Discrete Applied Mathematics 224 (2017) 16–44

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

An algorithm computing combinatorial specifications of permutation classes

Frédérique Bassino [a,*], Mathilde Bouvel [b], Adeline Pierrot [c], Carine Pivoteau [d], Dominique Rossin [e]

[a] Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS UMR 7030, F-93430 Villetaneuse, France
[b] Institut für Mathematik, Universität Zürich, Winterthurerstr. 190, CH-8057 Zürich, Switzerland
[c] LRI, Université Paris-Sud, CNRS, Université Paris Saclay, Bat. 650 Ada Lovelace, 91405 Orsay Cedex, France
[d] LIGM UMR 8049, Université Paris Est and CNRS, Université Paris Saclay, Marne-la-Vallée, France
[e] LIX UMR 7161, École Polytechnique and CNRS, Palaiseau, France

ARTICLE INFO

Article history:
Received 3 September 2015
Received in revised form 14 February 2017
Accepted 16 February 2017
Available online 23 March 2017

Keywords:
Pattern avoidance
Permutation class
Combinatorial specification
Generating function
Random sampler

ABSTRACT

This article presents a methodology that automatically derives a combinatorial specification for a permutation class $\mathcal{C}$, given its basis B of excluded patterns and the set of simple permutations in $\mathcal{C}$, when these sets are both finite. This is achieved considering both pattern avoidance and pattern containment constraints in permutations. The obtained specification yields a system of equations satisfied by the generating function of $\mathcal{C}$, this system being always positive and algebraic. It also yields a uniform random sampler of permutations in $\mathcal{C}$. The method presented is fully algorithmic.

1. Introduction

Permutation classes (and the underlying pattern order on permutations) were defined in the seventies, and since then the enumeration of specific permutation classes (*i.e.*, sets of permutations closed under taking patterns) has received a lot of attention. In this context, as in many in combinatorics, a recursive description of the permutations belonging to the class is often the key towards their enumeration. This recursive description is *a priori* specific to the class studied. But more recently, the substitution decomposition (along with other general frameworks, see [30, and references therein]) has been introduced for the study of permutation classes: it provides a general and systematic approach to their study, with a recursive point of view. This tool has already proved useful in solving many enumerative problems [1,3,5,8, among others], but also in other areas like algorithmics [12–14].

The goal of the current paper is to systematize even more the use of substitution decomposition for describing recursively and enumerating permutation classes. Our main result is an algorithm that computes a combinatorial specification (in the sense of Flajolet and Sedgewick [19]) for any permutation class containing finitely many simple permutations. Note that this problem has been addressed already in [1,16], however with much less focus on the algorithmic side. Moreover, we introduce in this article a generalization of permutation classes that we call restrictions: while every permutation class is characterized

* Corresponding author.
E-mail addresses: bassino@lipn.univ-paris13.fr (F. Bassino), mathilde.bouvel@math.uzh.ch (M. Bouvel), adeline.pierrot@lri.fr (A. Pierrot), pivoteau@univ-mlv.fr (C. Pivoteau), dominique.rossin@lix.polytechnique.fr (D. Rossin).

http://dx.doi.org/10.1016/j.dam.2017.02.013
0166-218X/

by a set of forbidden patterns, a restriction is described giving a set of forbidden patterns and a set of mandatory patterns. Our algorithm also allows to compute a specification for restrictions containing finitely many simple permutations.

The article is organized as follows. We start by recalling the necessary background in Section 2: permutation classes, substitution decomposition, and the symbolic method. Section 3 gives a more detailed presentation of our results. Here, we dedicate specific attention to explaining the differences between our work and those of [1,16], and to putting our result in a more global algorithmic context (namely, we describe an algorithmic chain from the basis B of a class $\mathcal{C}$ to random sampling of permutations in $\mathcal{C}$). With the next sections, we enter the technical part of our work. After briefly solving the case of substitution-closed classes in Section 4, we explain in two steps how to obtain a combinatorial specification for other classes $\mathcal{C}$. Section 5 gives an algorithm producing an ambiguous system of combinatorial equations describing $\mathcal{C}$. Next, Section 6 describes how to adapt this algorithm to obtain a combinatorial specification for $\mathcal{C}$. Finally, Section 7 illustrates the whole process on examples.

