Preserving the number of cycles of length k in a growing uniform permutation

Philippe Duchon
LaBRI
Université de Bordeaux
Talence, France
philippe.duchon@labri.fr

Romaric Duvignau[*]
LIF
Aix Marseille Université
Marseille, France
romaric.duvignau@lif.univ-mrs.fr

Submitted: Mar 17, 2016; Accepted: Oct 26, 2016; Published: Nov 10, 2016
Mathematics Subject Classifications: 05A05, 68Q87

Abstract

The goal of this work is to describe a uniform generation tree for permutations which preserves the number of k-cycles between any permutation (except for a *small* unavoidable subset of optimal size) of the tree and its direct children. Moreover, the tree we describe has the property that if the number of k-cycles does not change during any k consecutive levels, then any further random descent will always yield permutations with that same number of k-cycles. This specific additional property yields interesting applications for exact sampling. We describe a new random generation algorithm for permutations with a fixed number of k-cycles in $n + \mathcal{O}(1)$ expected calls to a random integer sampler. Another application is a combinatorial algorithm for exact sampling from the Poisson distribution with parameter $1/k$.

Keywords: random permutations; k-cycles in permutations; generation tree; random generation

1 Introduction

1.1 Overview of results

Throughout the paper, $k \geqslant 1$ is a predefined constant representing some fixed length of cycles in permutations.

It is a well-known[1] fact, that given an infinite vector $(\sigma_n)_{n\geqslant 1}$ such that σ_n is a permutation of n elements chosen uniformly at random, $c_k(\sigma_n)$, the number of cycles of

[*] This work was done while the second author was affiliated with LaBRI, Université de Bordeaux.

[1] see for example, [5] and the references therein, or [2] for total variation distance bounds.

length k in σ_n, converges in distribution to a random Poisson distributed variable X_k with expectation $1/k$, *i.e.*,

$$\mathbb{P}(c_k(\sigma_n) = \ell) \to \frac{e^{-1/k}}{k^\ell l!}, \quad \text{as } n \to \infty.$$

It is a standard result of probability theory that convergence in distribution can always be realized as almost sure convergence. In our setting, this means one can construct a probability distribution μ on *sequences* of permutations (whose n-th element σ_n is always uniform among permutations of size n) such that, for μ-almost all sequences, $c_k(\sigma_n)$ converges to a Poisson random variable with parameter $1/k$. Since we are dealing with integer valued sequences, convergence is equivalent to the sequence being ultimately constant. The main contribution of this paper is the explicit combinatorial description of one such probability distribution.

Since permutations have the very particular property that the number of permutations of each size is a multiple of the number of permutations of the preceding size, a particular type of probability distributions on size-indexed sequences of permutations is given by so-called *generation trees*: infinite trees with the unique permutation of size 1 as the root, where each permutation of size n has exactly $n+1$ children, each a permutation of size $n+1$, and such that each permutation of size n appears exactly once among the nodes of level n. The corresponding probability distribution on sequences corresponds to a *random descent* in the tree, where, at each step, one of the children is picked uniformly at random, independently of the previous choices. Because each permutation appears only once in the tree, the obtained sequences of permutations are very particular; any permutation in the sequence uniquely determines all previous permutations. Still, such sequences are sufficient for our purpose, so that we solely concentrate on the description of one such generation tree.

Our tree construction also exhibits a property that can be described in standard probabilistic terminology. Not only do μ-almost all permutation sequences ultimately stabilize their number of k-cycles; we explicitly describe a subclass of permutations (called non-special permutations) such that μ-almost all sequences contain such a permutation, and that all permutations that follow a non-special permutation (that is, its descendents in the tree) are also non-special, and have the same number of k-cycles. In probabilistic terms, the stabilization time T for the number of k-cycles is not a stopping time, but is upper bounded by a stopping time T', which is none other than the index of the first non-special permutation, and for which we prove that $T \leqslant T' \leqslant T + k$ holds a.s.

One direct consequence is that $c_k(\sigma_{T'})$ is exactly Poisson distributed with parameter $1/k$; this provides a "purely discrete" algorithm for sampling from this Poisson distribution by only using integers and no floating point computations. It also allows for the sampling from the distribution of the number of k-cycles in a uniform permutation of size n, in constant expected time (independently of n).

This paper is an extensive generalization of a previous work by the authors [6] where a similar construction is given for the case $k = 1$.

1.2 Outline of the paper

The paper is organized as follows. In the next section, we give some notations and briefly discuss generation trees for permutations in general. In Section 3 we describe our generation tree, define a number of operations and prove their properties. The proof of the main theorem is in Section 4. In Section 5, we use the tree to describe new algorithms for random generation and simulation of random variates. We finish the paper with a conclusion outlining further possible applications and directions for future research along the same vein.

2 Uniform generation trees for permutations

2.1 Notations and definitions

We use the notation $[a, b]$ for the set of integers i such that $a \leqslant i \leqslant b$ and $[n] = [1, n] = \{1, \ldots, n\}$ for the set of the first n positive integers. We shall write $\mathcal{S}_V$ for the set of all permutations over the set V (bijections from V to itself). If $\sigma \in S_V$, we set $|\sigma| = |V|$ and call it the *size* of σ. The set $\mathcal{S} = \cup_n \mathcal{S}_{[n]}$ is thus the set of all permutations on initial segments of positive integers. When V is not specified, the phrase "permutation of size n" refers to an element of $\mathcal{S}_{[n]}$.

A *cycle* in a permutation $\sigma \in S_V$ is a minimum nonempty subset V that is closed under the action of σ; we slightly abuse this definition by considering the empty set to be a cycle (of length 0).

A cycle C of length k, abbreviated as a k-cycle, is a cycle such that $|C| = k$. We shall denote by $\mathcal{L}_\sigma(j)$ the length of the cycle containing j in the permutation σ; we further assume $\mathcal{L}_\sigma(j) = 0$ if j is not an element of the permutation σ.

A *uniform generation tree* for permutations can be seen as the description, for each nonnegative integer n, of a bijection ϕ_n from $\mathcal{S}_{[n]} \times [n+1]$ to $\mathcal{S}_{[n+1]}$. The countable set $\mathcal{S}$ can thus be seen as the nodes of an infinite rooted tree, with the unique permutation of size 1 as the root, and where each permutation $\sigma \in \mathcal{S}_{[n]}$ has exactly $n+1$ children, obtained by applying ϕ_n to the pairs (σ, i) for $i \in [n+1]$. The $n!$ nodes at distance $n-1$ from the root are then all permutations of $\mathcal{S}_{[n]}$. Such a generation tree can also be seen as the description of a procedure for the random generation of uniform permutations: from a random permutation of size n and an independent uniform integer in the range $[n+1]$, ϕ_n gives us a uniform random permutation of size $n+1$. Thus, starting from the root and repeating $n-1$ times the simple procedure of moving to a uniformly chosen child of the current node yields a uniform permutation of size n. We refer to this procedure as a *random descent* (possibly infinite) in the tree.

Consistently with the tree terminology, $\phi_n(\sigma, i)$ will be called the *i-th child of* σ, and σ will be called the *parent* of each of its children. All permutations in the subtree rooted at σ are collectively called the *descendants* of σ.

2.2 Some classical generation trees

Many simple combinatorial descriptions of uniform generation trees can be given. We briefly describe two of them.

- Last value insertion: from a permutation $\sigma \in \mathcal{S}_{[n-1]}$ and an integer $i \in [n]$, we obtain a new permutation $\sigma' \in \mathcal{S}_{[n]}$ as follows: set $\sigma'(n) = i$, and for $1 \leqslant j \leqslant n-1$, $\sigma'(j) = \sigma(j)$ if $\sigma(j) < i$, and $\sigma'(j) = 1 + \sigma(j)$ if $\sigma(j) \geqslant i$. In other words, i gives the value of $\sigma'(n)$, and all previous values at least i are shifted up by 1.

- Cycle insertion: this generation scheme is best described by its action on the cycles of the permutation. From a permutation $\sigma \in \mathcal{S}_{[n-1]}$ and an integer $i \in [n]$, we obtain a new permutation $\sigma'' \in \mathcal{S}_{[n]}$ as follows: if $i = n$, simply set $\sigma''(n) = n$ and, for $1 \leqslant j \leqslant n-1$, $\sigma''(j) = \sigma(j)$; otherwise, set $\sigma''(i) = n$, $\sigma''(n) = \sigma(i)$, and $\sigma''(j) = \sigma(j)$ for all other values of j. In other words, n is inserted "right after i" in its pre-existing cycle – or is added as a new fixed point, if $i = n$.

Because last value insertion often shifts many values by 1, it can dramatically change the number of k-cycles – that is, $c_k(\sigma)$ and $c_k(\phi_n(\sigma, i))$ can be very different. For instance, the first child of the permutation $(12 \dots k)(k+1 \dots 2k) \dots (n-k+1 \dots n)$ formed entirely of n/k cycles of length k has no k-cycles (it is cyclic for odd k, and, for even k, it has n/k $(k/2)$-cycles and one $(n/2+1)$-cycle).

On the other hand, the number of k-cycles can only change by ±1 under cycle insertion: it increases by 1 if the selected element is in a $(k-1)$-cycle (or if n is added as a new fixed point for $k = 1$), and decreases by 1 if j was previously in a k-cycle. As one descends in the tree according to the random choices of children, the probability of changing the number of k-cycles becomes arbitrarily small: it is exactly $2/n$ on the n-th step, as on average there is 1 element belonging to each cycle length. One can easily prove, though, that in an infinite random descent into the tree, this change in the number of k-cycles will almost surely happen infinitely often. The property holds for $k = 1$: the last child of any permutation always has one more fixed point; since the series $\sum_n 1/n$ is divergent, the second Borel-Cantelli lemma implies that almost all infinite descents pass through infinitely many last children. The property for general k follows by induction on k: since the expected number of k-cycles in permutations of size $n \geqslant 1$ is exactly $1/k$ (thus finite), the fact that the number of k-cycles almost surely increases infinitely often in a random descent implies that it also decreases infinitely often; and, under cycle insertion, the number of k-cycles decreases exactly when the number of $(k+1)$-cycles increases.

The specific generation tree we describe in this paper has the following properties:

1. For each $n \geqslant 2$ not multiple of k, all permutations at level n have the same number of k-cycles as their parent in the tree.

2. For each $n \geqslant 2$ such that $n = mk$, all but exactly $\frac{(mk)!}{k^m m!} 2^{m-1} = o(n!)$ permutations in $\mathcal{S}_{[n]}$ have the same number of k-cycles as their parent.

3. Whenever a permutation has the same number of k-cycles as its k-th ancestor in the tree, then all its children (and, by immediate induction, all its descendants) also have this property.

4. If a permutation σ does not have the same number of k-cycles as its parent, then the difference is ± 1.

As a consequence, the evolution of the number of k-cycles in an infinite random descent through the tree is quite different from the "cycle insertion" generation tree: with probability 1, the descent will, at some (random) level in the tree, reach a permutation whose number of k-cycles is the same as its k-th antecedent; and, once this happens, this number of k-cycles will remain constant for the rest of the descent.

The first property above is only possible because of a specific property on the distribution of k-cycles in random permutations, which we have not been able to find in previous literature: if $\lfloor n/k \rfloor = \lfloor n'/k \rfloor$, then for any integer ℓ, the proportion of permutations having exactly ℓ k-cycles is the same in $\mathcal{S}_{[n]}$ and in $\mathcal{S}_{[n']}$. We give an elementary proof, using generating functions, of this fact in the next section, and a bijective proof is a byproduct of our tree construction.

