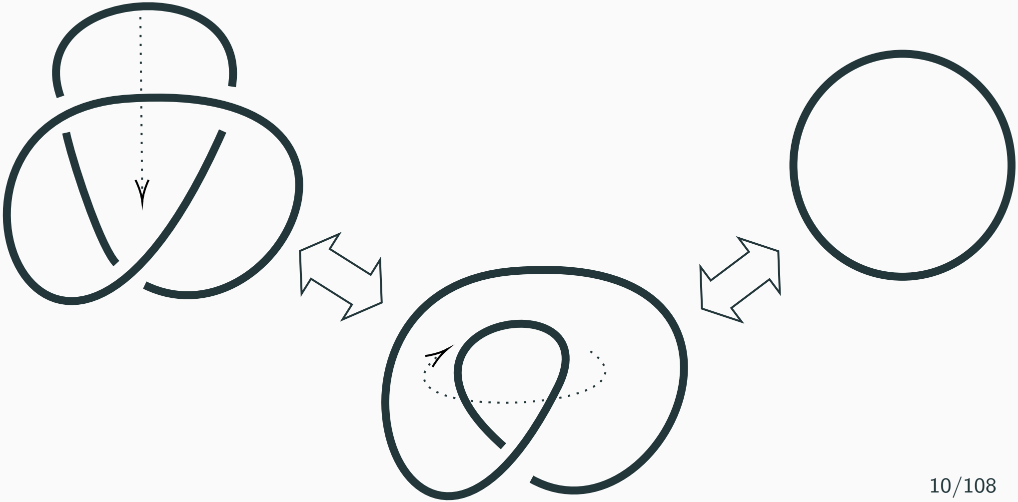


Figure 5: 自明な図式

自明結び目の図式の例



自明性判定問題

定義 1.4 (自明性判定問題)

入力:結び目の図式 D

出力:D が自明結び目の図式ならば*'yes'*，それ以外なら*'no'*

自明性判定問題

自明な結び目の図式の中には,
複雑で見た目での判断が難しい
ものも存在
→ 計算機を用いて判定
→ 図式をどうやって入力する?

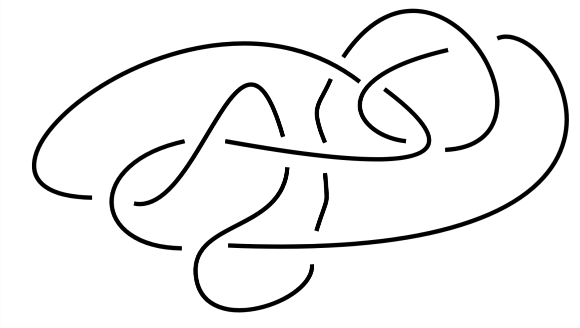


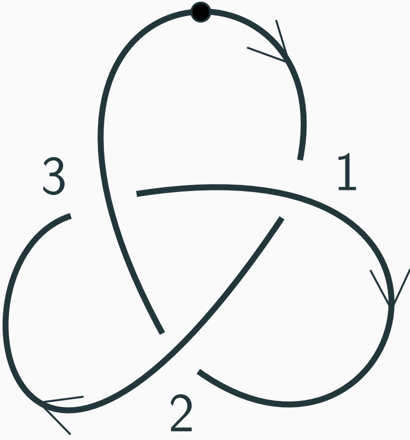
Figure 6: 複雑な自明結び目の図式

出典：Louis H. Kauffman, Sofia Lambropoulou, *Hard Unknots and Collapsing Tangles*, Introductory Lectures on Knot Theory (2011),187-247.

ガウスコード表示の例

定義 1.5 (ガウスコード表示)

1. 図式 D の全ての交点に番号をふる
2. D の交点以外の点に基点をとり，基点から結び目をたどる
3. 交点 i に到達した時に，
   - 下を通るなら $-i$ を並べる
   - 上を通るなら i を並べる



(-1,2,-3,1,-2,3)

結び目の符号化

ガウスコード表示は結び目の図式を一意に表す

ガウスコード表示は交点数 n に対して $\mathcal{O}(n)$ の長さ

$\Rightarrow$ 入力のサイズは交点数で測る

先行研究

自明性判定問題について以下のことが知られている

- NP に属する (Hass-Lagarias-Pippenger 1999)
- co-NP に属すると予想される (Kuperberg 2014)

$\Rightarrow$ 自明性判定問題は NP 完全ではないと予想される

先行研究

本演習で参考にしたアルゴリズムの計算量：$\mathcal{O}(3^n)$
(Burton-Ozlen　2012)

Burton-Ozlen のアルゴリズムの最悪計算量は $\mathcal{O}(3^n)$ だが，実際には多項式時間に近い振る舞い

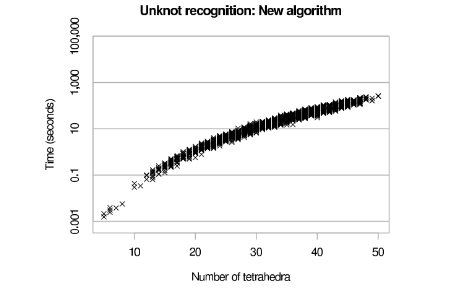


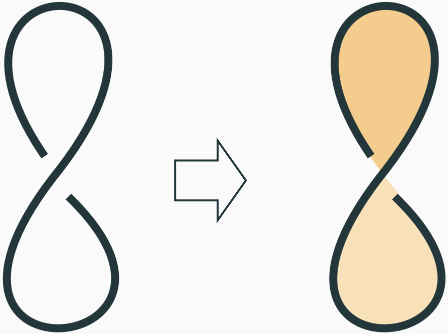
Figure 7: Burton-Ozren のアルゴリズムの実行時間

出典：Benjamin A. Burton, Melih Ozlen, *A FAST BRANCHING ALGORITHM FOR UNKNOT RECOGNITION WITH EXPERIMENTAL POLYNOMIAL-TIME BEHAVIOUR*, Preprint, **https://arxiv.org/abs/1211.1079** (2012)

自明性判定問題へのアプローチ

定理 1.1

結び目 K が自明 $\Leftrightarrow$ K を境界とする円盤 D が存在



Burton-Ozlen のアルゴリズムではこの円盤を見つけることで自明性を判定

自明性判定問題へのアプローチ

本演習では，Burton-Ozlen のアルゴリズムを基礎に結び目補空間の単体分割の face pairing グラフの木幅を固定した自明性判定問題に対して動的計画法を用いたアルゴリズムを構成

結び目補空間

3 次元球面

定義 2.1 (3 次元球面)

$$S^3 = \{(x, y, z, w) \in \mathbb{R}^4 | x^2 + y^2 + z^2 + w^2 = 1\}$$

を 3 次元球面と呼ぶ

- 4 次元球の表面
- $\mathbb{R}^3$ の中では描けない
- $S^3 \simeq \mathbb{R}^3 \cup \{\infty\}$

以降，結び目は S^3 の中にあると考える

トーラス

定義 2.2

$S^1 \times S^1$ をトーラスと呼ぶ

右図の青い線をロンジチュード呼ぶ

- ドーナツの形
- 中身は空

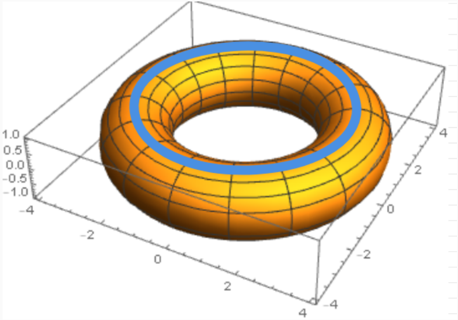


Figure 8: トーラス

結び目近傍, 補空間

定義 2.3 (結び目近傍)

結び目 K を芯にするようなトーラスを結び目の近傍といい, $N(K)$ で表す

定義 2.4 (補空間)

$\overline{K} = cl(S^3 - N(K))$ を K の補空間という

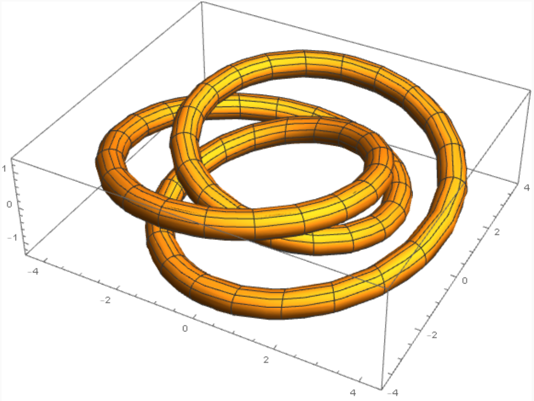
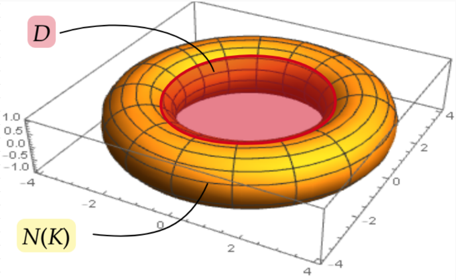


