

# KEG上におけるクロスキャップ数の計算

Kaito Yamada

2022-11-17

## 概要

クロスキャップ数の計算効率化(仮)

# 目次

<b>1 はじめに</b>	<b>3</b>
1.1 研究背景 . . . . .	3
1.1.1 $S^+$ とは . . . . .	3
1.1.2 KEG とは . . . . .	3
1.2 先行研究 . . . . .	4
1.2.1 $C(K) = 3$ の場合 . . . . .	4
1.2.2 KEG の同値判定 . . . . .	4
1.3 研究目的 . . . . .	4
<b>2 アルゴリズムと計算量</b>	<b>4</b>
2.1 グラフの拡張 . . . . .	4
2.1.1 頂点の分割のアルゴリズム . . . . .	5
2.1.2 $RI^+$ のアルゴリズム . . . . .	5
2.1.3 計算量 . . . . .	5
2.2 $S^+$ の対象の列挙 . . . . .	6
2.2.1 列挙のアルゴリズム . . . . .	6
2.2.2 計算量 . . . . .	6
2.3 $S^+$ の適用 . . . . .	7
2.3.1 KEG における $S^+$ のアルゴリズム . . . . .	7
2.3.2 操作例 . . . . .	7
2.3.3 計算量 . . . . .	8
<b>3 まとめ</b>	<b>9</b>
<b>謝辞</b>	<b>9</b>

# 1 はじめに

## 1.1 研究背景

### 1.1.1 $S^+$ とは

### 1.1.2 KEG とは

前論文 [2] で結び目をオイラー・グラフとして解釈し、Knot Eulerian Graph(KEG) のデータ構造を提案した。今回のアルゴリズムは KEG に対して定義されているが、データ構造の細部について、より適切な変更と命名を施しておく。

まず、視覚的な判別性の向上のため、図 1 に示すように奇数交点の頂点を円、偶数交点の頂点を正方形で表すこととした。

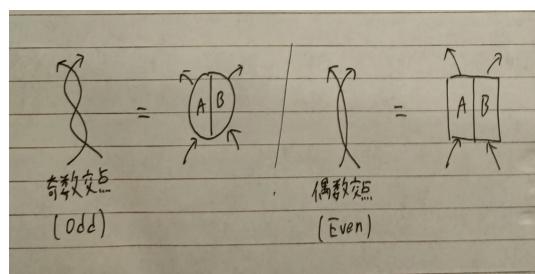


図 1: 頂点の偶奇

そして、図 2 のように、頂点を軸で区切った際の左右を極 (pole) と呼ぶこととする。極は順序を気にせず A/B と命名され、頂点 X の極は  $P_X$  と表す。この時  $P_X = A$  あるいは  $P_X = B$  である。また、空頂点の極は N(None) とする。

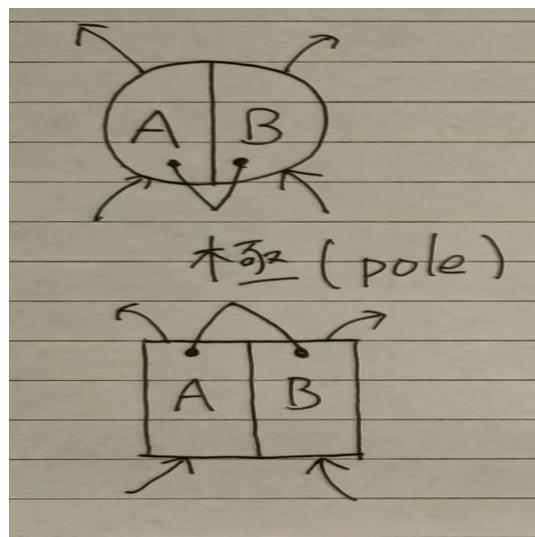


図 2: 頂点の極

また、頂点に出入りする辺の向きについて、図 3 に示す 2 パターンをそれぞれ軸対称 (axial symmetry)、非軸対称 (asymmetry) とする。