The number of permutations having a different number of k-cycles from their parent, as described in the second property above, is also minimum possible, as we shall prove in the next section. Overall, this means that our generation tree is optimal in this sense.

3 The generation tree

The main objective of our generation tree is to preserve, as much as possible, the number of k-cycles between any permutation σ and its direct children in the tree. Our construction often corresponds to the cycle insertion procedure as described in the previous section.

Cycle insertion will change the number of k-cycles exactly when the picked integer i belongs to a $(k-1)$- or k-cycle, so a natural idea when trying to design the generation tree is to use cycle insertion in all other cases (thus obtaining only permutations in which the maximum element is neither in a k- nor a $(k+1)$-cycle), and somehow "fix" the remaining cases.

In these two cases, we will deviate from the cycle insertion construction. When $\mathcal{L}_\sigma(i) = k-1$, we will most often produce a permutation σ' with $\mathcal{L}_{\sigma'}(n) = k+1$ which cannot be obtained otherwise. When $\mathcal{L}_\sigma(i) = k$, we will most often replace i with n in its cycle, then somehow insert i somewhere else, taking care not to create or suppress any k-cycles.

The situation is made a bit more complicated by the requirement for the hereditary property (condition 3 in the previous section).

3.1 Description of the tree

We now give the global description of the tree, that is, describe, for any $\sigma \in \mathcal{S}_{[n]}$ and $i \in [n]$, how to construct the i-th child σ' of σ. The description uses several operations and notations which will only be described later.

Generation-Tree

The root of our tree is the unique permutation of size 1.
Let $n \geqslant 2$, $\sigma \in \mathcal{S}_{[n-1]}$, $i \in [n]$, $\tau = \mathrm{insert}(\sigma, i, n)$ and $(\ell, \zeta) \in [0..n-1] \times \mathcal{S}_{[n]}$ such that:

$$(\ell, \zeta) = \begin{cases} (\mathcal{L}_\sigma(i), \tau) & \text{if } \mathcal{L}_\sigma(i) \neq k, \\ (\mathcal{L}_\sigma(\gamma(\sigma)), \mathrm{insert}(\tau, \sigma^{-1}(\gamma(\sigma)), i)) & \text{if } \mathcal{L}_\sigma(i) = k \text{ and } i < \gamma(\sigma), \\ (0, \mathrm{insert}(\tau, i, i)) & \text{if } \mathcal{L}_\sigma(i) = k \text{ and } i > \gamma(\sigma). \end{cases}$$

In our generation tree, the i-th child σ' of σ is defined in the following fashion:

I. If all other rules do not apply: $\sigma' = \zeta$.

II. If $\ell = k - 1$ and σ is not special: $\sigma' = \mathrm{pop}(\zeta, i)$.

III. If $\ell = 2k - 1$, ζ is special and $k \nmid n$: $\sigma' = \mathrm{shift}^{-1}(\zeta)$.

IV. If $\ell = 2k - 1$, ζ is special and $k \mid n$: $\sigma' = \mathrm{cut}(\zeta)$.

V. If $\ell = 2k$, σ is special, $i > \delta(\sigma)$ and $k \nmid n$:

  (a) If $\mathcal{L}_\sigma(i) = 2k$: $\sigma' = \mathrm{shift}(\zeta)$.

  (b) If $\mathcal{L}_\sigma(i) = k$: $\sigma' = \mathrm{insert}(\zeta, \delta(\zeta, i), i)$.

VI. If $\ell = k - 1$ and σ is special:

  (a) If $\mathcal{L}_\sigma(i) = k - 1$: $\sigma' = \zeta$.

  (b) If $\mathcal{L}_\sigma(i) = k$: $\sigma' = \mathrm{merge}(\zeta, i)$.

In the description of our generation tree, we use the following notations:

- Some permutations will be called *special*; their full definition is postponed to § 3.3.2. Special permutations have only cycles of lengths k and $2k$, with the possible addition of a single *special cycle*, denoted $\mathrm{sc}(\sigma)$, whose length is at most $k - 1$. Non-special permutations have the important property of having only non-special permutations as their children, all of which have the same number of k-cycles as their parent.

- $\mathcal{L}_\sigma(j)$ is the length of the cycle of j in σ, with the convention $\mathcal{L}_\sigma(n) = 0$;

- $\gamma(\sigma)$ is the largest element of σ which is not in a k-cycle, with the convention $\gamma(\sigma) = 0$ if there is no such element;

- when σ is special, $\delta(\sigma)$ is the smallest element of the special cycle of σ: $\delta(\sigma) = \min(\mathrm{sc}(\sigma))$, with the convention $\delta(\sigma) = 0$ if $|\mathrm{sc}(\sigma)| = 0$. Moreover, we set

$$\delta(\sigma, i) = \begin{cases} \delta(\sigma) & \text{if } |\mathrm{sc}(\sigma)| \neq 0, \\ i & \text{if } |\mathrm{sc}(\sigma)| = 0. \end{cases}$$

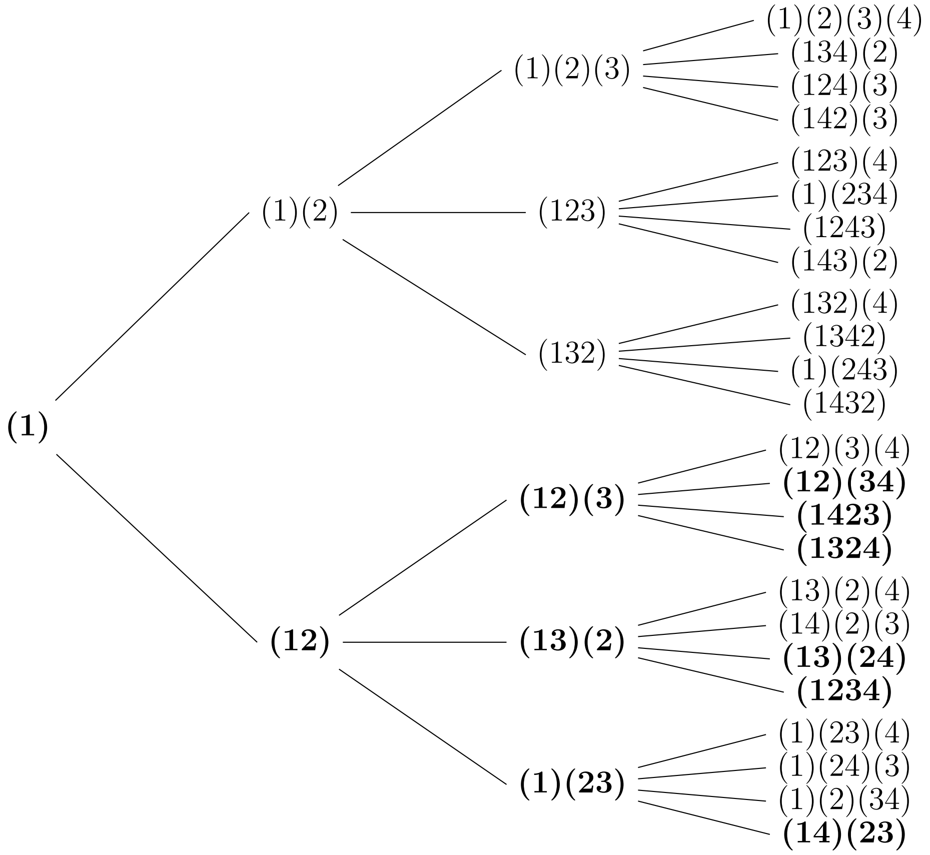


Figure 1: Our generation tree for $k = 2$ up to level 4 with special permutations drawn in bold.

The notation $\tau = \text{insert}(\sigma, a, b)$ corresponds to a variant of cycle insertion: if b is not an element of σ, then it is exactly the result of inserting b after a in σ; if b is an element of σ, then one must first "remove" it from its cycle, *i.e.*, set $\tau(a) = b$, $\tau(b) = \sigma(a)$, $\tau(\sigma^{-1}(b)) = \sigma(b)$ unless $b = \sigma(b)$, and $\tau(j) = \sigma(j)$ for all other j.

The tree description uses some additional operations pop, shift, cut and merge, which will be defined later.

Theorem 1. *The description* Generation-Tree *defines a uniform generation tree for permutations, satisfying the conditions 1, 2, 3 and 4 above.*

The next subsection is devoted to some enumerative considerations. The rest of the section contains the definitions of the missing operators, and the proof of some of their properties.

3.2 Optimum number of changing permutations

In any possible uniform generation tree for permutations, we will call a permutation *changing* if it has a different number of k-cycles from that of its parent. Because the distribution of parameter c_k over $\mathcal{S}_{[n-1]}$ is different from the distribution of c_k over $\mathcal{S}_{[n]}$ when n is a multiple of k, any generation tree will have at least some number of changing permutations at those levels. We now make this quantitatively precise.

We start by proving the fact, mentioned in § 2.2, that the distribution of the c_k statistic over $\mathcal{S}_{[n]}$ only changes when n is a multiple of k.

Let $f_k(n, \ell)$ denote the number of permutations of size n with exactly ℓ k-cycles, and $q_k(n, \ell) = f_k(n, \ell)/n!$ their proportion among permutations of size n. We will call such permutations (n, ℓ)-permutations.

Proposition 2. *For any $n \geqslant 2$ and ℓ, we have*

$$f_k(n, \ell) = \begin{cases} n f_k(n-1, \ell) & \text{if } k \nmid n, \\ n f_k(n-1, \ell) + (-1)^{m-\ell} \frac{(mk)!}{k^m \ell!(m-\ell)!} & \text{if } n = mk. \end{cases}$$

Proof. The statement is equivalent to a reformulation in terms of the q_k's: $q_k(n, \ell) = q_k(n-1, \ell)$ if n is not a multiple of k, and

$$q_k(n, \ell) - q_k(n-1, \ell) = (-1)^{m-\ell} \frac{\binom{m}{\ell}}{k^m m!},$$

if $n = km$.

We start from the generating function for permutations according to size (exponential, counted by variable x) and number of k-cycles (ordinary, counted by variable y). From ([14], Thm. 4.7.2), this generating function is

$$G_k(x, y) = \sum_{n,\ell} q_k(n, \ell) x^n y^\ell = \frac{\exp(\frac{x^k(y-1)}{k})}{1-x}.$$

Multiplying by $1 - x$, we get a new generating function:

$$(1-x) G_k(x, y) = \sum_{n,\ell} \left(q_k(n, \ell) - q_k(n-1, \ell)\right) x^n y^\ell = \exp\left(\frac{x^k(y-1)}{k}\right).$$

This new function is analytic in y and x^k, so the coefficient of x^n is zero unless n is a multiple of k, proving the first case. The second case follows immediately from the Taylor expansion of the exponential function. □

The first case of Proposition 2 shows that it is at least conceivable to look for a generation tree where every permutation of size not multiple of k has the same number of k-cycles as its parent. For n multiple of k, it gives us a lower bound on the total number of changing permutations at level n of any uniform generation tree.

