

キャタピラの列挙

PD 菊地 洋右

情報理工学系研究科数理情報学専攻

概要

データベースなどに代表されるデータの集合に対し、そのデータの生成、抽出に必要となるツールとして列挙アルゴリズムがある。現在、様々な対象に対して列挙アルゴリズムが提案されている。ここではキャタピラというグラフに対して直径を考慮するという手法で効率的な列挙を行う。

1 はじめに

グラフの個数の数え上げには様々な研究がある [2, 3, 6, 9], 一方、それらのグラフのすべてを実際に計算機等で出力することは、グラフの個数が一般に膨大であるため困難であった。しかし、近年の計算機の性能の向上に伴い、計算機による列挙が可能になり、様々なアルゴリズムも提案されてきた [1, 5, 8]。ただし、先にも述べたように列挙する対象の個数は膨大であることに変わりはない。例えば今回、扱ったキャタピラについて言えば頂点数が $n+4$ であるキャタピラの個数は、 $2^n + 2^{\lfloor n/2 \rfloor}$ であることが知られている [6]。そこでパラメータの十分大きな値に対しても、列挙を行える効率的な列挙が必要となる。ここで、重複なく、かつ、もれなく、指定されたクラスのグラフを生成することを、指定されたクラスのグラフの列挙と呼ぶ。本研究は、木の主要な部分クラスであるキャタピラの列挙に関するものである [7]。

これまでに、様々な木の列挙に関する研究がある。例えば、根付き木の列挙アルゴリズム [1] や、平面植木の列挙アルゴリズム [8] などである。また、応用においては分子構造中のパスを発見する研究がある [4]。

本研究では、まず、頂点の個数が n であり、直径が d であるキャタピラを列挙するアルゴリズムを設計した。このアルゴリズムを用いることで、頂点数が n であるキャタピラを列挙するアルゴリズム、頂点数が n である全てのキャタピラの、全ての平面埋め込みを列挙するアルゴリズムを設計した。

2 列挙の概要

キャタピラを列挙するにあたり、まずキャタピラを一意に表わす文字列を考える必要がある。ここではキャタピラの直径を与える頂点に着目した。直径というパラメータを導入し、頂点数が n , 直径が d のキャタピラの列挙アルゴリズムを設計し、このアルゴリズムをサブルーチンとして用いることで頂点数 n のキャタピラの列挙が効率的に行えた。直径を与える頂点は一意には決まらないが、直径を与える頂点を選び固定すると、直径を与える頂点の次数列は対称なものを同一視すれば一意に表わすことができる。そこで、この直径を与える頂点の次数列に辞書式順を導入し一意に表現する文字列を導入した。このキャタピラを表現する文字列はキャタピラの頂点の個数が n , 直径が d であるとき $d-1$ 桁で、各桁の値の和が $n-d+1$ であるような列の部分集合である。そこで、 $d-1$ 桁で各桁の値の和が $n-d+1$ であるような列を辞書式順に生成して、キャタピラを表現する文字列だけを出力するアルゴリズムを設計した。単純に辞書式順に生成した場合、キャタピラを表現する文字列かどうかの判定が計算時間に影響する。そのためアルゴリズムにおいて文字列がキャタピラを表現するかどうかを判定するためいくつかの工夫をした。その工夫の一つとしてこの列挙に適したデータ構造の提案をした。

3 効率化のための工夫

$d-1$ 桁の $n-d$ 進数と見なせる文字列で, 各桁の値の和が $n-d-1$ であるもののすべてを次々に生成する. この生成には, 簡易な方法を用いると文字列 1 つあたり $O(d)$ 時間かかる. また, キャタピラを表現するかどうかを判定し出力するとき判定にも, 簡易な方法を用いると文字列 1 つあたり $O(d)$ 時間かかる.

この判定を文字列 1 つあたり $O(1)$ 時間で行なうことができるデータ構造を提案した. このデータ構造は提案したアルゴリズムにおいて文字列 1 個あたり定数時間で更新できる. さらに, このデータ構造を用いることにより $O(1)$ 時間で, キャタピラを表現するかどうかを判定ができる.

このデータ構造により提案したアルゴリズムの作業用計算領域は $O(d)$, 計算時間は $O(g(n, d))$ となる. ここで, $g(n, d)$ は頂点の個数が n , 直径が d であるようなキャタピラの個数である. すなわち, アルゴリズムはキャタピラを 1 個あたり平均 $O(1)$ 時間で生成する.

さらに, アルゴリズム中で逆文字列が連続する場合高々一つの逆文字列を生成するだけで, 連続する逆文字列全て生成することのないようにした.

この工夫により, 作業用計算領域は $O(d)$ であり, $O(d)$ 時間の前処理の後, $O(1)$ 時間ごとに 1 つのキャタピラを生成するアルゴリズムが得られた.

4 まとめ

本研究では, 頂点の個数が n であり, 直径が d であるすべてのキャタピラを, 重複なく, かつ, もれなく高速に列挙するアルゴリズムを与えた. 本論文のアルゴリズムは, そのようなキャタピラを 1 個あたり, 定数時間で生成する. また, 本研究の発表を第 16 回 回路とシステム (軽井沢) ワークショップで行い, 研究奨励賞を受賞した.

現在はこの研究と関連したグレイコードの研究を行っている.

今後の方向としては, (1) この研究をもとにした直径が k の木を高速に列挙するアルゴリズムの開発. (2) [4] の研究と結びつけデータ抽出への応用が考えられる.

参考文献

- [1] Beyer, T. and Hedetniemi, S. M., Constant Time Generation of Rooted Trees, *SIAM J. Comput.*, 9 (1980) 706–712.
- [2] Biggs, N. L., Lloyd, E. K., Willson, R. J., *Graph Theory 1736-1936*, Clarendon Press, Oxford, (1976)
- [3] Chauve, C., Dulucq, S. and Rechnitzer, A., Enumerating Alternating Trees, *Journal of Combinatorial Theory, Series A*, 94 (2001) 142–151.
- [4] De Raedt, L. and Kramer, S., The Level-wise Version Space Algorithm and its Application to Molecular Fragment Finding, *Proc. of Int'l. Joint Conf. on Artificial Intelligence*, (2001)853–859.
- [5] Li, Z. and Nakano, S., Listing All Connected Plane Triangulations, *Proc. of Canadian Conference on Computational Geometry* (2001) 121–124.
- [6] Harary, F. and Schwenk, A. J., The Number of Caterpillars, *Discrete Mathematics*, 6 (1973) 359–365.
- [7] Kikuchi, Y., Tanaka, H., Nakano, S. and Shibata, Y., How to Obtain the Complete List of Caterpillars, 9th Annual International Conference, COCOON 2003, Lecture Notes in Comp. Sci., Vol. 2697, pp. 329–338 (2003).
- [8] Nakano, S., Efficient Generation of Plane Trees, *Information Processing Letters*, 84 (2002) 167–172.
- [9] Simone, C. D., Lucertini, M., Pallottino, S. and Simeone, B., Fair Dissections of Spiders, Worms, and Caterpillars, *Networks*, 20 (1990) 323–344.