Pattern Avoidance and Pattern Graphs

Christian Bean

1 Introduction

Definition 1.1. A length k *pattern*, p, is a permutation of length k. An *occurrence* of p in a length n permutation π is a set of indices $\{i_1, i_2, i_3\}$ with $1 \leq i_1 < i_2 < i_3 \leq n$ such that $\pi(i_1)\pi(i_2)\pi(i_3)$ is order equivalent to p. For example an occurrence of 132 would be a set of indices such that $\pi(i_1) < \pi(i_3) < \pi(i_2)$. We say that π *avoids* p if there is no occurrence of p in π.

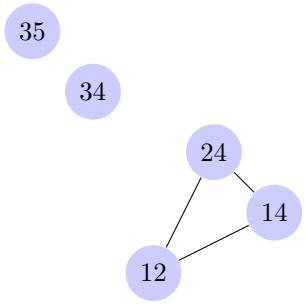
One of the most important questions in the theory of pattern avoidance is understanding the structure of permutation sets. A famous example is that the permutations which are stack-sortable are precisely those that avoid 231, as shown by Knuth [1]. We present here a novel way to associate a graph to the occurrences of a pattern in a given permutation. This approach leads to connections between the properties of graphs and the structure of permutation sets.

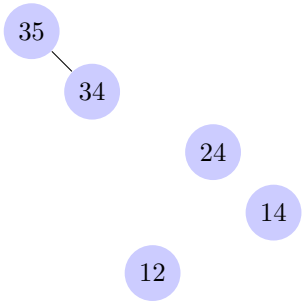
Definition 1.2. Given a permutation π and two patterns p and q, we define the *pattern graph* $G_p^q(\pi)$ to be the graph with

(i) vertices as the occurrences of p in π,

(ii) an edge between v_1 and v_2 if and only if the corresponding occurrences of p are both contained in any one occurrence of q.

For example, let $\pi = 35124$ and $p = 12$. Then the graph on the left is $G_p^{123}(\pi)$ and the one on the right is $G_p^{132}(\pi)$.





2 Problems

Problem 2.1. The graph $G_{12}^{123}(\pi)$ is bipartite if and only if π avoids 123.

Problem 2.2. The graph $G_{12}^{132}(\pi)$ is bipartite if and only if π avoids the pattern 1432.

Problem 2.3. The graph $G_{12}^{123}(\pi)$ is a clique if and only if it empty, a single vertex or a triangle.

Problem 2.4. The graph $G_{12}^{132}(\pi)$ is a clique if and only if π avoids 123, 213 and 3412.

There is Sage code that can be used to generate these graphs for any patterns p and q available on GitHub [2]. This will allow to generate more problems and more pattern avoidance related sets to work with.

References

[1] D. E. Knuth. *The Art of Computer Programming*, Volume 3. Addison-Wesley, 1973.

[2] H. Ulfarsson, *Pattern Graphs*, GitHub repository, 2015, https://github.com/ulfarsson/pattern-graphs.