Corollary 3. *In any uniform generation tree, for any $m \geqslant 1$, at least $\frac{(mk)!}{k^m m!} 2^{m-1}$ permutations of level mk are changing permutations.*

Proof. If $n = km$ and $m - \ell$ is even, then by Proposition 2 there are more (n, ℓ)-permutations than children of $(n-1, \ell)$-permutations. Hence at least $f_k(n, \ell) - nf_k(n-1, \ell)$ among the (n, ℓ)-permutations must have a parent with a different number of k-cycles. Summing over the possible ℓ gives the lower bound we are looking for: the number of changing permutations at level $n = mk$ must be at least

$$\frac{(mk)!}{k^m m!} \sum_{\substack{\ell=0 \\ m-\ell \text{ even}}}^{m} \binom{m}{\ell} = \frac{(mk)! 2^{m-1}}{k^m m!}. \qquad \square$$

3.3 Some special families of permutations

Our construction is strongly dependent on a family of permutations that we identify as *special.* These permutations play a key role in our generation tree and are exactly the permutations that have descendants with a different number of k-cycles than themselves.

Special permutations are defined through a smaller family of permutations which we call *critical.*

A last family of permutations, *quasi-special* permutations, plays a minor role in the construction and are described at the end of this section.

3.3.1 Critical permutations

In Proposition 3, we gave an *a priori* lower bound on the number of *changing* permutations, valid for any possible uniform generation tree. We now define a subset of permutations with the appropriate cardinality, which will play this role in our tree.

If σ is a permutation over V and $i \in V$, $\sigma^{[\ell]}(i)$ stands for the set $\{\sigma(i), \cdots, \sigma^{\ell}(i)\}$ of the ℓ first iterated images of i under σ. More generally, we let $\sigma^{[a,b]}(i) = \{\sigma^j(i) \mid a \leqslant j \leqslant b\}$ and in particular $\sigma^{[a,b]}(i) = \emptyset$ if $b < a$, and $\sigma^{[0,0]}(i) = \{\sigma^0(i)\} = \{i\}$.

Definition 4. For any permutation σ over some set V of integers, let $C_1, \ldots, C_m$ be the cycles of length other than k in σ, ordered by their smallest element. Also, let $s_i = \min C_i$ be the minima of these cycles, with the convention that $s_{m+1} = |\sigma| + 1$.

All cycles of length k in σ are considered **critical**.

Among the other cycles, C_i is also said to be **critical** if C_{i-1} is critical (for $i > 1$), $|C_i| = 2k$, and

$$\sigma^k(s_i) = \min \sigma^{[k,2k-1]}(s_i) \cup \{s_{i+1}\}.$$

Definition 5. A permutation σ is **critical** if all its cycles are critical.

Note that critical permutations have only k- and $(2k)$-cycles, with extra conditions on the $(2k)$-cycles.

Permutations of size mk made of m k-cycles are easily enumerated: there are exactly $(mk)!/(m!k^m)$ of them. Note that this is exactly the product of all integers not a multiple of k up to $2mk$.

It is easy to see that critical permutations of size mk with ℓ k-cycles are in bijection with permutations of the same size with exactly m k-cycles, among which an even number

$m - \ell$ are selected. Indeed, merging the selected cycles pairwise into $2k$-cycles (in order of their smallest elements), we obtain a critical permutation (see § 3.6.1 for the precise definition of this bijection). Note that we can easily reverse the procedure and identify which cycles have been chosen in the process.

Thus the number $C(mk, \ell)$ of critical permutations of size mk with ℓ k-cycles (which implies that $m - \ell$ has to be even) is given by

$$C(mk, \ell) = \frac{(mk)!}{k^m m!} \binom{m}{\ell}.$$

Summing over values of ℓ with the appropriate parity, we get the total number $C(mk)$ of critical permutations of size mk as $C(mk) = 2^{m-1} C(mk, 0)$. By definition, $C(n) = 0$ if n is not a multiple of k.

These permutations will be exactly the changing permutations in our generation tree. Note that their number matches the lower bound on the number of such permutations given by Proposition 3.

3.3.2 Special permutations

We now define the set of **special** permutations, which, in our tree, will be exactly those permutations with at least one descendent having a different number of k-cycles. In other words, special permutations will be all ancestors of critical permutations.

Definition 6. A permutation σ is said to be **special** if it has at most one cycle of length strictly less than k (called its *special cycle*, and denoted $\mathrm{sc}(\sigma)$), and the elements not in this cycle form a critical permutation.

Note that for $k = 1$, critical and special permutations are the same.

Contrary to critical permutations, there are special permutations of all sizes n. The length of the special cycle must be $|\mathrm{sc}(\sigma)| = n \bmod k$. However, for any n, all special permutations of size n have numbers of k-cycles of the same parity: for a critical permutation over the elements not in $\mathrm{sc}(\sigma)$ to exist, $\lfloor n/k \rfloor - \ell$ must be even. As for critical permutations, it is not very hard to count the number of special permutations of a given size n.

In order to construct a special (n, ℓ)-permutation σ with $\lfloor n/k \rfloor - \ell$ even, we can first build the cycle $\mathrm{sc}(\sigma)$ of size $h = n \bmod k$: there are $\binom{n}{h}(h-1)!$ possibilities for this cycle if $h > 0$, and obviously 1 possibility if $h = 0$. The rest of the permutation is any critical permutation with ℓ k-cycles over the remaining elements.

Hence the number $S(n, \ell)$ of special permutations of size $n = mk + h$ $(0 \leqslant h < k)$ with ℓ k-cycles, is given by

$$S(n, \ell) = \begin{cases} \frac{(mk)!\binom{m}{\ell}}{k^m m!} & \text{if } h = 0, \\ \binom{mk+h}{h}(h-1)!\frac{(mk)!\binom{m}{\ell}}{k^m m!} & \text{otherwise}. \end{cases}$$

Summing over values of ℓ yields the total number $S(n)$ of special permutations, which we write as a proposition for further reference:

Proposition 7. *For $n = mk + h$ with $0 \leqslant h \leqslant k-1$, the total number $S(n)$ of special permutations of size n is given by*

$$S(n) = \begin{cases} (h-1)! & \textit{if } m = 0, \\ \frac{1}{2}\left(\frac{2}{k}\right)^m \frac{(mk)!}{m!} & \textit{if } h = 0 \ \textit{ and } \ m \geqslant 1, \\ \frac{1}{2h}\left(\frac{2}{k}\right)^m \frac{(mk+h)!}{m!} & \textit{otherwise}. \end{cases}$$

Let us close this section by introducing an additional notation we shall use on special permutations in the following sections.

Definition 8. For any special permutation σ of size n, we let $p_1(\sigma), p_3(\sigma), \ldots, p_{2m-1}(\sigma)$ be the m minima of the m $(2k)$-cycles of σ, and we let $p_{2j}(\sigma) = \sigma^k(p_{2j-1}(\sigma))$ for $j \in [m]$. The elements $p_j(\sigma)$ for $j \in [2m]$ are called the *critical elements* of σ.

3.3.3 Quasi-special permutations

We now define a third family of permutations which we need in order to describe our uniform generation tree when $k \geqslant 2$.

Definition 9. For $k \geqslant 2$, a permutation σ over V is said to be **quasi-special** if:

1. it has one $(k-1)$-cycle D_1 and one $(k+1)$-cycle D_2;
2. the permutation τ formed by the elements of σ belonging neither to D_1 nor to D_2 is *critical*;
3. $\gamma(\sigma)$ is in D_2 and $\min(D_1) < \sigma(\gamma(\sigma))$;
4. if τ has $(2k)$-cycles, its last critical element is smaller than $\min(D_1)$.

For $k = 1$, a permutation is quasi-special if and only if it is special.

Note that no constraints are imposed on the elements of the cycle D_2 other than $\gamma(\sigma)$ and its image; in fact, these elements can be anything between 1 and $\gamma(\sigma) - 1$.

We now compute the number of quasi-special permutations of a given size $n = mk$, when $k \geqslant 2$. Starting from an (mk, m)-permutation σ, select an even nonzero number $m - \ell$ of its cycles, write γ for the maximum element of the selected cycles, and identify, among the selected cycles not containing γ, the one with the largest minimum element; call this element p. We obtain a quasi-special permutation by removing $\sigma(p)$ from its cycle, and inserting it after γ, then merging the remaining selected cycles pairwise in increasing order of their minima. This is bijective; starting from the quasi-special permutation, one can easily recover γ (the maximum of the $(k+1)$-cycle), remove its image and insert it after the minimum of the $(k-1)$-cycle, and split all $(2k)$-cycles into k-cycles.

Hence, the number $Q(mk)$ of quasi-special permutations of size mk is given by

$$Q(mk) = \frac{(mk)!}{k^m m!} \sum_{\substack{\ell=2 \\ \ell \text{ even}}}^{m} \binom{m}{\ell} = \frac{(mk)!}{k^m m!}\left(2^{m-1} - 1\right).$$

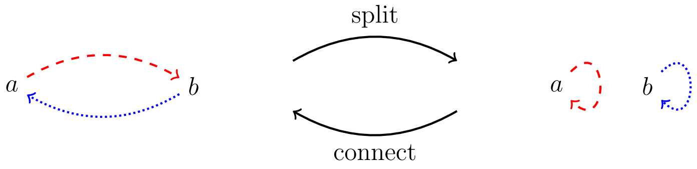


Figure 2: Illustration of split and connect procedures. Dashed and dotted lines represent (potentially empty) portions of a cycle, *i.e.*, iterated images under σ starting from the source of the edge until the target is met.

By the same reasoning, we obtain a formula for the number of quasi-special permutations of size $n = mk$ with exactly ℓ k-cycles ($m - \ell < m$ even),

$$Q(mk, \ell) = \frac{(mk)!}{k^m m!} \binom{m}{\ell}.$$

3.4 Some simple cycle manipulations

As a preliminary to defining the operators on permutations that we need in order to fully describe our generation tree, we now define some simple transformations in terms of cycles.

We have already described the insert operation that we use to add an element to a cycle.

We will have on occasion to *remove* an element from a permutation. This is a partial inverse to the insert operation. When we mention the permutation σ' obtained by removing an element b from its cycle in a permutation $\sigma \in \mathcal{S}_V$, we mean that $\sigma' \in \mathcal{S}_{V \setminus \{b\}}$ is such that $\sigma = \mathrm{insert}(\sigma', \sigma^{-1}(b), b)$. Equivalently, $\sigma' = \mathrm{remove}(\sigma, b)$ is defined as follows:

- if $\sigma(b) \neq b$, $\sigma'(\sigma^{-1}(b)) = \sigma(b)$,
- for all $j \in V \setminus \{\sigma^{-1}(b), b\}$, $\sigma'(j) = \sigma(j)$.

We still need two more operations on permutations that are defined in terms of their cycles: *splitting* a cycle into two cycles, and the opposite operation of *connecting* two different cycles. Both operations correspond to the same composition with the transposition $\tau_{a,b} = (a\ b)$, and the effect (splitting a cycle or connecting cycles) depends only on whether a and b lie inside the same cycle in the original permutation.

If $\sigma \in \mathcal{S}_V$ and a, b are two elements of V, then the permutation $\sigma' = \tau_{a,b} \circ \sigma$ is defined as:

- $\sigma'(\sigma^{-1}(b)) = a$, and $\sigma'(\sigma^{-1}(a)) = b$,
- for all $i \in V \setminus \{\sigma^{-1}(a), \sigma^{-1}(b)\}$, $\sigma'(i) = \sigma(i)$.