2. Some background on permutations and combinatorial specifications

2.1. Permutation patterns and permutation classes

A permutation σ of size $|\sigma| = n$ is a bijective map from $[1..n] = \{1, \dots, n\}$ to itself. We represent a permutation by a word $\sigma = \sigma_1\sigma_2 \dots \sigma_n$, where each letter σ_i denotes the image of i under σ. We denote ε the only permutation of size 0; ε is also called the empty permutation.

Definition 2.1. For any sequence s of k distinct integers, the *normalization* of s is the permutation π of size k which is order-isomorphic to s, *i.e.*, $s_\ell < s_m$ whenever $\pi_\ell < \pi_m$.

For any permutation σ of size n, and any subset $I = \{i_1, \dots, i_k\}$ of $\{1, \dots, n\}$ with $i_1 < \cdots < i_k$, σ_I denotes the permutation of size k obtained by normalization of the sequence $\sigma_{i_1} \dots \sigma_{i_k}$.

Definition 2.2. A permutation π is a *pattern* of a permutation σ if and only if there exists a subset I of $\{1, \dots, |\sigma|\}$ such that $\sigma_I = \pi$. We also say that σ *contains* or *involves* π, and we write $\pi \preccurlyeq \sigma$. A permutation σ that does not contain π as a pattern is said to *avoid* π.

Example 2.3. The permutation $\sigma = 316\,452$ contains the pattern 2431 whose occurrences are 3642 and 3652. But σ avoids the pattern 2413 as none of its subsequences of length 4 is order-isomorphic to 2413.

The pattern containment relation $\preccurlyeq$ is a partial order on permutations, and permutation classes are downsets under this order. In other words:

Definition 2.4. A set $\mathcal{C}$ of permutations is a *permutation class* if and only if for any $\sigma \in \mathcal{C}$, if $\pi \preccurlyeq \sigma$, then we also have $\pi \in \mathcal{C}$.

Throughout this article, we take the convention that a permutation class only contains permutations of size $n \geq 1$, *i.e.*, $\varepsilon \notin \mathcal{C}$ for any permutation class $\mathcal{C}$.

Every permutation class $\mathcal{C}$ can be characterized by a unique antichain B (*i.e.*, a unique set of pairwise incomparable elements) such that a permutation σ belongs to $\mathcal{C}$ if and only if it avoids every pattern in B (see for example [1]). The antichain B is called the *basis* of $\mathcal{C}$, and we write $\mathcal{C} = Av(B)$. The basis of a class $\mathcal{C}$ may be finite or infinite; it is described as the permutations that do not belong to $\mathcal{C}$ and that are minimal in the sense of $\preccurlyeq$ for this criterion.

2.2. Simple permutations and substitution decomposition of permutations

The description of permutations in the framework of constructible structures (see Section 2.3) that will be used in this article relies on the substitution decomposition of permutations. Substitution decomposition is a general method, adapted to various families of discrete objects [25], that is based on core items and relations, and in which every object can be recursively decomposed into core objects using relations. In the case of permutations, the core elements are simple permutations and the relations are substitutions.

Definition 2.5. An *interval* of a permutation σ of size n is a non-empty subset $\{i, \dots, (i+\ell-1)\}$ of consecutive integers of $\{1, \dots, n\}$ whose images by σ also form a set of consecutive integers. The *trivial* intervals of σ are $\{1\}, \dots, \{n\}$ and $\{1, \dots, n\}$. The other intervals of σ are called *proper*.

Definition 2.6. A *block* (resp. *normalized block*) of a permutation σ is any sequence $\sigma_{i_1} \dots \sigma_{i_m}$ (resp. any permutation σ_I) for $I = \{i_1, \dots, i_m\}$ an interval of σ.

Definition 2.7. A permutation σ is *simple* when it is of size at least 4 and it contains no interval, except the trivial ones.

Note that no permutation of size 3 has only trivial intervals (so that the condition on the size is equivalent to "at least 3").