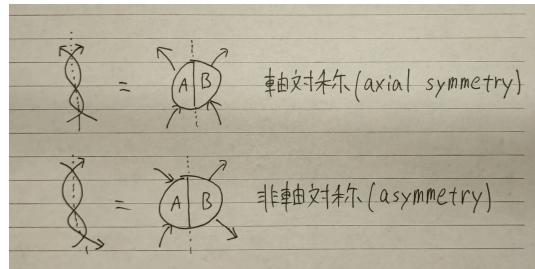


図 3: 軸対称と非軸対称

## 1.2 先行研究

### 1.2.1 $C(K) = 3$ の場合

$C(K) = 3$  の場合については、伊藤昇先生の論文 [1] にて完全なリストが作成されている。ここでは結び目に対して手作業で網羅的に  $S^+$  を施し、生成された結果のうち、結び目として同値なものを削除している。

### 1.2.2 KEG の同値判定

## 1.3 研究目的

## 2 アルゴリズムと計算量

### 2.1 グラフの拡張

グラフを拡張して、 $S^+$  の対象となりうる辺がすべて出てくるようにする。グラフを満足に”ほぐす”には、以下の 2 操作が必要となる。

- 頂点の分割
- $RI^+$

ここでは例として、図 4 の step1 の結び目を拡張していく。

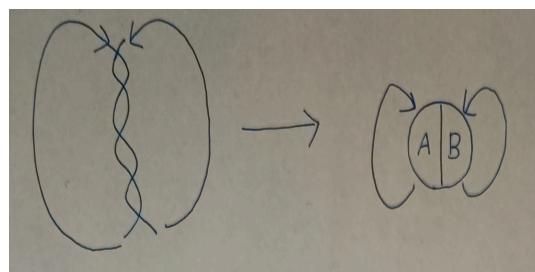


図 4: step1(拡張前)

### 2.1.1 頂点の分割のアルゴリズム

頂点内の領域に対して  $S^+$  を適用するため、一つの頂点を明示的に分割し、間に繋ぐ 2 辺を追加する。繋ぐ辺の向きは二種類存在する（軸に対して垂直/平行）

頂点は、それぞれ以下のように分割できる。分割した状態を図 5 に示す。

- Odd → Odd + Even / Even + Odd
- Even → Odd + Odd / Even + Even

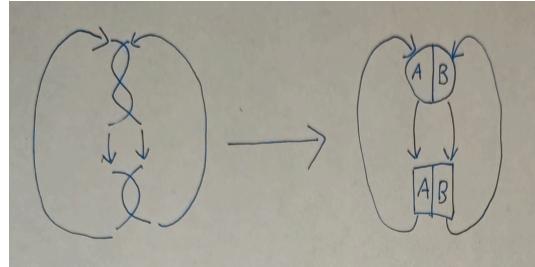


図 5: step1(分割)

### 2.1.2 RI<sup>+</sup> のアルゴリズム

解消できる自己ループを持つ Odd を追加することで、RI<sup>+</sup> を表現する。

$(u, v, P_u, P_v)$  という辺があるとき、その辺を削除し、Odd 頂点 O を追加し、辺  $(u, O, P_u, A)$ ,  $(O, O, B, A)$ ,  $(O, v, B, T_v)$  という辺を追加する。RI<sup>+</sup> をした状態を図 6 に示す。

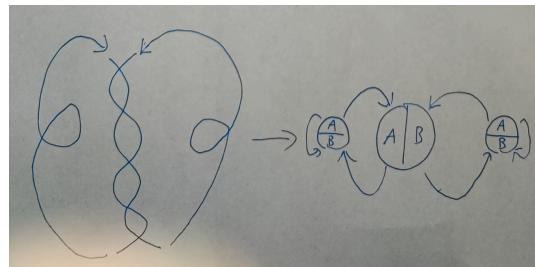


図 6: step1(RI<sup>+</sup>)

どちらの操作も施すと図 7 のようになる。この時、頂点の分割は頂点に対する操作、RI<sup>+</sup> は辺に対する操作であることから、頂点の分割 → RI<sup>+</sup> の順で行う必要がある。

### 2.1.3 計算量

頂点の分割は、ある分割の仕方 (O-OE, O-OO 縛り) で、頂点を 2 つに分割すると、辺の削除が 4 回、辺の追加が 6 回発生する。V 個の頂点すべてに分割を施すので、 $O(V)$