Figure 9: 三葉結び目の近傍

自明性判定問題へのアプローチ

定理 2.1

結び目 K が自明
$\Longleftrightarrow K$ を境界とする円盤 D が存在
$\Longleftrightarrow \partial\overline{K}$ のロンジチュードを境界とする円盤 D が存在 ($\partial M =$ M の境界)

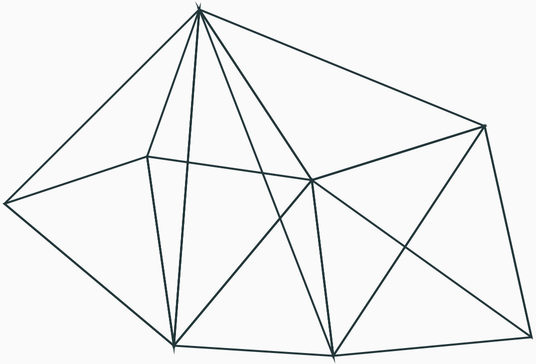


単体分割

滑らかなままでは図形を考えにくい $\longrightarrow$ 単体分割を考える

定義 2.5

3 次元図形を四面体の組み合わせで表現したものを単体分割と呼ぶ

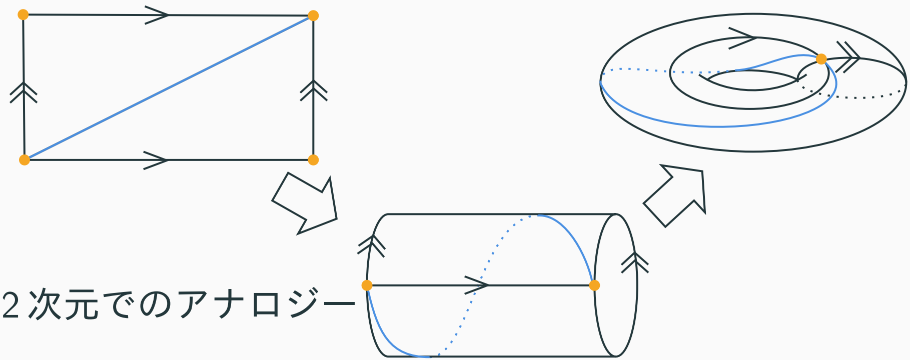


少なくとも本演習で出てくる図形は全て四面体の組み合わせで表現できる

1頂点単体分割

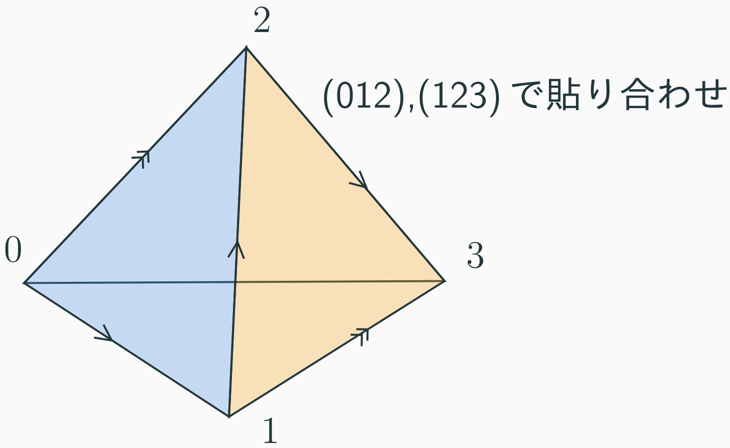
定義 2.6 (1頂点単体分割)

全ての頂点が *1* 点に集まっているような単体分割を *1* 頂点単体分割と呼ぶ



2次元でのアナロジー

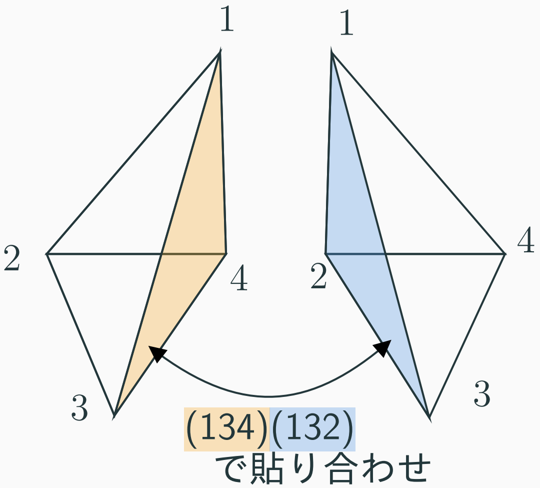
1 頂点単体分割の例



自明結び目の補空間の 1 頂点分割の例

単体分割の符号化

四面体の頂点に番号をふり，どの面を貼り合わせるかを記憶することで計算機で単体分割を扱うことができる



単体分割

定理 (再掲)

結び目 K が自明
$\iff K$ を境界とする円盤 D が存在
$\iff \overline{K}$ のロンジチュードを境界とする円盤 D が存在

$\overline{K}$ は単体分割を考えることで計算機上で表現できる
$\overline{K}$ 内の円盤はどう表現する? $\rightarrow$ Normal surface理論

Normal surface 理論

Normal surface

定義 3.1 (Normal surface)

曲面 S が単体分割 $\mathcal{T}$ に適切に埋め込まれており，各四面体 $\Delta_i \subset \mathcal{T}$ と横断的にのみ交わるとき，S を *normal surface* と呼ぶ

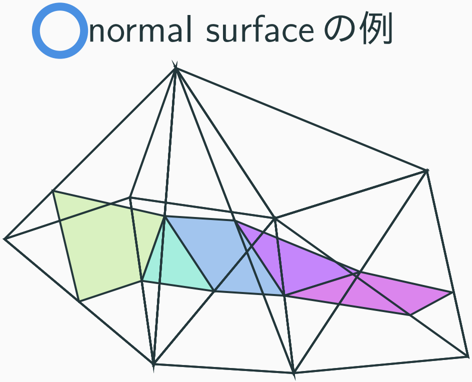
適切に埋め込まれている

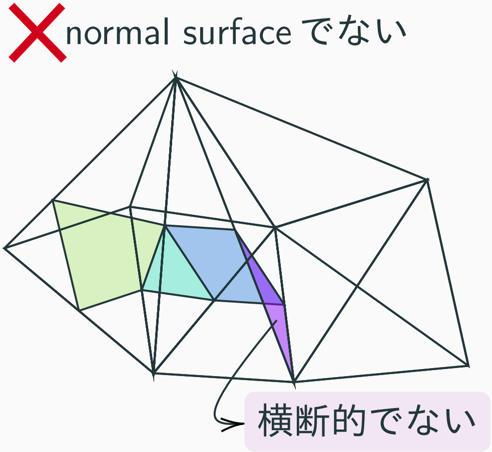
$\leftrightarrow$ S が自己交差せず，$\partial S \subset \partial \mathcal{T}$ を満たす

横断的に交わる

$\leftrightarrow$ S が各四面体と頂点のみ，辺のみ，面のみで交わらない

Normal surface の例





自明性判定問題へのアプローチ

定理 3.1 (Haken　1961)

結び目 K が自明
$\Longleftrightarrow \overline{K}$ のロンジチュードを境界とする円盤 D が存在
$\Longleftrightarrow \overline{K}$ の単体分割 $\mathcal{T}$ のロンジチュードを境界とする *normal surface* S が存在

Elementary disc

定義 3.2

四面体と横断的に交わる平面はどの辺上に頂点があるかによって，*4* つの三角形と *3* つの四角形のいずれか
この三角形と四角形を *elementary disc* という