If a and b are in the same cycle in σ, then the above operation will result in splitting the cycle into two cycles, one containing a (and its iterated images under σ up to $\sigma^{-1}(b)$) and one containing b (and its iterated images up to $\sigma^{-1}(a)$); to make it clear, we shall use $\mathrm{split}(\sigma, a, b)$ to denote the obtained permutation, with the convention that $\mathrm{split}(\sigma, a, b) = \sigma$ if a and b are already in different cycles. On the other hand, if a and b are in different cycles in σ, this composition results in merging the cycles of a and b into a single one; to make it clear, we will use $\mathrm{connect}(\sigma, a, b)$ in this case, and we set $\mathrm{connect}(\sigma, a, b) = \sigma$ when a and b are in the same cycle in σ.

3.5 Insertion into $(k-1)$-cycles

We now define an operator pop which is the key ingredient of our generation tree. In our construction, it is used to define the i-th child of a permutation σ in most cases where i is in a k- or $(k-1)$-cycle of σ. By itself, it would be a sufficient ingredient to define a generation tree with the "preservation of c_k" property (conditions 1 and 2 of our tree); most of the added complexity is there because we want to ensure the "heredity" property (condition 3).

This operator can also be seen as providing a bijective proof of Proposition 2, as we shall see.

3.5.1 The pop operator

Let $\sigma \in \mathcal{S}_V$ be some permutation, and $i \in V$, such that

- σ is not critical;
- $\mathcal{L}_\sigma(i) = k$;
- $\sigma(i)$ is the largest element of its cycle in σ, and $\sigma(i) > \gamma(\sigma)$.

Under these conditions, we define $\sigma' = \mathrm{pop}(\sigma, i) \in \mathcal{S}_V$ below. The important properties of pop are given by the following proposition.

Proposition 10. pop *defines a bijection between, on the one hand, non-critical permutations σ with a marked element i such that $\sigma(i) > \gamma(\sigma)$, $\sigma(i)$ is the maximum of its cycle, and $\mathcal{L}_\sigma(i) = k$; and, on the other hand, permutations σ' on the same set such that $\mathcal{L}_{\sigma'}(\gamma(\sigma')) = k + 1$ and σ' is not quasi-special. Moreover, whenever $\sigma' = \mathrm{pop}(\sigma, i)$, then $c_k(\sigma') = c_k(\sigma) - 1$.*

Note that the conditions on σ' imply that it is not special: if $k > 1$, the existence of a $(k + 1)$-cycle is sufficient; if $k = 1$, quasi-special and special permutations are the same.

To define pop, we first need to define $p(\sigma)$ for non critical permutation σ, and in some cases, we further define $p'(\sigma)$ and $p''(\sigma)$.

Definition 11. For any non-critical permutation σ, $p(\sigma)$ is the smallest element in a non-critical cycle. Furthermore, if $\mathcal{L}_\sigma(p(\sigma)) > k$, we define $p'(\sigma)$ as the smallest element in a non-critical cycle that is not in $\sigma^{[0..k-2]}(p(\sigma))$ (excluding $p(\sigma)$ in the case $k = 1$), and $p''(\sigma)$ as the second smallest such element.

pop-OPERATOR

Let $(\sigma, i) \in S_V \times V$ be such that σ is not critical, $\mathcal{L}_\sigma(i) = k$, $\sigma(i)$ is the largest element of its cycle and $\sigma(i) > \gamma(\sigma)$. We set $\gamma = \sigma(i)$, $p = p(\sigma)$, $x = \sigma^{k-1}(p)$, $y = \sigma^k(p)$, $\sigma_1 = \mathrm{insert}(\sigma, \gamma, x)$ and $\sigma_2 = \mathrm{insert}(\sigma, \gamma, y)$.
The permutation $\sigma' = \mathrm{pop}(\sigma, i)$ is defined as follows:

1. If $\mathcal{L}_\sigma(p) \neq k+1$, and condition 2 does not apply:
   - (a) If $\sigma(p) = p$: $\sigma' = \mathrm{insert}(\sigma, \gamma, p)$.
   - (b) If $\sigma(p) \neq p$: $\sigma' = \mathrm{insert}(\sigma, \gamma, \sigma^{-1}(p))$.
2. (a) If $\mathcal{L}_\sigma(p) = 2k+1$, $x = p'$ and $y = p''$: $\sigma' = \mathrm{split}(\sigma_1, p, p'')$.
   - (b) If $\mathcal{L}_\sigma(p) = 2k+1$ and $y = p'$: $\sigma' = \mathrm{split}(\sigma_1, p, p')$.
3. If $\mathcal{L}_\sigma(p) = k+1$:
   - (a) If $x = p'$:
     - i. If $y = p''$: $\sigma' = \mathrm{insert}(\sigma_1, p'', p'')$.
     - ii. If $y \neq p''$ and $\mathcal{L}_\sigma(p'') \neq k-1$: $\sigma' = \mathrm{insert}(\sigma_1, p'', y)$.
     - iii. If $\mathcal{L}_\sigma(p'') = k-1$: $\sigma' = \mathrm{connect}(\sigma_2, p, p'')$.
   - (b) If $y = p'$: $\sigma' = \mathrm{insert}(\sigma_1, p', p')$.
   - (c) If $p' \notin \{x, y\}$ and $\mathcal{L}_\sigma(p') \neq k-1$: $\sigma' = \mathrm{insert}(\sigma_1, p', y)$.
   - (d) If $\mathcal{L}_\sigma(p') = k-1$: $\sigma' = \mathrm{connect}(\sigma_2, p, p')$.

Note that under the condition $\mathcal{L}_\sigma(p(\sigma)) > k$, $p'(\sigma)$ and $p''(\sigma)$ are both well defined, since it is assumed that the non critical cycle containing $p(\sigma)$ has at least $k+1$ elements and we only exclude $k-1$ of them. In the case where $k = 1$, $p(\sigma)$ is still excluded, but the cycle not being critical forbids it from being reduced to the 2-cycle $(p(\sigma)\ p'(\sigma))$.

The definition of the operator is given in the description pop-OPERATOR.

All rules are presented in Figure 3. Note that some rules actually define σ' in the same way, and are kept separated for later convenience: 1a is identical to rule 1b in that they define $\sigma' = \mathrm{insert}(\sigma, \gamma, \sigma^{-1}(p))$; rules 2a and 2b both define $\sigma' = \mathrm{split}(\sigma_1, p, y)$; 3(a)i and 3b both define $\sigma' = \mathrm{insert}(\sigma_1, y, y)$. Similarly, 3(a)ii and 3c correspond to the same action applied to p'' for the former, and p' for the latter; this is also true of 3(a)iii and 3d.

Remark 12. Let us consider how this operator works when $k = 1$. In this case, the target set of the operator are non-special permutations σ' where $\gamma(\sigma')$ is in a 2-cycle.

Moreover, we have $x = p$, $y = \sigma(p)$ and $i = \gamma(\sigma)$. Hence some rules are never applied: rule 1a because p cannot be a fixed point, rules 2a and 3a because x cannot be p', rule 3b because p cannot form a 2-cycle with p', and rule 3d because there are no 0-cycles. The three remaining rules (1b, 2b and 3c) are recalled in a simpler form with the description of the tree for $k = 1$ (in Section 4.2) and depicted in Figure 4.

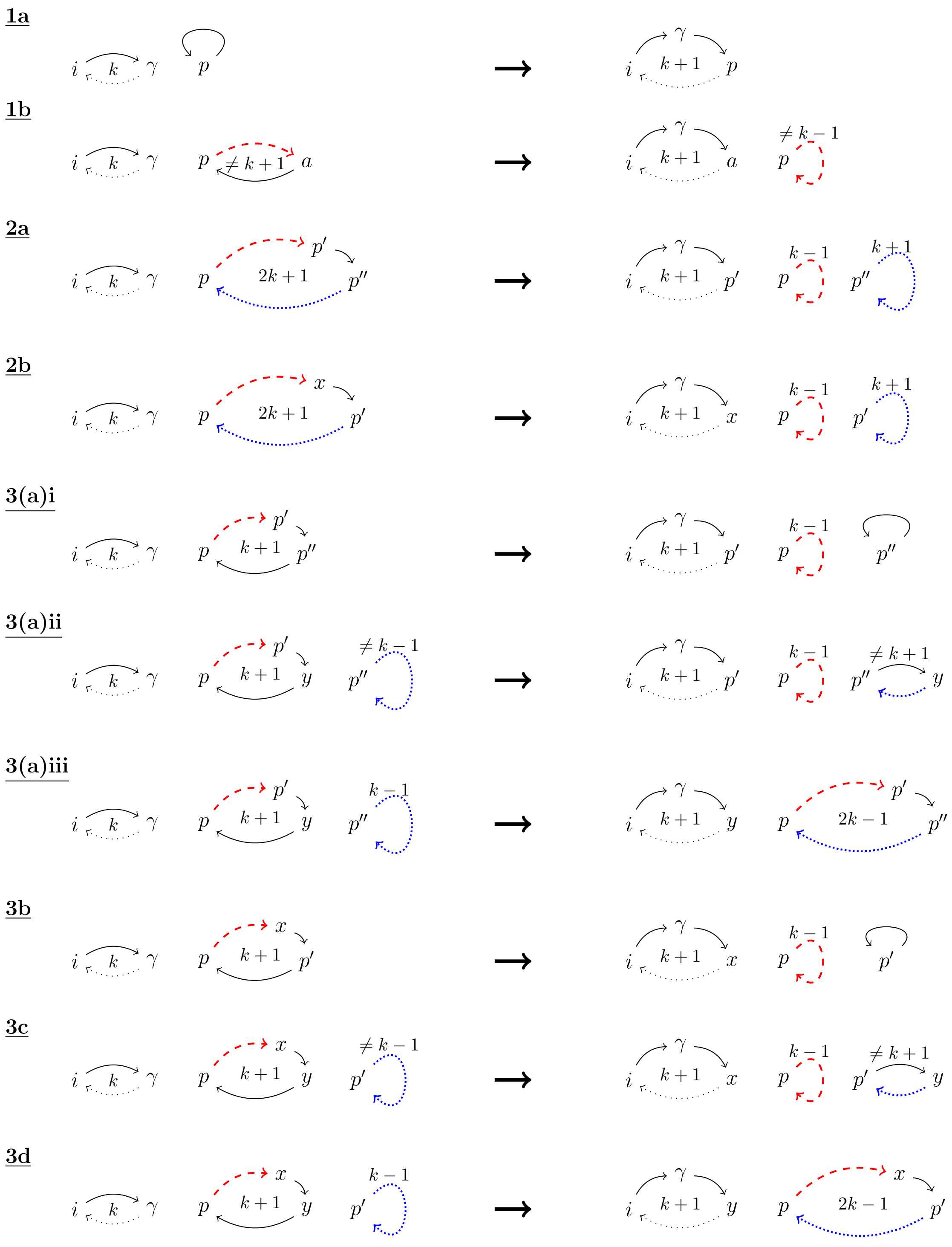

1a
i k γ p → i γ k+1 p
1b
i k γ p ≠k+1 a → i γ k+1 a ≠k−1 p
2a
i k γ p p′ 2k+1 p′′ → i γ k+1 p′ k−1 p k+1 p′′
2b
i k γ p x 2k+1 p′ → i γ k+1 x k−1 p k+1 p′
3(a)i
i k γ p p′ k+1 p′′ → i γ k+1 p′ k−1 p p′′
3(a)ii
i k γ p p′ k+1 y ≠k−1 p′′ → i γ k+1 p′ k−1 p ≠k+1 p′′ y
3(a)iii
i k γ p p′ k+1 y k−1 p′′ → i γ k+1 y p p′ 2k−1 p′′
3b
i k γ p x k+1 p′ → i γ k+1 x k−1 p p′
3c
i k γ p x k+1 y ≠k−1 p′ → i γ k+1 x k−1 p ≠k+1 p′ y
3d
i k γ p x k+1 y k−1 p′ → i γ k+1 y p x 2k−1 p′

Figure 3: Illustration of the different rules of the pop operator for $k \geqslant 2$. Plain lines indicate direct images under σ or σ', dashed and dotted lines indicate portions of cycles. Lengths of cycles are shown as integers inside or over cycles. All cycles are implicitly non critical and hence of length not k. Conditions are not shown (see text).

