

KEG上におけるクロスキャップ数の計算

Kaito Yamada

2022-11-06

概要

目次

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

1 はじめに

1.1 研究背景

1.1.1 S^+ のアルゴリズムとは

1.1.2 KEG とは

1.2 先行研究

1.2.1 $C(K) = 3$ の場合

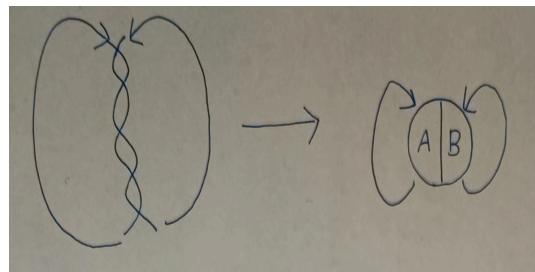
1.2.2 KEG の同値判定

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。

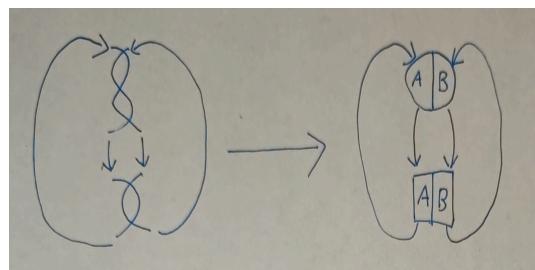


2.1.1 頂点の分割のアルゴリズム

頂点内の紐に対して S^+ を適用したいので、一つの頂点を明示的に分割し、間を繋ぐ 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）

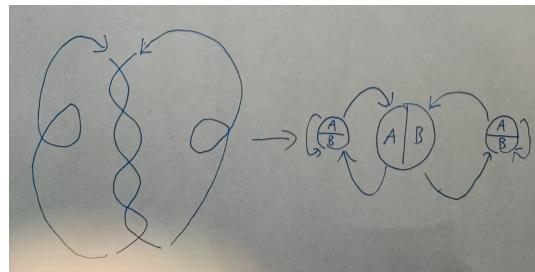
頂点は、それぞれ以下のように分割できる

- Odd \rightarrow Odd + Even / Even + Odd
- Even \rightarrow Odd + Odd / Even + Even



2.1.2 RI⁺ 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⁺ を表現する
(u, v, Tu, Tv) 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, Tu, A),
(O, O, B, A), (O, v, B, Tv) という辺を追加する。



2.1.3 計算量

2.2 S⁺ の対象の列挙

2.2.1 列挙のアルゴリズム

S⁺ 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⁺ 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_e を求める。C_e の長さはグラフの辺の総数と等しいので、E。
2. C_e を 2 個連結する。
3. i=[0, E-1] をループし、e_s=C_e[i] とする。j=[1, E-1] とし、e_g=C_e[i+j] とする。

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

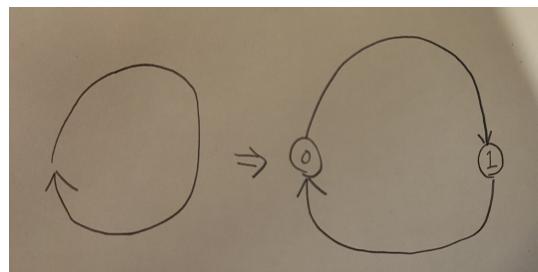
KEG を対象とした S^+ のアルゴリズムはいかに示す通りである。

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, T_a, A), (b, O, T_b, B), (O, c, B, T_c), (O, d, A, T_d) の 4 辺を追加する。
3. C_e で e_s と e_g 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, T_u, T_v) を (v, u, T_v, T_u) にする。
4. [e_s, e_g] 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。

2.3.2.1 step0 → step1 の例

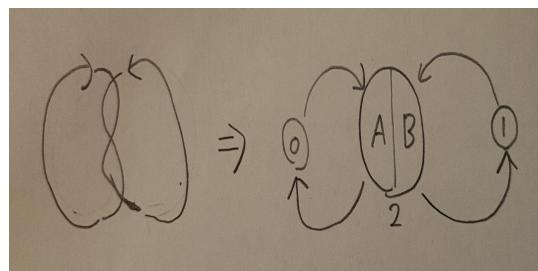


$[(0, 1, N, N), (1, 0, N, N)]$

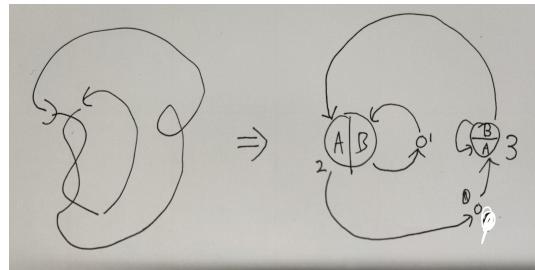
区間 $[0, 1]$ を対象に S^+ を行う。

1. Odd 頂点 2 を追加。
2. a=0, T_a=N, b=1, T_b=N, c=1, T_c=N, d=0, T_d=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 边。



2.3.2.2 step1 → step2-2 の例 step1 に RI⁺ を行い、Odd 頂点 3 が追加され以下の状態になる。



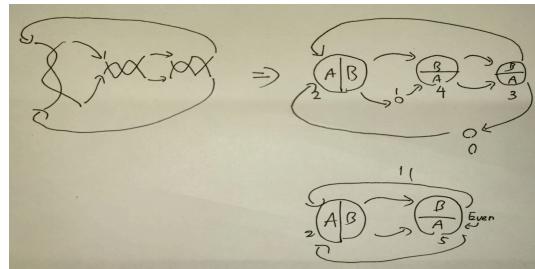
$[(1, 2, N, B), (2, 0, A, N), (0, 3, N, A), (3, 3, B, A), (3, 2, B, A), (2, 1, B, N)]$
区間 $[0, 3]$ を対象に S^+ を行う。

1. Odd 頂点 4 を追加
2. $a=1, T_a=N, b=2, T_b=B, c=3, T_c=B, d=3, T_d=A$ 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、Even 頂点 5 の追加と空頂点の削除をすると以下の通りになる。

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



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 $\rightarrow O(1)$
2. Odd 頂点への 4 辺の追加 \rightarrow 定数なので $O(1)$
3. e_s と e_g の間の辺を逆にする \rightarrow 最大 $E-2$ 辺に行われる所以 $O(E)$

4. 逆転前の辺を削除する → 最大 E-2 辺に行われる所以 $O(E)$

2.2 節で列挙したペアのそれぞれについて S^+ を適用するため、各計算ステップの最大次数を見ると E^2 回のループで $O(E)$ の操作をするためトータルで $O(E^3)$

3 まとめ

謝辞

本論文を作成するにあたり、— みなさまに感謝の意を表します。

参考文献

- [1] Noboru Ito and Yusuke Takimura. Crosscap number three alternating knots. *J. Knot Theory Ramifications*, 31(4):2250026–1–2250026–11, 6 2022.
 - [2] Noboru Ito and Kaito Yamada. Plumbing and computation of crosscap number. *JP Journal of Geometry and Topology*, 26(2):103–115, 11 2021.