愚直にやると、それぞれ 2 通りの分割の仕方があるため、 $2^V$  通りのグラフが発生しうる。これは  $S^+$  の際に補正を加えることで、E 通りに抑えることができる可能性がある。

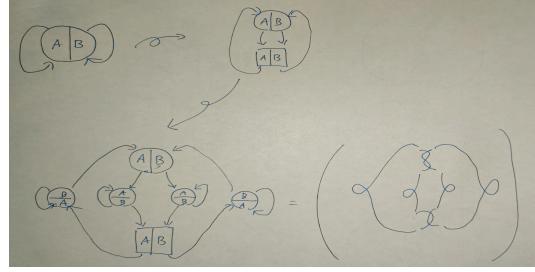


図 7: step1(分割+RI<sup>+</sup>)

RI<sup>+</sup> では、ある辺について、頂点追加が 1 回、辺の削除が 1 回、辺の追加が 3 回発生する。頂点の分割で頂点数が倍になっているので、 $2V$  頂点に対して定数操作を行うため、 $O(V)$  全体で  $O(V)$  あるいは  $O(2^V)$

## 2.2 S<sup>+</sup> の対象の列挙

### 2.2.1 列挙のアルゴリズム

S<sup>+</sup> はグラフへの操作だが、本質的にはオイラー閉路のうち二辺に対する操作である。あるグラフ G が与えられたとき、G に対応するオイラー閉路(辺のリスト)が一つに定まるが、その閉路の中で S<sup>+</sup> の対象となりうる辺のペアは複数現れる。よってそのペア(操作対象)を列挙することが必要である。各ペアの列挙は、以下の操作で実現される。

1. グラフ G のオイラー閉路 C<sub>e</sub> を求める。C<sub>e</sub> の長さはグラフの辺の総数と等しいので、 $\text{len}(C_e) = E$ 。
2.  $i=[0, E-1]$  をループし、その中で  $j=[0, E-1]$  をループする。 $i \neq j$  の場合ペアが組めるので、 $e_s = C_e[i]$ 、 $e_g = C_e[j]$  とおいて  $(e_s, e_g)$  のペアを作ることができる。

コード例を以下に示す。

```

1 Ce = get_euler_circuit(graph)
2 for i in range(len(E)):
3     for j in range(len(E)):
4         if i == j:
5             continue
6         es = Ce[i]
7         eg = Ce[j]
8

```

結果として生成されるペアは  $E(E-1)$  組となる。

### 2.2.2 計算量

1. オイラー閉路の取得 → DFS なので  $O(E + V)$
2. オイラー閉路の連結 → 長さ  $E$  のものを結合するので  $O(E)$
3. ペアの列挙 → 各辺を  $e_s$  として、 $e_g$  の候補が  $E-1$  通りなので  $O(E^2)$

$E = 2V \Rightarrow E + V = E + \frac{1}{2}E = \frac{3}{2}E$  であるため、 $O(E + V) = O(E)$  とみなせる。

## 2.3 $S^+$ の適用

### 2.3.1 KEG における $S^+$ のアルゴリズム

KEG を対象とした  $S^+$  のアルゴリズムはいかに示す通りである。

1. Odd 頂点 O を追加する。
2. a, b から O に、O から c, d に繋がるように辺を追加する。この時 a と d, b と c がそれぞれ O の同じ側 (A/B) に繋がるようにする。即ち、(a, O, P<sub>a</sub>, A), (b, O, P<sub>b</sub>, B), (O, c, B, P<sub>c</sub>), (O, d, A, P<sub>d</sub>) の 4 辺を追加する。
3. C<sub>e</sub> で e<sub>s</sub> と e<sub>g</sub> の間にある辺全てを逆向きにした辺を追加する。即ち、(u, v, P<sub>u</sub>, P<sub>v</sub>) を (v, u, P<sub>v</sub>, P<sub>u</sub>) にする。
4. [e<sub>s</sub>, e<sub>g</sub>] の辺を削除する