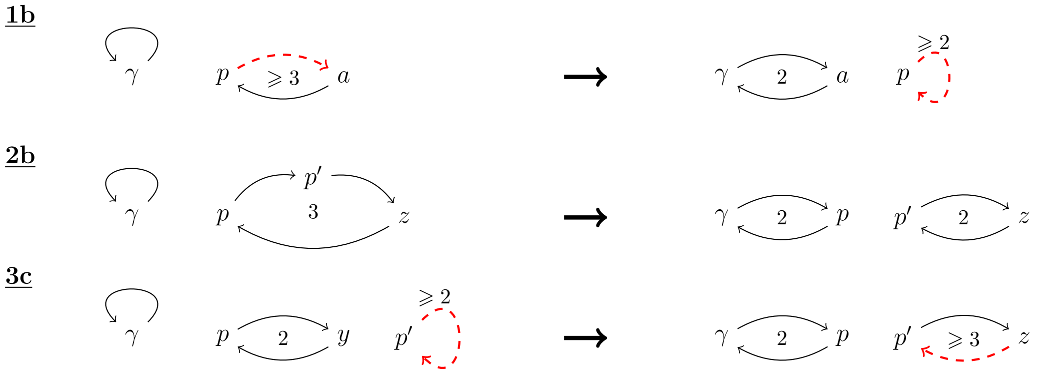


Figure 4: Illustration of pop's rules for $k = 1$.

The proof of Proposition 10 is split in the four following lemmas.

Lemma 13. *For $k = 1$,* pop *is onto.*

Proof. We construct a preimage (σ, i) for any non-quasi-special permutation σ' with $\mathcal{L}_{\sigma'}(\gamma(\sigma')) = 2$. Let $p = p(\sigma')$, $p' = p'(\sigma')$ and $i = \gamma = \gamma(\sigma')$.

We distinguish three cases, building in each case a permutation σ where i is a fixed point. In each case, the other fixed points of σ are exactly those of σ'.

1b) If $p \neq \sigma'(\gamma)$: $\sigma = \text{insert}(\sigma', \sigma'^{-1}(p), \sigma'(\gamma))$. In this situation, p is in a cycle of length at least 2 in σ', hence in a cycle of length at least 3 in σ (hence $p = p(\sigma)$) and $\sigma(p) \neq p'(\sigma)$. Hence rule 1b applies to (σ, i) (rule 2b does not apply) and one easily checks that $\sigma' = \text{pop}(\sigma, i)$.

2b) If $p = \sigma'(\gamma)$ and $\mathcal{L}_{\sigma'}(p') = 2$: $\sigma = \text{insert}(\sigma', \sigma'(p'), p)$. In this case, p is inserted before p' into a 2-cycle, hence is in a 3-cycle in σ. Consequently, $p(\sigma) = p$, and $p'(\sigma) = \sigma(p)$; thus rule 2b applies to (σ, i) and gives σ'.

3c) If $p = \sigma'(\gamma)$ and $\mathcal{L}_{\sigma'}(p') \geqslant 3$: we let $\sigma_1 = \text{insert}(\sigma', p, p)$ and we get the permutation σ as $\sigma = \text{insert}(\sigma_1, p, \sigma'(p'))$; that is, we extract p and $\sigma'(p')$ from their cycles and construct a new 2-cycle with them. We have p in a 2-cycle in σ, but p' is in another cycle, thus we still have $p = p(\sigma)$ and $p' = p'(\sigma)$; rule 3c applies on (σ, i) and produces σ'. □

Lemma 14. *For $k \geqslant 2$,* pop *is onto.*

Proof. We construct a preimage (σ, i) for any non-quasi-special permutation σ' with $\mathcal{L}_{\sigma'}(\gamma(\sigma')) = k + 1$. More precisely, for each possible σ', we identify one of the ten rules of pop and describe a preimage (σ, i) to which said rule applies, and yields σ'. In each case, σ has exactly one more k-cycle than σ'.

Let $\gamma = \gamma(\sigma')$, $i = \sigma'^{-1}(\gamma)$, $a = \sigma'(\gamma)$, C the cycle of γ (also containing i and a). Let furthermore σ'' be the permutation obtained after removing the cycle C from σ' and adding the element a as a fixed point, *i.e.*, $\sigma'' = \text{insert}(\sigma|_{V\setminus C}, a, a)$. We note $\tilde{p} = p(\sigma'')$.

In all the following cases (numbered by the corresponding rule of pop which yields σ'), it should be clear that the cycle containing i (and γ) in σ shall be of size k (hence critical), since the defined permutation σ will always remove a from its cycle (and leave other k-cycles untouched, resulting in σ having exactly one more k-cycle than σ').

Furthermore, all other cycles with a minimum less than $\tilde{p}$ shall not be modified in the construction of σ, hence these cycles shall remain critical in σ. Under these two conditions, we deduce directly that $\tilde{p} = p(\sigma)$ if the cycle of $\tilde{p}$ is not critical in σ. In the following, we will not recall these two "easy" properties of σ.

Let us first deal with two specific cases.

1.a) If $a = \tilde{p}$, we let $\sigma = \text{insert}(\sigma', \tilde{p}, \tilde{p})$.

1.b) If $\mathcal{L}_{\sigma'}(\tilde{p}) < k - 1$, we let $\sigma = \text{insert}(\sigma', \sigma'^{-1}(\tilde{p}), a)$.

In these first two situations, $\tilde{p}$ is in a cycle of length strictly less than k in σ: in the first case, it is a fixed point in σ (and we assumed $k \geqslant 2$), and in the second case, it is in a cycle of length $\mathcal{L}_{\sigma'}(\tilde{p}) + 1 < k$. Thus $\tilde{p}$ is in a non critical cycle in σ, and therefore $\tilde{p} = p(\sigma)$.

In the first case, rule number 1a applies on (σ, i) and yields σ'. In the second case, rule 1b is the only applicable rule, and its application to (σ, i) results in the permutation σ'.

We now move to the remaining cases. From now on, we assume $a \neq \tilde{p}$ and $\mathcal{L}_{\sigma'}(\tilde{p}) \geqslant k - 1$.

Let us further define two distinguished elements in σ': $\tilde{p}'$ and $\tilde{p}''$.

We define $\tilde{p}'$ (resp. $\tilde{p}''$) as the smallest element (resp. second smallest element) in a non-critical cycle in σ'' that does not belong to $\sigma'^{[0,k-2]}(\tilde{p})$. Since σ'' contains a fixed point larger than $\tilde{p}$ (namely, a), $\tilde{p}'$ is always well defined. Moreover, note that if $\tilde{p}' = a$ and no element $\tilde{p}''$ can be found in σ'' (that is, $\tilde{p}'$ is in the largest non critical cycle in σ''), then the permutation σ' is quasi-special (and hence out of the scope of the lemma): indeed, in this situation, $\gamma = \gamma(\sigma')$ is in a $(k+1)$-cycle, $\tilde{p}$ is the minimum of a $(k-1)$-cycle, $\tilde{p}' = \sigma'(\gamma)$ is larger than $\tilde{p}$, and all other cycles are critical in σ''. If $\tilde{p}' \neq a$, at the very least, two elements larger than $\tilde{p}$ are in non critical cycles in σ''. Overall, $\tilde{p}'$ and $\tilde{p}''$ are always well-defined under our current assumptions.

In all remaining cases, we shall have $\sigma'^{[0,k-2]}(\tilde{p}) = \sigma^{[0,k-2]}(\tilde{p})$, thus we get $\tilde{p}' = p'(\sigma)$ and $\tilde{p}'' = p''(\sigma)$ whenever $p'(\sigma)$ and $p''(\sigma)$ are in non critical cycles of σ.

We shall see in each rule that $p(\sigma) = \tilde{p}$, thus, for elements larger than $\tilde{p}$ (which includes $\tilde{p}'$ and $\tilde{p}''$), not being in critical cycles of σ reduces to not being in k-cycles; this is easily checked for the different rules.

Let us now consider all cases where $\mathcal{L}_{\sigma'}(\tilde{p}) > k$.

3.a.iii) If $\mathcal{L}_{\sigma'}(\tilde{p}) = 2k - 1$, $\sigma'^{k-1}(\tilde{p}) = \tilde{p}'$, and $\sigma'^{k}(\tilde{p}) = \tilde{p}''$: we set $\sigma_1 = \text{split}(\sigma', \tilde{p}, \tilde{p}'')$ and $\sigma = \text{insert}(\sigma_1, \tilde{p}, a)$. In this situation, $\tilde{p}$ is in a $(k+1)$-cycle in σ, $\tilde{p}' = \sigma^{k-1}(p(\sigma))$ and $\mathcal{L}_{\sigma}(\tilde{p}'') = k - 1$, thus rule 3(a)iii applies to (σ, i) and produces σ'.

3.d) If $\mathcal{L}_{\sigma'}(\tilde{p}) = 2k-1$ and $\sigma'^k(\tilde{p}) = \tilde{p}'$: $\sigma_1 = \mathrm{split}(\sigma, \tilde{p}, \tilde{p}')$ and $\sigma = \mathrm{insert}(\sigma_1, \sigma_1^{-1}(\tilde{p}), a)$. In this case, $\tilde{p}$ is in a $(k+1)$-cycle and $\tilde{p}'$ in a $(k-1)$-cycle in σ. We have $\sigma' = \mathrm{pop}(\sigma, i)$ by rule 3d.

1.b) In all remaining cases where $\mathcal{L}_{\sigma'}(\tilde{p}) > k$: we let $\sigma = \mathrm{insert}(\sigma', \sigma'^{-1}(\tilde{p}), a)$ and we get $\sigma' = \mathrm{pop}(\sigma, i)$ through rule 1b, by the following discussion.

In this case, we increase the length of the cycle of $\tilde{p}$ by one, and since by hypothesis it was different from $k-1$, this cannot create a k-cycle. Moreover, we shall have $p(\sigma) = \tilde{p}$, as the only possibilities to create a critical cycle of length $2k$ are captured by the previous two cases. Indeed, since we insert a in the second part of the $2k$-cycle, the first part should already satisfy the critical condition in order for the condition to hold for the full cycle. However, this implies either $\tilde{p}' = \sigma'^k(\tilde{p})$ or $\tilde{p}' = \sigma'^{k-1}(\tilde{p})$ and $\tilde{p}'' = \sigma'^k(\tilde{p})$; and both of these situations are already covered by the two previous cases.