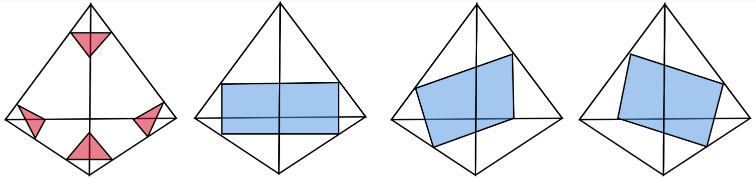
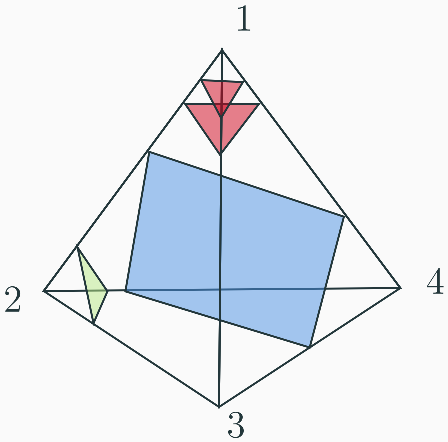


Figure 10: 7 つの elementary disc

Normal surface のベクトル表示

normal surface S が各四面体において，どの elementary disc をいくつ含むかを数えることにより，normal suface をベクトル $\boldsymbol{v}(S) \in \mathbb{Z}^{7n}$ で表現できる



$$v = (\ 2\ ,\ 1\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ )$$

Normal surfaceのベクトル表示

n個の四面体に$1 \cdots n$までの番号をふり，i番目の四面体の elementary disc を以下のように表す

$$\boldsymbol{v}_i = (v^{\triangle}_{i,1}, v^{\triangle}_{i,2}, v^{\triangle}_{i,3}, v^{\triangle}_{i,4}, v^{\square}_{i,1}, v^{\square}_{i,2}, v^{\square}_{i,3}) \in \mathbb{Z}^7$$

$\mathcal{T}$の normal surface は$\boldsymbol{v_i}$をn個並べて表現

$$\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n) \in \mathbb{Z}^{7n}$$

Normal surface のベクトル表示

$S(\boldsymbol{v}) =$ ベクトル $\boldsymbol{v}$ で表される曲面 S

$\boldsymbol{v}(S) =$ 曲面 S のベクトル表示 $\boldsymbol{v}$

- normal surface$S \Rightarrow$ ベクトル $\boldsymbol{v}(S) \cdots$ 一意に定まる
- ベクトル $\boldsymbol{v} \Rightarrow$ 曲面 $S(\boldsymbol{v}) \cdots$ 一意に定まるが normal surface とは限らない

しかし，$S(\boldsymbol{v})$ が normal surface になる $\boldsymbol{v}$ の必要十分条件が知られている

Normal surface のベクトル表示

定理 3.2 (Haken 1961)

以下の条件を満たすことが，$S(\boldsymbol{x})$ が *normal surface* になる必要十分条件である

- $\boldsymbol{x} \geq \mathbf{0}$
- $A\boldsymbol{x} = \mathbf{0}$*(macthing equations)*
- *quadrilateral constrain*

各条件の説明

$\boldsymbol{x} \geq \mathbf{0}$

ベクトルの各要素は負にならない
(elementary disc の数が負という状況はありえない)

各条件の説明

$$Ax = \mathbf{0}$$

張り合わせている三角形の 2 辺を通る elementary disc の個数が一致

$$x_{i,s}^{\triangle} + x_{i,t}^{\square} = x_{j,u}^{\triangle} + x_{j,v}^{\square}$$

という式が最大で $6n$ 個存在
$\rightarrow$ 行列で表したものが
$Ax = \mathbf{0}$(macthing equations)

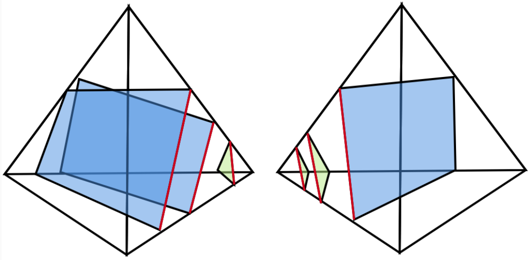


Figure 11: 貼り合わせの様子 (赤い線の数が一致する)

各条件の説明

quadrilateral constrain(以下 QC)

一つの四面体の中に四角形はたかだか 1 種類

$\Longleftrightarrow$ 各四面体において次の 3 条件のうちいずれかが成立

(a). $x^{\square}_{i,1} \geq 0, x^{\square}_{i,2} = 0, x^{\square}_{i,3} = 0$

(b). $x^{\square}_{i,1} = 0, x^{\square}_{i,2} \geq 1, x^{\square}_{i,3} = 0$

(c). $x^{\square}_{i,1} = 0, x^{\square}_{i,2} = 0, x^{\square}_{i,3} \geq 1$

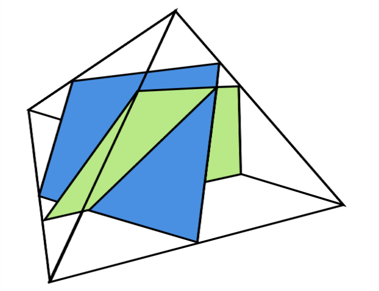


Figure 12: QC が成り立たない例

自明性判定問題へのアプローチ

定理 3.3 (Haken　1961(再掲))

結び目 K が自明
$\Longleftrightarrow \overline{K}$ のロンジチュードを境界とする円盤 D が存在
$\Longleftrightarrow \overline{K}$ の単体分割 $\mathcal{T}$ のロンジチュードを境界とする *normal surface* S が存在

Normal surface のベクトル表示

定理 3.1，定理 3.2 より自明性判定問題は以下の問題に帰着される

問題 3.1

入力: $\overline{K}$ の単体分割 $\mathcal{T}$ の符号

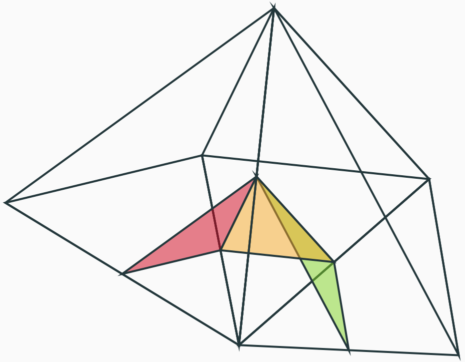
以下を満たすベクトル $\boldsymbol{x} \in \mathbb{Z}^{7n}$ は存在するか

- *normal surface* $\iff \begin{cases} \boldsymbol{x} \geq \boldsymbol{0} \\ A\boldsymbol{x} = \boldsymbol{0} \\ \textit{quadrilateral constrains(QC)} \end{cases}$
- $S(\boldsymbol{x})$ は円盤で，その境界は $\partial\mathcal{T}$ のロンジチュード

Vertex link

定義 3.3 (Vertex link)

$\mathcal{T}$ の 1 つの頂点を囲むような *normal surface* を *vertex link* という



Vertex link

定理 3.4

$\overline{K}$ の単体分割を $\mathcal{T}$ とする
vertex link の境界は $\mathcal{T}$ のロンジチュードにならない

$\Rightarrow$normal surface の探索から vertex link は除いてよい

1 頂点単体分割の vertex link

さらに，$\mathcal{T}$ が 1 頂点単体分割であるとき，以下が知られている

定理 3.5 (Burton-Ozlen 2012)

$S \subset \mathcal{T}$ が *vertex link* でない $\iff \exists i, j [v^{\triangle}_{i,j} = 0]$

任意の結び目の補空間には，1 頂点単体分割が存在することが知られている

$\Rightarrow$ 任意の結び目補空間に対して定理 3.5 を適用できる

オイラー標数

S の形を知るためにオイラー標数を考える

定義 3.4 (オイラー標数)

S を曲面とする

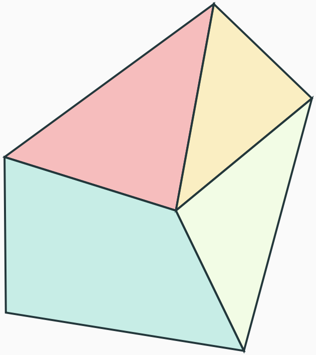
$$\chi(S) = (\text{面の数}) - (\text{辺の数}) + (\text{頂点の数})$$