### 2.3.2 操作例

ここでは単純な例を二つ上げ、他の例については付録にて示す。

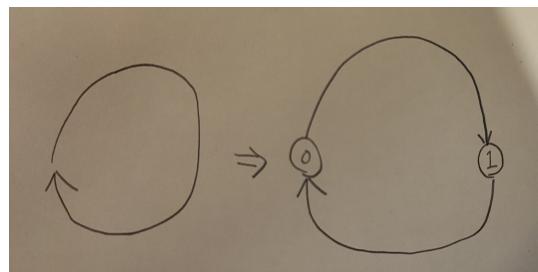


図 8: step0

#### 2.3.2.1 step0 → step1 の例 $[(0, 1, N, N), (1, 0, N, N)]$

区間  $[0, 1]$  を対象に  $S^+$  を行う。

1. Odd 頂点 2 を追加。
2. a=0, P<sub>a</sub>=N, b=1, P<sub>b</sub>=N, c=1, P<sub>c</sub>=N, d=0, P<sub>d</sub>=N として、頂点 2 と繋がる辺を追加。 (0, 2, N, A), (1, 2, N, B), (2, 1, B, N), (2, 0, A, N) の 4 辺。
3. (0, 1, N, N), (1, 0, N, N) を消去。

最終的に残る辺は  $[(0, 2, N, A), (1, 2, N, B), (2, 1, B, N), (2, 0, A, N)]$  の 4 辺。

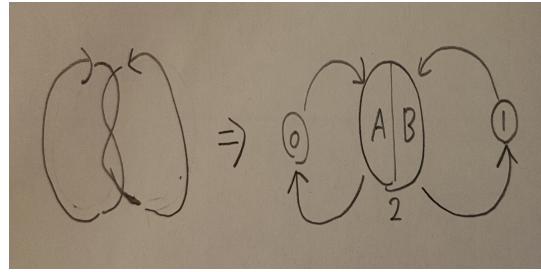


図 9: step1

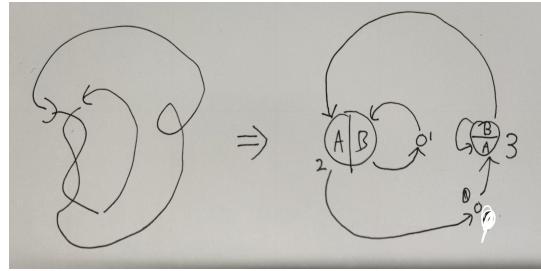


図 10: step1(RI<sup>+</sup>)

**2.3.2.2 step1 →step2-2 の例** 初めに、step1 に RI<sup>+</sup> を行い、Odd 頂点 3 が追加された図 10 の状態にする。

$[(1, 2, N, B), (2, 0, A, N), (0, 3, N, A), (3, 3, B, A), (3, 2, B, A), (2, 1, B, N)]$   
区間  $[0, 3]$  を対象に  $S^+$  を行う。

1. Odd 頂点 4 を追加

2.  $a=1, P_a=N, b=2, P_b=B, c=3, P_c=B, d=3, P_d=A$  として、頂点 3 と繋がる 4 辺を追加。 $(1, 4, N, A), (2, 4, B, B), (4, 3, B, B), (4, 3, A, A)$  の 4 辺。
3.  $(1, 2, N, B), (3, 3, B, A)$  を消去。
4.  $(2, 0, A, N), (0, 3, N, A)$  を反転して  $(0, 2, N, A), (3, 0, A, N)$  にする。

$[(3, 2, B, A), (2, 1, B, N), (1, 4, N, A), (2, 4, B, B), (4, 3, B, B), (4, 3, A, A), (0, 2, N, A), (3, 0, A, N)]$

頂点 3, 4 は Odd+Odd→Even に統合でき、統合結果の Even 頂点 5 の追加と空頂点の削除をすると図 11 の通りになる。

$[(2, 5, B, B), (5, 2, B, A), (2, 5, B, A), (5, 2, A, A)]$

### 2.3.3 計算量

ある KEGにおいて、任意の 2 辺に対する  $S^+$  の結果を得るのに必要な計算量を考えていく。

1. 頂点追加  $\rightarrow O(1)$