Now let us check that rule 1b is the rule that applies to (σ, i), *i.e.*, that rule 2 does not preempt it. By definition of $\tilde{p}$, it is not in a critical $2k$-cycle. Hence, if $\sigma'^k(\tilde{p}) = \tilde{p}'$ or $(\tilde{p}', \tilde{p}'') = (\sigma'^{k-1}(\tilde{p}), \sigma'^k(\tilde{p}))$, the cycle of $\tilde{p}$ must be of size other than $2k$. Increasing such a cycle cannot produce a $(2k+1)$-cycle and thus rule 2 will never be used on σ.

We are now left with the only remaining case: $\mathcal{L}_{\sigma'}(\tilde{p}) = k-1$. In this part, we will often have to insert two elements, one after the other, after the same element in σ'; we use here the notation $\tau' = \mathrm{insert}(\tau, i, a, b)$ as a short form for $\tau' = \mathrm{insert}(\mathrm{insert}(\tau, i, b), i, a)$.

The three next cases cover the case $a = \tilde{p}'$.

3.a.i) If $a = \tilde{p}'$ and $\mathcal{L}_{\sigma'}(\tilde{p}'') = 1$: we set $\sigma = \mathrm{insert}(\sigma', \sigma'^{-1}(\tilde{p}), \tilde{p}', \tilde{p}'')$. In this case, rule 3(a)i applies to (σ, i) and yields σ', since $\tilde{p}$ is in a $(k+1)$-cycle of σ (hence $\tilde{p} = p(\sigma)$) whose last two elements starting from $\tilde{p}$ are $p'(\sigma)$ and $p''(\sigma)$.

3.a.ii) If $a = \tilde{p}'$ and $\mathcal{L}_{\sigma'}(\tilde{p}'') \notin \{1, k+1\}$: we set $\sigma = \mathrm{insert}(\sigma', \sigma'^{-1}(p), \tilde{p}', \sigma'(\tilde{p}''))$. We have then $\tilde{p}$ in a $(k+1)$-cycle in σ (thus $\tilde{p} = p(\sigma)$) with $\sigma^{k-1}(p(\sigma)) = \tilde{p}' = p'(\sigma)$. The cycle of $\tilde{p}''$ is one shorter in σ than in σ'; however, such a cycle cannot be of length k (thus $p''(\sigma) = \tilde{p}''$) nor $k-1$. Hence, we have $\sigma' = \mathrm{pop}(\sigma, i)$ using rule 3(a)ii.

2.a) If $a = \tilde{p}'$ and $\mathcal{L}_{\sigma'}(\tilde{p}'') = k+1$: we set $\sigma_1 = \mathrm{insert}(\sigma', \sigma'^{-1}(\tilde{p}), \tilde{p}')$ and the permutation $\sigma = \mathrm{connect}(\sigma_1, \tilde{p}, \tilde{p}'')$. In this situation, $\tilde{p}$ is in $(2k+1)$-cycle of σ whose elements in positions $k-1$ and k (starting from $\tilde{p}$) are $p'(\sigma)$ and $p''(\sigma)$. Here rule 2a applies to (σ, i) and gives σ'.

Finally, the last three cases cover the remaining situations.

3.b) If $\mathcal{L}_{\sigma'}(\tilde{p}') = 1$: we set $\sigma = \mathrm{insert}(\sigma', \sigma'^{-1}(\tilde{p}), a, \tilde{p}')$. We have $\tilde{p}$ in a $(k+1)$-cycle of σ with $\tilde{p}' = \sigma^{-1}(\tilde{p})$, hence $\tilde{p} = p(\sigma)$ and $\tilde{p}' = p'(\sigma)$. In this situation, rule 3b applies to (σ, i) and produces σ'.

3.c) If $\mathcal{L}_{\sigma'}(\tilde{p}') \notin \{1, k+1\}$: we set $\sigma = \text{insert}(\sigma', \sigma'^{-1}(\tilde{p}), a, \sigma'(\tilde{p}'))$. In this case, $\mathcal{L}_\sigma(\tilde{p}) = k+1$ and $\mathcal{L}_\sigma(\tilde{p}') = \mathcal{L}_{\sigma'}(\tilde{p}') - 1 \notin \{k-1, k\}$. Hence we have $p(\sigma) = \tilde{p}$ and $p'(\sigma) = \tilde{p}'$, and rule 3c applies to (σ, i) and yields σ'.

2.b) If $a \neq \tilde{p}'$ and $\mathcal{L}_{\sigma'}(p') = k+1$: we set $\sigma_1 = \text{insert}(\sigma', \sigma'^{-1}(\tilde{p}), a)$, and the permutation $\sigma = \text{connect}(\sigma_1, \tilde{p}, \tilde{p}')$. The cycle of $\tilde{p}$ in σ is then of length $2k+1$, and $\sigma^k(\tilde{p}) = \tilde{p}' = p'(\sigma)$. Rule 2b applies to (σ, i) and yields σ'. □

To prove that pop is one-to-one, we will need the following lemma, which essentially states that $p(\sigma)$ can be recovered from σ'.

Lemma 15. *Assume $k \geqslant 2$. Let (σ, i) be such that $\sigma' = \text{pop}(\sigma, i)$ is well defined, and let σ'' be obtained by removing the cycle containing i from σ' and adding $a = \sigma'(\gamma(\sigma'))$ as a fixed point. Then $p(\sigma'') = p(\sigma)$.*

Proof. Let $p = p(\sigma)$ and, if they exist, $p' = p'(\sigma)$ and $p'' = p''(\sigma)$; let also $\tilde{p} = p(\sigma'')$.

Analysing the ten rules depicted by Figure 3, we observe that cycles of σ other than those containing i, p, and possibly p' and p'' all appear identically in σ''; furthermore, the elements of the cycle containing i are absent from σ''. Consequently, since $p = p(\sigma)$ is the smallest element not in a critical cycle, $p(\sigma'')$ is either equal to p (if its cycle in σ'' is not critical) or larger than p (if its cycle in σ'' is critical). To prove that we have $p = \tilde{p}$, we need to prove that the latter case is not possible.

Note that the cycle of p in σ'' cannot be of length k, and can be of length $2k$ only when rule 1b is applied: rules 3(a)iii and 3d place p in a $(2k-1)$-cycle (and we have assumed $k \geqslant 2$); rule 1a places it in a $(k+1)$-cycle, and all other rules except 1b place it in a $(k-1)$-cycle; and, since rule 1b reduces by 1 the length of the cycle containing p and only applies if this length is not $k+1$, the length in σ'' cannot be k. Overall, only rule 1b could result in p being in a critical cycle (of length $2k$) of σ''.

Now, in rule 1b, the cycle containing p in σ' (hence in σ'' as well) is obtained by just removing $\sigma^{-1}(p)$. For this cycle to become critical in σ'', it would have to be of length $2k+1$ in σ, with the extra condition that $\sigma^k(p)$ be smaller than all elements in non-critical cycles of σ except $\sigma^{[0,k-1]}(p)$; this condition reduces exactly to $y = p'$ or $x = p'$, $y = p''$, which are excluded from rule 1b. Thus, even when rule 1b is applied, the cycle containing p is not critical, and we have $p = \tilde{p}$. □

Lemma 16. pop *is injective.*

Proof. Let (σ, i) be a pair on which pop is defined, and $\sigma' = \text{pop}(\sigma, i)$. Set $p = p(\sigma)$, $p' = p'(\sigma)$, $p'' = p''(\sigma)$ and $\gamma = \gamma(\sigma')$. It is clear that $\gamma = \sigma(i)$ for $k \geqslant 2$, and $\gamma = i$ for $k = 1$.

The first part of our proof shows that no permutation σ' can be obtained by applying two different rules, *i.e.*, the images of the ten rules are pairwise disjoint. For each rule, we give a condition that can only be satisfied when applying this one rule, and leave it to the reader to check that it is indeed the case. Rule 1b is the most complex, since we have to include the condition that rule 2 does not apply.

1. (a) $p = \sigma'(\gamma)$;

   (b) $p \neq \sigma'(\gamma)$ and $\mathcal{L}_{\sigma'}(p) \neq k-1$, and we are not in the situation (only posible if $k \geqslant 2$) where $\mathcal{L}_{\sigma'}(p) = 2k-1$ and $(\sigma'^k(p) = p'$ or $(\sigma'^{k-1}(p), \sigma'^k(p)) \neq (p', p''))$; these exceptions are not produced by this rule as the cycle from which rule 1b removes an element cannot be a critical one;

2. (a) $\mathcal{L}_{\sigma'}(p) = k-1$, $\sigma'(\gamma) = p'$ and $\mathcal{L}_{\sigma'}(p'') = k+1$;

   (b) $\mathcal{L}_{\sigma'}(p) = k-1$, $\mathcal{L}_{\sigma'}(p') = k+1$ and $\sigma'(\gamma) \neq p'$, and for $k=1$, this condition should instead be: $p = \sigma'(\gamma)$ and $\mathcal{L}_{\sigma'}(p') = 2$;

3. (a) i. $\mathcal{L}_{\sigma'}(p) = k-1$, $\sigma'(\gamma) = p'$ and $\mathcal{L}_{\sigma'}(p'') = 1$;

   ii. $\mathcal{L}_{\sigma'}(p) = k-1$, $\sigma'(\gamma) = p'$, and $\mathcal{L}_{\sigma'}(p'') \notin \{1, k+1\}$;

   iii. $\mathcal{L}_{\sigma'}(p) = 2k-1$, $\sigma'^{k-1}(p) = p'$ and $\sigma'^k(p) = p''$;

   (b) $\mathcal{L}_{\sigma'}(p) = k-1$ and $\mathcal{L}_{\sigma'}(p') = 1$;

   (c) $\mathcal{L}_{\sigma'}(p) = k-1$ and $\mathcal{L}_{\sigma'}(p') \notin \{1, k+1\}$, and for $k=1$, this condition should instead be: $p = \sigma'(\gamma)$ and $\mathcal{L}_{\sigma'}(p') \geqslant 3$;

   (d) $\mathcal{L}_{\sigma'}(p) = 2k-1$ and $\sigma'^k(p) = p'$.

All that remains to do to prove the lemma is to prove that each rule of pop is one-to-one. Suppose now, two distinct pairs (σ_1, i_1) and (σ_2, i_2) have the same image σ' by the same rule of pop. Necessarily, we must have $i_1 = i_2 = \sigma'^{-1}(\gamma)$, and by Lemma 15, $p(\sigma_1) = p(\sigma_2)$.

Analysing each rule, one may note that all cycles of length $2k$ in σ_1 and σ_2 with a minimum less than $p(\sigma_1)$ are critical and are present in σ'; these cycles must thus be identical in σ_1 and in σ_2. Similarly, the k-cycles of σ (other than the one containing i) are also present in $\mathrm{pop}(\sigma, i)$, so that σ_1 and σ_2 have the same k-cycles.

Finally, if the permutations σ_1 and σ_2 differ on the cycle of p (or p', or p'', depending on the rule), applying the same rule to both σ_1 and σ_2 results in different permutations; whereas if σ_1 and σ_2 differ in another place, the permutations obtained using the operator are again different as other cycles are not modified.

Hence $\sigma_1 = \sigma_2$ and pop is indeed an injective function. □

Lemma 13, 14, and 16 together prove Proposition 10.

3.5.2 A bijection for permutations with no k-cycles

Let $\mathcal{D}_n^k$ be the set of permutations of $[n]$ with no k-cycles, and $\mathcal{Q}_n^k$ the set of quasi-special permutations of $[n]$ with no k-cycles. Recall that we denote $f_k(n, 0) = |\mathcal{D}_n^k|$.