をオイラー標数という

オイラー標数は位相不変量

$\iff$ 連続変形できるならば同じ値になる

オイラー標数の例

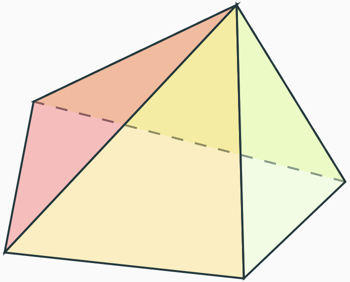


円盤:D

- (面の数) $= 4$
- (辺の数) $= 9$
- (頂点の数) $= 6$

$$\chi(D) = 4 - 9 + 6 = 1$$

オイラー標数の例



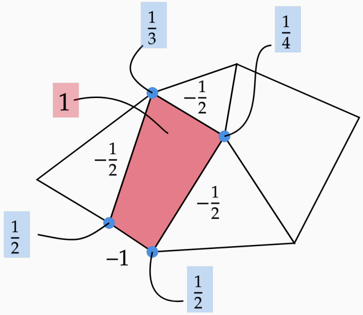
球面:S

- (面の数) $= 5$
- (辺の数) $= 8$
- (頂点の数) $= 5$

$$\chi(S) = 5 - 8 + 5 = 2$$

Normal suface のオイラー標数

normal surface のオイラー標数は各 elementary disc の貢献度を考えることにより，線形写像 $\chi : \mathbb{Z}^{7n} \to \mathbb{Z}$ として表現できる．



(四角形の貢献度)$= 1 - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} - 1 + \frac{1}{2} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{1}{12}$

Normal surface のオイラー標数

S を結び目補空間 $\overline{K}$ の単体分割 $\mathcal{T}$ 内の normal surface とし，S を表すベクトルを $\boldsymbol{v}$ とする

$$\chi(\boldsymbol{v}) > 0 \iff S \text{ は円盤または球面}$$

$\chi(\boldsymbol{v})$ は線形関数であるから $\chi(\boldsymbol{v}) > 0$ は線形条件である

Crushing procedure

定理 3.6 (Crushing procedure(Jaco-Rubinstein　2003))

$\mathcal{T}$ を結び目補空間 $\overline{K}$ の単体分割とし，$S \subset \mathcal{T}$ を以下を満たす *normal surface* とする

- *vertex link* でない
- $\chi(S) > 0$
- ∂S は $\mathcal{T}$ のロンジチュードではない

このとき，$\mathcal{T}$ より少ない四面体で $\overline{K}$ の *1* 頂点単体分割を具体的に構成できる

Normal surface のベクトル表示

今考えているのは次の問題であった

問題 (再掲)

入力: $\overline{K}$ の単体分割 $\mathcal{T}$ の符号

以下を満たすベクトル $\boldsymbol{x} \in \mathbb{Z}^{7n}$ は存在するか

- $\boldsymbol{x} \geq \boldsymbol{0}$
- $A\boldsymbol{x} = \boldsymbol{0}$
- *quadrilateral constrains(QC)*
- $S(\boldsymbol{x})$ は円盤で，その境界は $\partial\mathcal{T}$ のロンジチュード

自明性判定問題へのアプローチ

以上から，次の手順で問題3.1を解けることがわかる

アルゴリズム 3.1 (Burton-Ozlen　2012)

1. *vertex link*ではなく，$\chi(S) > 0$を満たす*normal surface*Sを見つける
   - Sが見つからないとき →*no*を出力して停止
   - Sが見つかったとき →*2*を実行
2. $\partial\mathcal{T}$上での∂Sの形を調べる
   - ∂Sがロンジチュード →*yes*を出力して停止
   - ∂Sがロンジチュード以外 →*crushing*を行い，*crushing*後の単体分割で*1*を実行

自明性判定問題へのアプローチ

よって，次を満たすベクトル $\boldsymbol{x} \in \mathbb{Z}^{7n}$ を探せば良い

- normal surface $\iff \begin{cases} \boldsymbol{x} \geq \mathbf{0} \\ A\boldsymbol{x} = \mathbf{0} \\ \text{quadrilateral constrains(QC)} \end{cases}$
- vertex link ではない $\iff \exists i, j[x^{\triangle}_{i,j} = 0]$
- $\chi(\boldsymbol{x}) > 0$

自明性判定問題へのアプローチ

次を満たすベクトル $\boldsymbol{x} \in \mathbb{Z}^{7n}$ を探せば良い

- $\boldsymbol{x} \geq \mathbf{0}$ $\cdots$ 線形
- $A\boldsymbol{x} = \mathbf{0}$ $\cdots$ 線形
- quadrilateral constrains(QC) $\cdots$ 非線形
- $\exists i, j [x_{i,j}^{\triangle} = 0]$ $\cdots$ 非線形
- $\chi(\boldsymbol{x}) > 0$ $\cdots$ 線形

自明性判定問題へのアプローチ

各四面体に対する QC，$x^{\triangle}_{i,j} = 0$ となる i, j が定まったと仮定する

- $\boldsymbol{x} \geq \boldsymbol{0}$
- $A\boldsymbol{x} = \boldsymbol{0}$
- QC
- $\exists i, j [x^{\triangle}_{i,j} = 0]$
- $\chi(\boldsymbol{x}) > 0$

$\Longrightarrow$

- $\boldsymbol{x} \geq \boldsymbol{0}$
- $A\boldsymbol{x} = \boldsymbol{0}$
- 各四面体に対する QC
- $x^{\triangle}_{i,j} = 0$
- $\chi(\boldsymbol{x}) > 0$

$f(x)$ $\cdots$ 線形　$g(x)$ $\cdots$ 非線形

自明性判定問題へのアプローチ

各四面体に対する QC，$x^{\triangle}_{i,j} = 0$ となる i, j が定まったと仮定する

- $\boldsymbol{x} \geq \mathbf{0}$ $\cdots$ 線形
- $A\boldsymbol{x} = \mathbf{0}$ $\cdots$ 線形
- 各四面体に対する QC $\cdots$ 線形
- $x^{\triangle}_{i,j} = 0$ $\cdots$ 線形
- $\chi(\boldsymbol{x}) > 0$ $\cdots$ 線形

自明性判定問題へのアプローチ

各四面体に対する QC，$x_{i,j}^{\triangle} = 0$ となる i, j が定まったと仮定する

$\chi(\boldsymbol{x})$ を目的関数だと思い，最大化することを考える

目的関数 $\chi(\boldsymbol{x})$

制約条件

- $\boldsymbol{x} \geq \boldsymbol{0}$
- $A\boldsymbol{x} = \boldsymbol{0}$
- 各四面体に定めた QC
- $x_{i,j}^{\triangle} = 0$
- $\boldsymbol{x} \in \mathbb{Z}^{7n}$ $\leftarrow$ 整数解を見つけるのは難しい

自明性判定問題へのアプローチ

各四面体に対する QC，$x^{\triangle}_{i,j} = 0$ となる i, j が定まったと仮定する

$\mathbb{Z}^{7n}$ から解を探す代わりに，$\mathbb{R}^{7n}$ から解を探すことを考える

目的関数 $\chi(\boldsymbol{x})$

制約条件

- $\boldsymbol{x} \geq \boldsymbol{0}$
- $A\boldsymbol{x} = \boldsymbol{0}$
- 各四面体に定めた QC
- $x^{\triangle}_{i,j} = 0$
- $\boldsymbol{x} \in \mathbb{R}^{7n}$

この問題は線形計画問題 $\Rightarrow$ 多項式時間で解くことができる

自明性判定問題へのアプローチ

各四面体に対するQC，$x_{i,j}^{\triangle} = 0$ となる i, j が定まったと仮定する

線形計画問題の解から，制約条件を満たす整数解を構成できる

$\because$ 制約条件は全て線形なので，線形計画問題の解は $\boldsymbol{x} \in \mathbb{Q}^{7n}$

よって，ある正の有理数 $k \in \mathbb{Q}$ が存在して $k\boldsymbol{x} \in \mathbb{Z}^{7n}$

この時，$\chi(k\boldsymbol{x}) = k\chi(\boldsymbol{x}) > 0$

自明性判定問題へのアプローチ

各四面体に対する QC,$x^{\triangle}_{i,j} = 0$ となる i, j を分枝限定法で定めるのが Burton-Ozlen のアルゴリズム

前述したように Burton-Ozlen のアルゴリズムの最悪計算量は $\mathcal{O}(3^n)$ だが，実際には多項式時間に近い振る舞い

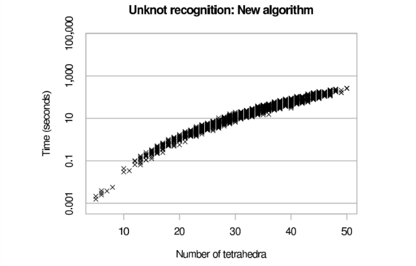


Figure 13: Burton-Ozren のアルゴリズムの実行時間

出典：Benjamin A. Burton, Melih Ozlen, *A FAST BRANCHING ALGORITHM FOR UNKNOT RECOGNITION WITH EXPERIMENTAL POLYNOMIAL-TIME BEHAVIOUR*, Preprint, **https://arxiv.org/abs/1211.1079** (2012)

自明性判定問題へのアプローチ

本演習では，各四面体に対するQC,$x_{i,j}^{\triangle} = 0$となるi, jを，結び目補空間の単体分割のface pairingグラフの木分解に対して動的計画法を用いて定めるアルゴリズムを提案する

FPT アルゴリズム

木分解

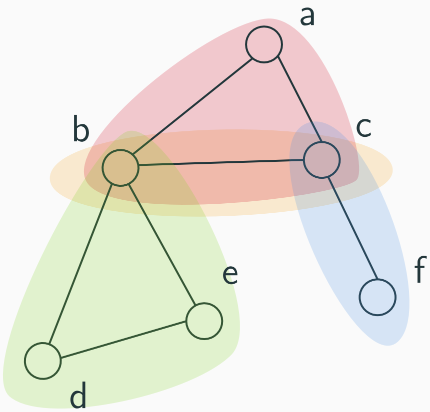
定義 4.1 (木分解)

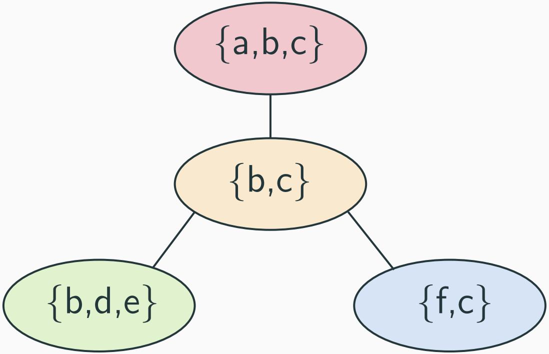
木 $\mathscr{T} = (V_{\mathscr{T}}, E_{\mathscr{T}})$ で，以下の条件を満たすものをグラフ $G = (V, E)$ の木分解という

- $\mathscr{T}$ の各節点 $X \in V_{\mathscr{T}}$ は V の部分集合
- 各辺について $(u, v) \in E \Rightarrow \mathscr{T}$ の節点 X で $u \in X \wedge v \in X$ となるものが存在
- 各頂点 $v \in V$ について，$S = \{X \in V_{\mathscr{T}} | v \in X\}$ は $\mathscr{T}$ の部分木を誘導する

木分解はグラフに対して一意には定まらないことに注意

木分解の例





木幅

定義 4.2 ($\mathscr{T}$ の木幅)

$\mathscr{T} = (V_{\mathscr{T}}, E_{\mathscr{T}})$ をグラフ G の木分解とする.

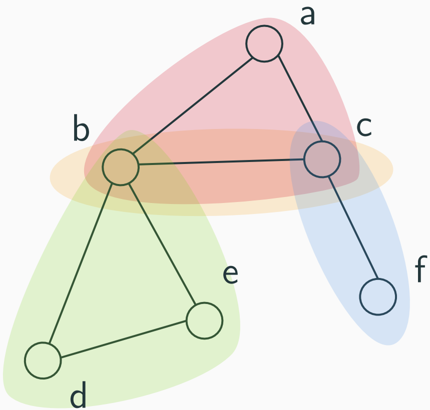
$$\max_{X \in V_{\mathscr{T}}} |X| - 1$$

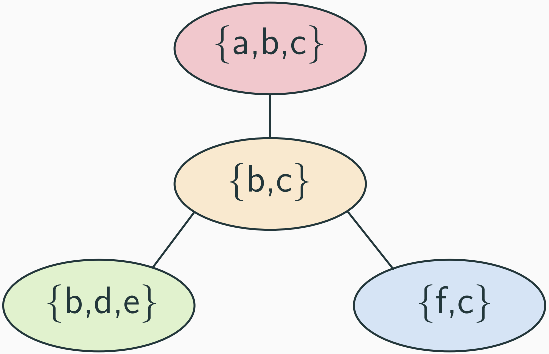
を $\mathscr{T}$ の木幅と呼ぶ

定義 4.3 (G の木幅)

G の全ての木分解を考え，その最小値を G の木幅と呼ぶ.

木分解の例





この例だと，木幅 $= \max_{X \in V_{\mathscr{T}}} |X| - 1 = 2$

FPT アルゴリズム

定義 4.4 (FPT アルゴリズム)

n を入力サイズ，k を n と独立なパラメーターとする
最悪計算量が $\mathcal{O}(f(k)\mathit{poly}(n))$ となるようなアルゴリズムを
Fixed Parametr Tractable (FPT) アルゴリズムという
ここで $f(k)$ は任意の関数，$\mathit{poly}(n)$ は多項式関数

(例)
計算量が $\mathcal{O}(2^k n)$ となるアルゴリズム
k を定数として扱うと 2^k は定数 $\Rightarrow$ FPT アルゴリズム

FPT アルゴリズムの例

最小頂点被覆

解のサイズを固定することで，$\mathcal{O}(2^k n)$

頂点 **3** 彩色問題

木幅を固定することで $\mathcal{O}(3^k n)$

k は n と独立なので NP 完全 (困難) な問題でも，特定のインスタンスに関しては高速に解くことができる

特に一般のグラフでは NP 完全 (困難) な問題でも，木幅を固定したグラフの木分解に対して動的計画法を用いることで，多項式時間限定になる問題が多く存在

提案アルゴリズム

Face pairing グラフ

定義 5.1 (Face pairing グラフ)

$\mathcal{T}$ を単体分割とする．四面体を頂点として，四面体同士に貼り合わせがあるならその頂点を辺でつなぐ

この操作で作られるグラフを *face pairing* グラフという

提案アルゴリズム

アルゴリズム 5.1 (提案アルゴリズム)

K を結び目，D をその図式とする．K の補空間 $\overline{K}$ の単体分割を $\mathcal{T}$ とし，$\mathcal{T}$ の *face pairing* グラフを $G_{\mathcal{T}}$ とする．また，$k \in \mathbb{N}$ を定数とする．

入力 $\overline{K}$ の *1* 頂点単体分割 $\mathcal{T}$

出力 $\begin{cases} \textit{yes} & (G_{\mathcal{T}}\text{の木幅} \leq k \wedge K \text{ が自明}) \\ \textit{no} & (G_{\mathcal{T}}\text{の木幅} \leq k \wedge K \text{ が非自明}) \\ \textit{no} & (G_{\mathcal{T}}\text{の木幅} > k) \end{cases}$

提案アルゴリズム

アルゴリズム

1. $\mathcal{T}$ の *face pairing* グラフ $G_{\mathcal{T}}$ を構成
2. $G_{\mathcal{T}}$ の良い木分解 $\mathscr{T}$ を構成
   - *if* $\mathscr{T}$ の木幅 $> k \to$ *no* を出力して停止
3. $\mathscr{T}$ に動的計画法を適用して，$\chi(\boldsymbol{x}) > 0$ とできる *QC*,$x^{\triangle}_{i,j} = 0$ となる i, j を求める

本発表では，動的計画法の詳細については立ち入らない

計算量

アルゴリズムの計算量：$\mathcal{O}(3^h \text{poly}(n))$
ただし $h =$ ($\mathscr{T}$ の高さ)
h は n と独立ではないので提案アルゴリズムは多項式時間限定であることは示せていない

多項式時間限定だが，証明できてないだけでは?
$\Rightarrow$ 計算機実験

実験

実装

提案アルゴリズムを Regina を用いて実装

Regina ・・・ 低次元トポロジーや normal surface に関する計算を行うことができるソフトウェア

Regina

Software for
low-dimensional topology

https://regina-normal.github.io/ より引用

入力データ

非自明結び目

Regina で標準にサポートされている全ての最小交点数 12 以下の素な結び目の最小交点図式から構成した補空間の単体分割 (2997 個)