Using only the operator just defined, we can easily describe a bijection χ_n from $\mathcal{D}_{n-1}^k \times [n] \setminus \mathcal{SP}_{n-1}^k$ to $\mathcal{D}_n^k \setminus \mathcal{Q}_n^k$, where the excluded set $\mathcal{SP}_{n-1}^k$ is the set of pairs (σ, i) in which σ is special and $\mathcal{L}_\sigma(i) = k-1$. Note that for $\mathcal{SP}_{n-1}^k$ to be nonempty, n must be a multiple of k (the existence of a $(k-1)$-cycle in a special permutation implies that the size is $-1 \bmod k$).

This bijection χ_n gives a combinatorial proof of the recurrence relation for $f_k(n,0)$ given in Proposition 2 (which we already proved analytically, in the proof of Proposition 3, for all numbers of k-cycles).

This bijection can be seen as a generalization of a previous bijection τ_n for derangements (permutations with no fixed points, that is, the case $k=1$) due to Rakotondrajao [13]. The bijection τ_n is defined from $\mathcal{D}^1_{n-1} \times [n] \setminus \{(\Delta^1_{n-1}, n)\}$ to $\mathcal{D}^1_n$ for odd n, and from $\mathcal{D}^1_{n-1} \times [n]$ to $\mathcal{D}^1_n \setminus \Delta_n$ for even n, where Δ_n is the unique critical derangement of size n, for n even: it is the derangement $(12)(34)\dots(n-1\,n)$.

The bijection of [13] gives a nice combinatorial proof of a well-known recurrence over the number of derangements, $d_n = |\mathcal{D}_n|$: $d_n = nd_{n-1} + (-1)^n$. Proposition 2 gives a generalization of this recurrence for any $k \geqslant 1$; note that by taking $k=1$, we recover the original recurrence over derangements $f_1(n,0) = nf_1(n-1,0) + (-1)^n$.

Let σ be a permutation of size $n-1$ with no k-cycles, and $i \in [n]$, such that σ is not special or $\mathcal{L}_\sigma(i) \neq k-1$. We define $\sigma' = \chi_n(\sigma, i)$ as:

- if $\mathcal{L}_\sigma(i) \neq k-1$: $\sigma' = \text{insert}(\sigma, i, n)$;
- if $\mathcal{L}_\sigma(i) = k-1$: $\sigma' = \text{pop}(\text{insert}(\sigma, i, n), i)$.

Theorem 17. *χ_n is a bijection from $\mathcal{D}^k_{n-1} \times [n] \setminus \mathcal{SP}^k_{n-1}$ to $\mathcal{D}^k_n \setminus \mathcal{Q}^k_n$.*

Proof. The first case describes a bijection between the pairs (σ, i) where i is not in a $(k-1)$-cycle and permutations with no k-cycles where n is not in a $(k+1)$-cycle; the second case is a bijection between the remaining pairs and permutations with no k-cycles where n is in a $(k+1)$-cycle. □

The link with the case $\ell = 0$ of Proposition 2 is as follows. When n is not a multiple of k, both $\mathcal{SP}^k_{n-1}$ and $\mathcal{Q}^k_n$ are empty, and χ_n constitutes a bijective proof of the recurrence

$$nf_k(n-1,0) = f_k(n,0).$$

When $n = mk$, one of the two sets is empty, depending on the parity of m:

- For even m, $\mathcal{SP}^k_{n-1} = \emptyset$: special permutations of size $n-1$ with no k-cycles must consist of one $(k-1)$-cycle and a number of $(2k)$-cycles. In this case, χ_n constitutes a bijective proof of the recurrence

  $$nf_k(n-1,0) = f_k(n,0) - Q(n,0).$$

- For odd m, $\mathcal{Q}^k_n = \emptyset$: quasi-special permutations of size n with no k-cycles consist of one $(k-1)$-cycle, one $(k+1)$-cycle and a number of $(2k)$-cycles. In this case, since each special permutation of size $n-1$ has exactly one $(k-1)$-cycle, it appears in exactly $k-1$ pairs in $\mathcal{SP}^k_{n-1}$, and χ_n constitutes a bijective proof of the recurrence

  $$nf_k(n-1,0) - (k-1)S(n-1,0) = f_k(n,0).$$

3.6 Some other operators for (special) permutations

We define here the three remaining operators that our tree construction relies on. These are necessary to preserve the *special* subtree, that is, the property that all special permutations are children of other special permutations in our generation tree. They all use the largest element of a permutation σ which is not in a k-cycle, denoted $\gamma(\sigma)$.

3.6.1 Decreasing the number of k-cycles: the merge operator

In our tree, this merge operator serves to define special children with one fewer k-cycles than their parent. Its direct effect on the cycle type is that two k-cycles are replaced by a $(2k)$-cycle; some care needs to be exercised to ensure that the result is a critical permutation.

Precisely the operator merge is defined over some *acceptable* pairs (σ, i) such that σ is a critical permutation over $\mathcal{S}_{[n]}$, n and i are in two different k-cycles, $\sigma(i)$ is the maximum of its cycle and $\sigma(i) > \gamma(\sigma)$.

This operator splits each $(2k)$-cycle of σ into two k-cycles, one for each critical element of the cycle, then reconstructs σ' by including the cycles containing n and i in the new $(2k)$-cycles set, based on the simple bijection shortly outlined § 3.3.1.

Let us define properly this bijection first. For a permutation τ of size mk constituted only of k-cycles, of which ℓ are marked (with $m - \ell$ even), we call $\mathrm{join}(\tau)$ the critical permutation of size mk obtained as follows:

- we order the unmarked cycles C_1, C_2, $\ldots$, $C_{m-\ell}$ of τ in increasing order of their minimal elements;
- we join these cycles pairwise by connecting C_i to C_{i-1} for even i, *i.e.*,
  - $\tau_0 = \tau$,
  - for $j \in [\frac{m-\ell}{2}]$, $\tau_j = \mathrm{connect}(\tau_{j-1}, \min(C_{2j}), \min(C_{2j-1}))$,
  - $\mathrm{join}(\tau) = \tau_{(m-\ell)/2}$.

Note that this operation defines a bijection from its input set to critical permutations, as we can easily recover the permutation τ formed by m cycles of length k from any critical permutation τ' of size mk by splitting the $(2k)$-cycles of τ'; the marked cycles of τ are exactly the k-cycles of τ'.

For (σ, i) an acceptable pair for merge, we define the permutation $\sigma' = \mathrm{merge}(\sigma, i)$ as follows:

1. split the $(2k)$-cycles of σ into k-cycles, *i.e.*, $\sigma_1 = \mathrm{join}^{-1}(\sigma)$; unmark the cycles of n and i to get σ'_1;
2. merge again the unmarked k-cycles of σ'_1 into $(2k)$-cycles, *i.e.*, $\sigma' = \mathrm{join}(\sigma'_1)$.

Proposition 18. merge *defines a bijection between its acceptable pairs and critical permutations where n is in a $(2k)$-cycle. The image has exactly two fewer k-cycles than the permutation in the pair.*

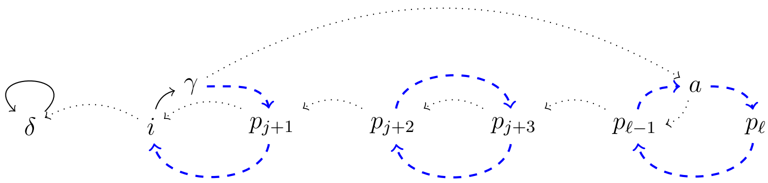


Figure 5: Illustration of the shift operator when i is a critical element: $i = p_j$. Dotted arrows represent new positions of elements after the application of the operator.

Proof. The first step in merge is a bijection between acceptable pairs and permutations with only k-cycles with an even number of unmarked cycles, with n belonging to an unmarked cycle; here, i can be recovered from the marked permuation by first recovering $\sigma(i)$ as the maximum element among the unmarked cycles not containing n.

The second step is reduced to join, and is also bijective from marked permutations with the cycle of n unmarked, and critical permutations with n in a $2k$-cycle. □

3.6.2 Special children: the shift operator

We now define an operator shift that bijectively maps some *valid* non-special permutation τ to some special permutation σ' on the same set, which contains at least one $(2k)$-cycle. The conditions on τ are as follows:

- $\mathcal{L}_\tau(\gamma(\tau)) = 2k + 1$;
- $\sigma = \text{remove}(\tau, \gamma(\tau))$ is a special permutation whose special cycle (if it exists) has length other than $k - 1$;
- $i = \tau^{-1}(\gamma(\tau)) > \delta(\sigma)$, where $\delta(\sigma)$ is the minimum element of the special cycle of σ (0 if $|\sigma| = 0 \bmod k$).

The purpose of the shift operator in the construction of our generation tree is to ensure the heredity condition: that special permutations only appear as children of special permutations. When trying to define the i-th child of a special permutation σ, if i happens to be in a cycle of length k or $2k$ in σ, using cycle insertion would result in a permutation that has a $(k+1)$- or $(2k+1)$-cycle, thus a non special permutation; and in other situations, cycle insertion may result in special permutations being children of non special ones, which we want to avoid completely. The shift operator is used to "switch children" in the tree in order to maintain the hereditary property.

Informally, the way shift^{-1} acts is by "shifting up" the sequence of critical elements of a special permutation (obtaining a non special permutation as the result) in such a way that the operation can be reversed. It is slightly more natural to define the reverse operation (from non special to special permutation) as shift, illustrated by Figure 5.

Let τ be some valid permutation in the sense described above. In the following, let $\gamma = \gamma(\tau)$, $\sigma = \text{remove}(\tau, \gamma)$, $i = \tau^{-1}(\gamma)$, $p_j = p_j(\sigma)$ for $j \in [2m]$ (where m is the number of $(2k)$-cycles in the special permutation σ; *i.e.*, the p_j's are the critical elements of σ), and $\delta = \delta(\sigma)$ if σ has a special cycle, $\delta = i$ otherwise (that is, if $|\sigma|$ is a multiple of k).

The permutation $\sigma' = \text{shift}(\tau, i)$ is defined as follows:

- If i is not a critical element of σ, then $\sigma' = \text{insert}(\tau, \delta, i)$.
- If $i = p_j$ for some $j \in [2m]$, then let ℓ be the smallest integer larger than j, such that $p_\ell > \min(\sigma^{[k-1]}(p_{\ell-1}))$ (if such an ℓ does not exist, we set $\ell = 2m+1$). Let $a = \min(\sigma^{[k-1]}(p_{\ell-1}))$, and let σ' be the permutation obtained after these $\ell - j + 2$ *insertions*:
  1. $\tau_{\ell-j} = \text{insert}(\tau, a, \gamma)$ and $\tau_{\ell-j-1} = \text{insert}(\tau_{\ell-j}, p_{\ell-1}, a)$;
  2. for $h \in \{0, \dots, \ell - j - 2\}$, $\tau_h = \text{insert}(\tau_{h+1}, p_{j+h}, p_{j+h+1})$;
  3. $\sigma' = \text{insert}(\tau_0, \delta, i)$.