非自明結び目の図式の符号 → 結び目補空間の単体分割 → 入力

非自明結び目の図式の符号は Regina 公式ページ

(`https://regina-normal.github.io/data.html`) からダウンロード

入力データ

自明結び目

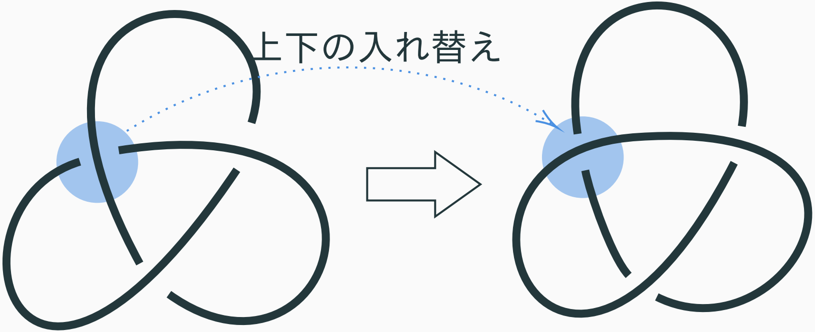
非自明結び目の図式の交点に対して交差交換を行い，
単調な図式を構成
その図式から構成した補空間の単体分割 (2997 個)

非自明結び目の図式の符号 $\xrightarrow{\text{交差交換}}$ 自明結び目の図式の符号

$\Rightarrow$ 結び目補空間の単体分割 $\Rightarrow$ 入力

交差交換

交差交換 $\cdots$ 交点の上下を入れ替える操作

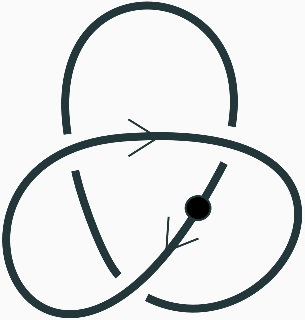


単調な図式

結び目 K の図式を D とする

ある基点から図式 D をたどった時に，各交点において上を先に通るとき，D を単調な図式という

D が単調 $\Rightarrow K$ は自明結び目



単調な図式の例

実験結果

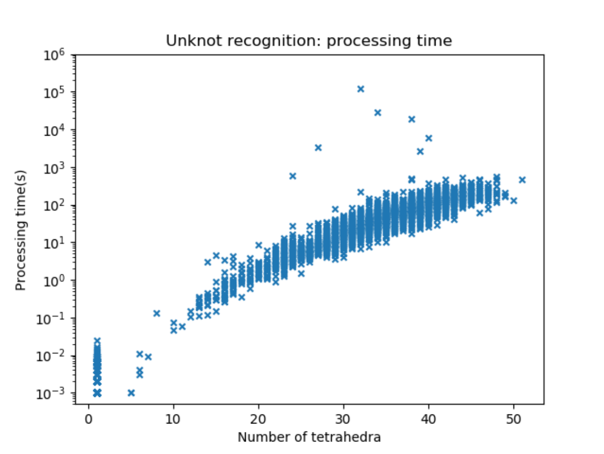


Figure 14: 実行時間

実験結果

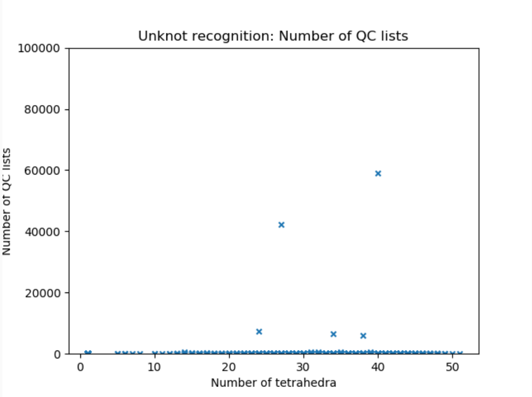


Figure 15: QC のリストの数

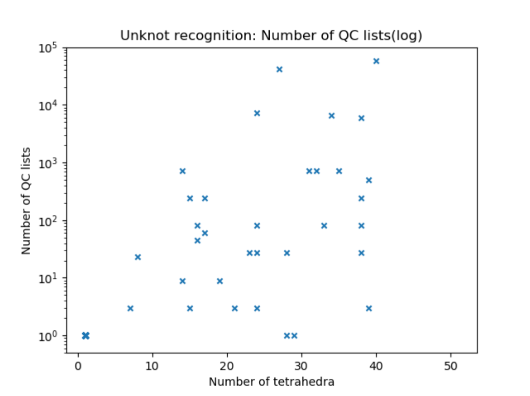
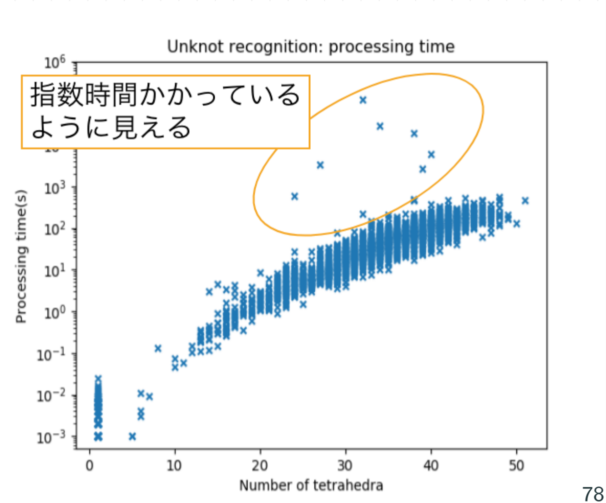


Figure 16: QC のリストの数 (log スケール)

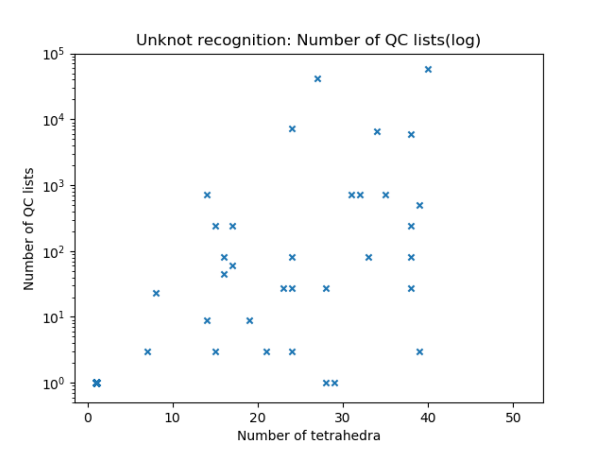
考察

実際にも多項式時間限定
ではなさそう



考察

計算量に本質的に関わる
QC のリストの数も指数
関数的に増加



まとめ

- 結び目補空間の単体分割の face pairing グラフの木分解に対して動的計画法を適用することで自明性判定を行うアルゴリズムを構成した
- 提案アルゴリズムが多項式時間限定であることは示せていない
- 計算機実験からも，提案アルゴリズムは多項式時間限定にはなりそうにない

今後の課題

- 実験
  - 素ではない結び目や，最小交点以外の図式，単調ではない図式を入力とする
- アルゴリズム改善
  - 提案アルゴリズムを修正して多項式時間限定にできるか
  - さらに制約を加えることで多項式時間限定になるか
  - ３次元多様体に対する線形分解など他の分解についてアルゴリズムを構成できるか

良い木分解

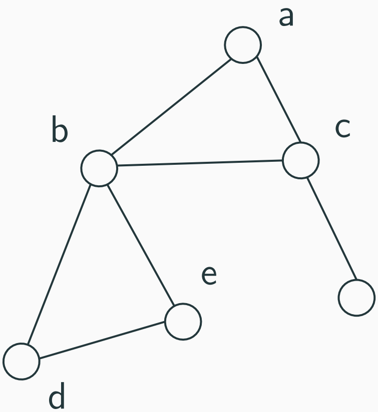
定義 6.1 (良い木分解)

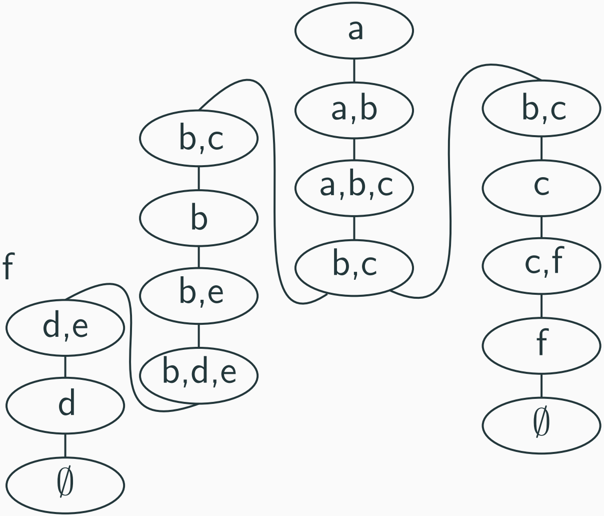
以下の条件を満たす木分解を良い木分解という.

- 葉は空集合
- 各節点の子の数は高々 *2* つ
    - 子の数が *1* つ $\rightarrow |X_i| = |X_{i-1}| \pm 1$
    - 子の数が *2* つ $\rightarrow X_i = X_j = X_k$(ただし，$X_j, X_k$ は X_i の子)

- 良い木分解に対しては動的計画法を考えやすい
- 良い木分解のみを考えても元のグラフの木幅は変化しない

良い木分解の例

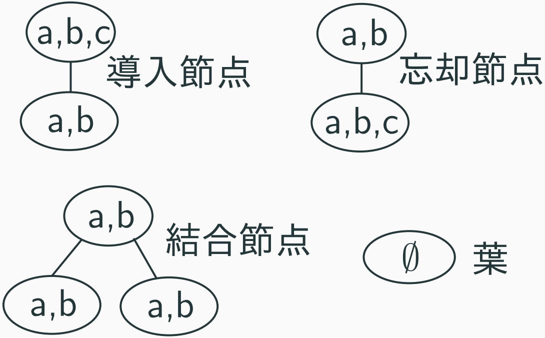




良い木分解上での動的計画法

良い木分解の節点は右の4つのいずれか

$\longrightarrow$ 動的計画法を考えるときは4つの動きを考えるだけで良い



提案アルゴリズム

アルゴリズム

1. $\mathcal{T}$ の *face pairing* グラフ $G_\mathcal{T}$ を構成
2. $G_\mathcal{T}$ の良い木分解 $\mathscr{T}$ を構成
    - *if* $\mathscr{T}$ の木幅 $> k \to$ *no* を出力して停止
3. $\mathscr{T}$ に動的計画法を適用して *QC*,$x^{\triangle}_{i,j} = 0$ となる i, j を求める

提案アルゴリズム

アルゴリズム

- $\chi(\boldsymbol{x}) > 0$ となる *QC*，i, j が
$$\begin{cases} \text{見つからない} \Rightarrow \textit{no} \text{ を出力して停止} \\ \text{見つかる} \Rightarrow \text{以下を行う} \end{cases}$$
- $\partial\mathcal{T}$ 上での ∂S の形を調べる
  - ∂S がロンジチュード $\rightarrow$*yes* を出力して停止
  - ∂S がロンジチュード以外 $\rightarrow$*crushing* を行い，*crushing* 後の単体分割から *1* 頂点単体分割を構成し *1* を実行

提案アルゴリズム

アルゴリズム (動的計画法)

(i). まだ選んでない $x_{i,j}^{\triangle}$ を一つ選ぶ

(ii). $\mathcal{N}$ を線形条件の集合とし，$\mathcal{N} = \{\boldsymbol{x} \geq 0, A\boldsymbol{x} = 0, x_{i,j}^{\triangle} = 0\}$ とする

(iii). $\mathscr{T}$ の各節点 X_i に対して，線形条件の集合のリスト $\mathscr{L}_i$ を次のように更新

提案アルゴリズム

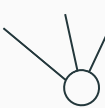
アルゴリズム (動的計画法)

X_i が葉

$$\mathscr{L}_i = \{\emptyset\}$$

face pairing グラフ

$\vdots$



QC lists

$\vdots$

$\mathscr{L}_i \quad \{\ \emptyset\ \}$

提案アルゴリズム

アルゴリズム (動的計画法)

X_i が導入節点

$\mathscr{L}_i = \{\}$ と初期化

$X_i = X_j \cup \{v\}$ とする

任意の $\mathcal{M} \in \mathscr{L}_j$ と，v における QC の各条件

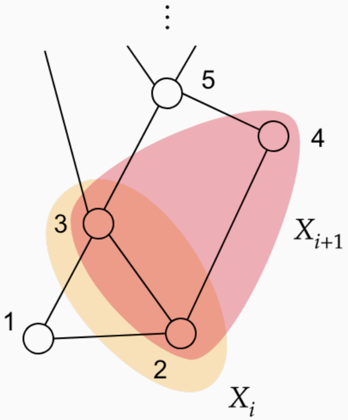
$q \in \{(a), (b), (c)\}$ に対して

$\mathcal{N} \cup \mathcal{M} \cup \{q\}$ のもとで $\chi(\boldsymbol{x}) > 0$ とできるなら，

$\mathcal{M} \cup \{q\}$ を $\mathscr{L}_i$ に追加

提案アルゴリズム

face pairing graph



QC list

$$N = \left\{ x \geq 0, Ax = 0, x^{\triangle}_{i,j} = 0 \right\}$$

$\mathscr{L}_{i+1}$ $\{\}$

$$\Uparrow$$

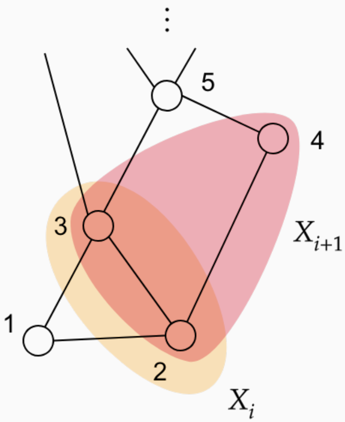
$\begin{pmatrix}1\\a\end{pmatrix}\begin{pmatrix}2\\c\end{pmatrix}\begin{pmatrix}3\\b\end{pmatrix} \cup \boxed{\begin{pmatrix}4\\a\end{pmatrix}} \cup N$のもとで$\chi(x) > 0$とできるか?

$$\mathscr{L}_i \quad \left\{ \begin{pmatrix}1\\a\end{pmatrix}\begin{pmatrix}2\\a\end{pmatrix}\begin{pmatrix}3\\b\end{pmatrix}, \cdots \begin{pmatrix}1\\a\end{pmatrix}\begin{pmatrix}2\\c\end{pmatrix}\begin{pmatrix}3\\b\end{pmatrix} \right\}$$

$$\vdots$$

提案アルゴリズム

face pairing graph



QC list

$$\mathcal{N} = \left\{x \geq 0, Ax = 0, x^{\triangle}_{i,j} = 0\right\}$$

$$\mathscr{L}_{i+1} \quad \left\{\binom{1}{a}\binom{2}{a}\binom{3}{b}\binom{4}{a}\right\}$$

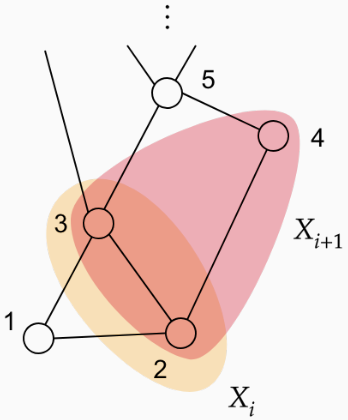
$\binom{1}{a}\binom{2}{c}\binom{3}{b} \cup \boxed{\binom{4}{b}} \cup \mathcal{N}$のもとで$\chi(x) > 0$とできるか?

$$\mathscr{L}_{i} \quad \left\{\binom{1}{a}\binom{2}{a}\binom{3}{b}, \cdots \binom{1}{a}\binom{2}{c}\binom{3}{b}\right\}$$

$$\vdots$$

提案アルゴリズム

face pairing graph



QC list

$$\mathbb{N} = \left\{x \geq 0,\ Ax = 0,\ x^{\Delta}_{i,j} = 0\right\}$$

$$\mathscr{L}_{i+1} \quad \left\{\binom{1}{a}\binom{2}{a}\binom{3}{b}\binom{4}{a},\ \binom{1}{a}\binom{2}{a}\binom{3}{b}\binom{4}{b}\right\}$$

$\binom{1}{a}\binom{2}{c}\binom{3}{b} \cup \boxed{\binom{4}{c}} \cup \mathbb{N}$のもとで$\chi(x) > 0$とできるか?

$$\mathscr{L}_{i} \quad \left\{\binom{1}{a}\binom{2}{a}\binom{3}{b},\ \cdots\ \binom{1}{a}\binom{2}{c}\binom{3}{b}\right\}$$

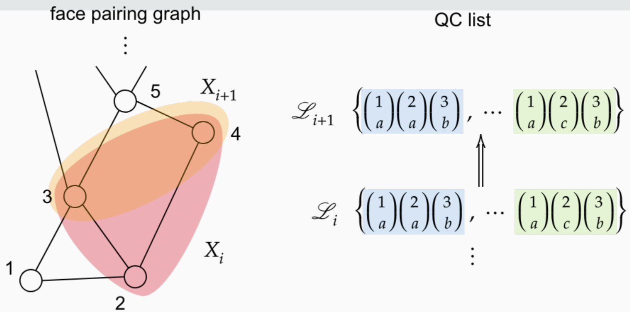
$$\vdots$$

提案アルゴリズム

アルゴリズム (動的計画法)

X_i が忘却節点

$$X_i = X_j - \{v\} \text{とする.}$$
$$\mathscr{L}_i = \mathscr{L}_j$$



提案アルゴリズム

アルゴリズム (動的計画法)

X_i が結合節点

$\mathscr{L}_i =$ と初期化 $X_i = X_j = X_k$ とし，

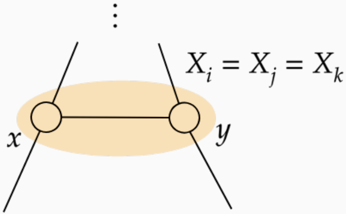
$\mathcal{M}_j \in \mathscr{L}_j, \mathcal{M}_k \in \mathscr{L}_k$ とする

$\forall v \in X_i[(\mathcal{M}_j$ における v の $QC) =$

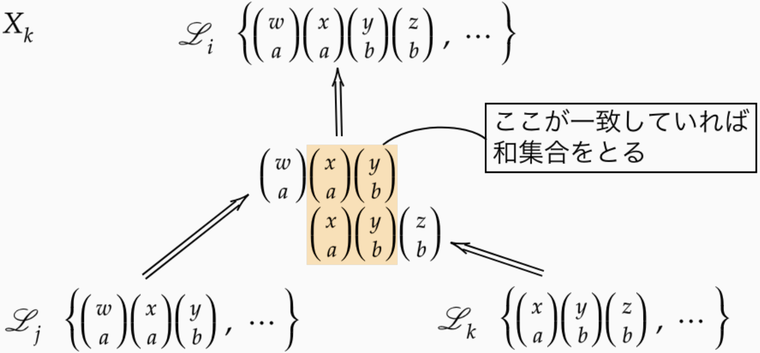
$(\mathcal{M}_k$ における v の $QC)]$ ならば $\mathscr{L}_i$ に $\mathcal{M}_j \cup \mathcal{M}_k$ を追加

提案アルゴリズム

face pairing graph



QC list



提案アルゴリズム

アルゴリズム

- 各節点において $\mathscr{L}_i = \emptyset$ となるなら *(i)* に戻る
- 全ての節点を訪れたとき
  最後の節点で保持している線形条件の集合のリストを $\mathscr{L}$ とする
  - 各 $\mathcal{M} \in \mathscr{L}$ の線形条件の元で $\chi(\boldsymbol{x}) > 0$ となるベクトル $\boldsymbol{x}$ を求め，そのベクトルが表す *normal surface*$S(\boldsymbol{x})$ を求める
  - ∂S が $\partial \mathcal{T}$ のロンジチュードとなる円盤 S が存在するならば *yes* を出力して停止
  - 上記の *normal surface* が存在しないなら *(i)* に戻る

計算量

以下では提案アルゴリズムの計算量の解析を行う
入力となる結び目補空間の単体分割における四面体の数を n とする

- $\mathcal{T}$ の face pairing グラフ $G_{\mathcal{T}}$ を構成
  - 各四面体において貼り合わせを調べれば良いので $\mathcal{O}(n)$
- $G_{\mathcal{T}}$ の良い木分解 $\mathscr{T}$ を構成
  - $\mathcal{O}(k^{\mathcal{O}(k^3)}n)$ で構成できることがわかっている [10]

計算量

動的計画法に関する各節点での動き

X_i が導入節点

$\mathscr{L}_i = \{\}$ と初期化
$X_i = X_j \cup \{v\}$ とする
任意の $\mathcal{M} \in \mathscr{L}_j$ と，v における QC の各条件
$q \in \{(a), (b), (c)\}$ に対して
$\mathcal{N} \cup \mathcal{M} \cup \{q\}$ のもとで $\chi(\boldsymbol{x}) > 0$ とできるなら，
$\mathcal{M} \cup \{q\}$ を $\mathscr{L}_i$ に追加

計算量

- $|\mathscr{L}_i| \in \mathcal{O}(3^h)$ である．ただし $h =$ ($\mathscr{T}$ の高さ)．また，$\chi(\boldsymbol{x}) > 0$ とできるかは線形計画問題なので $\mathcal{O}(\mathrm{poly}(n))$.
  よって，$\mathcal{O}(3^h \mathrm{poly}(n))$

X_i が忘却節点

$$X_i = X_j - \{v\} \text{ とする．}$$
$$\mathscr{L}_i = \mathscr{L}_j$$

- リストのコピーなので $\mathcal{O}(3^h n)$

計算量

X_i が結合節点

$\mathscr{L}_i =$ と初期化 $X_i = X_j = X_k$ とし，
$\mathcal{M}_j \in \mathscr{L}_j, \mathcal{M}_k \in \mathscr{L}_k$ とする
$\forall v \in X_i[(\mathcal{M}_j$ における v の QC$) =$
$(\mathcal{M}_k$ における v の QC$)]$ ならば $\mathscr{L}_i$ に $\mathcal{M}_j \cup \mathcal{M}_k$ を追加

- QC の比較に $\mathcal{O}(3^{k^2}n)$，リストのコピーに $\mathcal{O}(3^h n)$ なので，計算量は $\mathcal{O}(3^h n)$

計算量

- 全ての節点を訪れたとき
  最後の節点で保持している線形条件の集合のリストを $\mathscr{L}$ とする
  - 各 $\mathcal{M} \in \mathscr{L}$ の線形条件の元で $\chi(\boldsymbol{x}) > 0$ となるベクトル $\boldsymbol{x}$ を求め，そのベクトルが表す normal surface$S(\boldsymbol{x})$ を求める
  - ∂S が $\partial \mathcal{T}$ のロンジチュードとなる円盤 S が存在するならば yes を出力して停止
  - 上記の normal surface が存在しないなら (i) に戻る

∂S が $\partial \mathcal{T}$ のロンジチュードかどうかは $\mathcal{O}(n^2)$ 判定可能.
$|\mathscr{L}_i| \in \mathcal{O}(3^h)$ なので，計算量は $\mathcal{O}(3^h \text{poly}(n) n^2)$

計算量

crushing はたかだか $\mathcal{O}(n)$ 回

以上をまとめると，全体での計算量は $\mathcal{O}(3^h \text{poly(n)})$

$\mathscr{T}$ の高さ h は，n と独立ではないので提案アルゴリズムは多項式時間限定ではない．そこで，アルゴリズムの性質を調べるために計算機実験を行った

参考文献 i

[1] Joel Hass, Jeffrey C. Lagarias, and Nicholas Pippenger, *The computational complexity of knot and link problems*. J. Assoc. Comput. Mach. vol.46 (1999), no. 2, 185–211.

[2] Greg Kuperberg, *Knottedness is in NP, modulo GRH*, Advances in Mathematics. vol.256 (2014), 1 493-506.

参考文献 ii

[3] Benjamin A. Burton, Melih Ozlen, *A FAST BRANCHING ALGORITHM FOR UNKNOT RECOGNITION WITH EXPERIMENTAL POLYNOMIAL-TIME BEHAVIOUR*, Preprint, **https://arxiv.org/abs/1211.1079** (2012)

[4] WILLIAM JACO , J. HYAM RUBINSTEIN, *0–EFFICIENT TRIANGULATIONS OF 3–MANIFOLDS*, J. Differential Geom. vol.65 (2003), no. 1, 61–168.

参考文献 iii

[5] Benjamin A. Burton, *Introducing Regina, The 3-Manifold Topology Software*, Experimental Mathmatics,vol.13 (2004),267-272.

[6] Louis H. Kauffman, Sofia Lambropoulou, *Hard Unknots and Collapsing Tangles*, Introductory Lectures on Knot Theory (2011),187-247.

参考文献 iv

[7] Benjamin A. Burton, *A new approach to crushing 3-manifold triangulations*, Discrete Comput. Geom. vol.52 (2014), no. 1, 116–139.

[8] Kristóf Huszár, Jonathan Spreer, Uli Wagner, *On the treewidth of triangulated 3-manifolds*, Leibniz International Proceedings in Informatics, vol.99, 46:1-46:15, (2018)

参考文献 v

[9] De Witt Sumners, *Lifting the curtain: using topology to probe the hidden action of enzymes*, Notices of the American Mathematical Society vol.42(5),(1995),528-537.

[10] Bodlaender, Hans L. (1996), *A linear time algorithm for finding tree-decompositions of small treewidth*, SIAM Journal on Computing vol.25 (6): 1305–1317