In other words, each of the p_h (for $j \leqslant h \leqslant \ell - 2$) in τ is replaced by p_{h+1} in the cycle structure of τ, $p_{\ell-1}$ is replaced by a, a is replaced by γ, and $i = p_j$ is inserted after δ (or as a new fixed point if $\delta = i$, that is, if τ is of size 1 mod k). Thus, most cycle lengths are the same in σ' and τ; the only ones that change are the $(2k+1)$-cycle that becomes a $(2k)$-cycle, and the one cycle of length less than k (the special cycle in σ) that increases its size by one. The conditions on the choice of ℓ and a ensure that, even though the critical elements of σ are replaced by larger values, σ' is indeed a special permutation.

Proposition 19. *The* shift *operator defines a bijection between its valid inputs τ and special permutations σ' on the same set with the same number of k-cycles, where $\gamma(\sigma')$ is in a $(2k)$-cycle.*

Proof. We have already argued that σ' is special. It has the same number of k-cycles as τ; in fact, it has exactly the same k-cycles. In the definition, γ is placed in a $(2k)$-cycle; and, since the elements in k-cycles are the same in τ and in σ', we have $\gamma(\sigma') = \gamma$. Thus, shift(τ) is of the claimed type. Also, note that the validity conditions on τ (in particular, $i > \delta(\tau)$) ensures that $\delta(\sigma') = \delta$.

To prove that shift is indeed bijective, we prove that for any permutation σ' of the claimed type, there is only one possible preimage τ, and leave it to the reader to check that applying shift to this τ does yield σ'.

Starting from σ', we already know that γ can be identified as $\gamma(\sigma')$; i can also be identified as $\sigma'(\delta(\sigma'))$. The shifted critical elements can then be identified (in σ') as the critical elements larger than i, up to and including the critical element p_j such that $\gamma \in \sigma'^{[k-1]}(p_j)$ (possibly none, if this p_j is smaller than i). Then, τ can be recovered from σ' by replacing each of these critical elements by the next (the smallest one is replaced by i, and the last replaces γ), then inserting γ after i. □

3.6.3 Generating quasi-special permutations: the cut operator

Our last operator $\sigma' = \mathrm{cut}(\tau)$ defines a bijection between critical permutations τ with *at least* one $(2k)$-cycle, and quasi-special permutations σ' on the same set. Its direct effect on the cycle type is that a $(2k)$-cycle is replaced by a $(k-1)$-cycle and a $(k+1)$-cycle.

Recall the conditions on a quasi-special permutation σ': it has one $(k+1)$-cycle, one $(k-1)$-cycle, and any number of k-cycles and $(2k)$-cycles; when ignoring the two $(k \pm 1)$-cycles, it is special; the minimum element p in the $(k-1)$-cycle is larger than all "critical" elements; the maximum element γ in the $(k+1)$-cycle is larger than all elements not in k-cycles; and finally, its image $p' = \sigma'(\gamma)$ is larger than p.

Now let us start from a critical permutation τ with at least one $(2k)$-cycle.

Let us define how the permutation σ' is obtained through this operator. Let $m \geqslant 1$ be the number of $(2k)$-cycles in τ, $\gamma = \gamma(\tau)$ and let p be the largest critical element of τ such that $\gamma \notin \tau^{[k-1]}(p)$, *i.e.*,

$$p = \begin{cases} p_{2m-1}(\tau) & \text{if } \gamma \in \tau^{[k-1]}(p_{2m}(\tau)), \\ p_{2m}(\tau) & \text{if } \gamma \notin \tau^{[k-1]}(p_{2m}(\tau)). \end{cases}$$

The permutation σ' is obtained through the following process:

1. τ_1, a permutation of k-cycles only with some cycles marked, is obtained by splitting all $(2k)$-cycles of τ at their critical elements, that is, $\tau_1 = \mathrm{join}^{-1}(\tau)$;

2. mark two more cycles in τ_1 to get τ_2: those containing γ and p (these two cycles are distinct from the definition of p);

3. let $\tau_3 = \mathrm{join}(\tau_2)$ and $p' = \tau(p) = \tau_3(p)$, and set $\sigma' = \mathrm{insert}(\tau_3, \gamma, p')$.

Proposition 20. cut *defines a bijection between critical permutations with at least one* $(2k)$*-cycle, and quasi-special permutations on the same set. Furthermore, the numbers of* k*-cycles of* τ *and* $\mathrm{cut}(\tau)$ *are the same.*

Proof. We start by checking that σ' defined above is indeed a quasi-special permutation. Its cycle structure is appropriate: τ_3 has only k- and $(2k)$-cycles, and the last insertion operation removes an element from one of the k-cycles to insert it into another one. The fact that, when these two $(k \pm 1)$-cycles are ignored, the rest is a critical permutation is guaranteed by the join operation. The minimum element p in the $(k-1)$-cycle is larger than the largest critical element of τ_3. Obviously, the elements not in k-cycles are the same in τ and σ', so that $\gamma(\sigma')$ is indeed γ, and is in the $(k+1)$-cycle of σ' (and σ' and τ have the same number of k-cycles). Finally, p is the minimum of its cycle, so that $p' > p$ is also guaranteed. These are exactly the conditions of a quasi-special permutation.

We now check that any quasi-special permutation σ' can have only one preimage τ, and describe the inverse map; again, we leave it to the reader to check that this is indeed a preimage.

Starting from some arbitrary quasi-special permutation σ', we already know that $\gamma(\tau)$ must be $\gamma(\sigma')$, and can also recover p as the minimum of the $(k-1)$-cycle, and p' as

$\sigma'(\gamma)$. Thus, we can also recover τ_3 by inserting p' after γ ($\tau_3 = \text{insert}(\sigma', \gamma, p')$), then $\tau_2 = \text{join}^{-1}(\tau_3)$, then unmark the cycles containing γ and p to recover τ_1. Finally, τ is recovered as $\text{join}(\tau_1)$. □

4 Proof of the main theorem

In this section, we prove the main result of the paper, namely, that our definition, given in page 6, does define a uniform generation tree with the desired properties. We first prove the general case $k \geqslant 2$; the case $k = 1$, together with the somewhat simplified description of the tree, will be treated later.

4.1 Proof for $k \geqslant 2$

Before going to the proof of the theorem, we make some comments on the definition, and derive some easy properties.

First note that the conditions on the use of rules II to VI are mutually exclusive, so that, by the fact that rule I covers exactly the cases not covered by the others, our description uses exactly one rule for each possible pair (σ, i) (we will check later that the rules are well defined, *i.e.*, that the operations are always used with proper arguments).

The condition of rule I was not made explicit in the description, but it can be expressed easily. The rule is applied when one of the following statements holds:

- $\ell = 2k - 1$ and ζ is not special,
- $\ell = 2k$, and (σ is not special or $i \leqslant \delta(\sigma)$ or $k \nmid n$),
- $\ell \notin \{k - 1, 2k - 1, 2k\}$.

Now note that in the definition, ℓ is always either 0 or the length of a cycle in σ. This makes it clear that rule I is the only one which is applied both to special and non-special permutations: rules II, III, and IV are used only for non-special permutations (the latter two imply that σ contains a $2k - 1$ cycle), and rules V and VI for special permutations. Another useful property that always holds is $\ell + 1 = \mathcal{L}_\zeta(i)$, and $\zeta(i) \geqslant \gamma(\zeta)$ (just check the three cases of the definition of (ℓ, ζ)).

For the remainder of the proof, we note $\gamma = \gamma(\sigma)$ and $\mathcal{L}(j)$ for $\mathcal{L}_\sigma(j)$.

We now examine each individual rule and, for each one, we do three things. We check that the permutation σ' is well defined (that is, the corresponding operator can be applied); we determine whether σ' is special; and we determine the number j' of k-cycles of σ', assuming its parent permutation σ has j cycles of length k.

I. In this rule, either a fixed point is added, or the length of exactly one cycle containing i or γ increases by one:
   - if $\mathcal{L}(i) \neq k$, ζ corresponds to cycle insertion after i; since the cycle of i cannot be of length $k - 1$ (captured by rules II and VI), σ' has the same number of k-cycles as σ.

- if $\mathcal{L}(i) = k$ and $i < \gamma$, ζ corresponds to the permutation where i is replaced by n, then i is inserted just before γ, thus increasing the length of γ's cycle by one in σ'. Since $\mathcal{L}(\gamma) \neq k-1$ (captured again by rules II and VI), we get $j' = j$ again.
- if $\mathcal{L}(i) = k$ and $i > \gamma$, ζ is the permutation obtained by replacing i with n, then inserting i as a fixed point, hence $j' = j$ in this case as well.

Moreover if σ is non special, σ' is non special as well, because there are only three ways for this cycle insertion to create a special permutation, and each one is prevented in this situation:

- inserting into a cycle of length strictly less than $k-1$, but in this case σ would have been special;
- inserting into a cycle of length $k-1$, but rule I does not apply in this case;
- inserting into a cycle of length $2k-1$, but rules III and IV exactly capture this case when ζ is special.

If σ is special, there is only one possibility for σ' to be special as well, that is, if we insert into the special cycle, and it happens to be of length strictly less than $k-1$. This corresponds to the case $\ell < k-1$ (note the case $\ell = k-1$ is captured by rule VI). In the only other possible situation where rule I applies to a special permutation (that is, $\ell = 2k$ but the condition of rule V fails), the permutation σ' obtained after rule I is not special since it contains a cycle of length $2k+1$.

II. Since σ is not special in this rule, ζ, which is obtained from σ by inserting some value into a $(k-1)$-cycle, is not critical. Second, the newly created k-cycle in ζ contains i (just check the two first cases of the definition of ζ). Third, $\zeta(i)$ is either n or $\gamma(\sigma)$; in both cases, $\zeta(i)$ is maximal in its cycle and larger than the maximum element of ζ not in a k-cycle. These three conditions show that $\mathrm{pop}(\zeta, i)$ is well defined. By the properties of pop, σ' is not special and $j' = j$.

III. Here ζ is special, $\gamma(\zeta)$ is either n or γ, and $\gamma(\zeta)$ is in a $(2k)$-cycle of ζ. Hence ζ lies in the image of shift, so that σ' is well defined and non special, and $j' = j$.

IV. In this rule, ζ is critical and has a $(2k)$-cycle, hence the operator cut is defined on ζ; σ' is quasi-special (hence not special), and its number of k-cycles is the same as that of ζ (and of σ).

V. (a) Since σ is special, $i > \delta(\sigma)$ and n is not a multiple of k, so that the special cycle in σ, whose length is $n-1 \bmod k$, has length strictly less than $k-1$. In this case, $\zeta(i) = \gamma(\zeta)$ is in a $(2k+1)$-cycle of ζ, and removing this value from ζ yields a special permutation (which may be σ, or another special permutation). Thus ζ matches the criteria to apply the shift operator, and we get a special σ' with $j' = j$.

(b) In this situation, σ' is obtained by replacing i by n (keeping the same number of k-cycles) and placing it after $\delta(\sigma)$ thus making σ' special as well by only increasing the length of the special cycle; again, we have $j' = j$.

VI. (a) This rule (identical to rule I but kept separate because it plays a specific role in our proof) inserts n after i, thus creating a new k-cycle in σ'. Since other elements are left undisturbed, σ' is a special permutation with no special cycle (n is a multiple of k), and $j' = j + 1$.

(b) In this case, n and i are in different k-cycles in ζ, ζ is critical (the $(k-1)$-cycle of γ has been replaced by a k-cycle), and $\zeta(i)$ is larger than any element of ζ outside k-cycles; hence merge can be applied to ζ. The operator produces a critical permutation σ' with two k-cycles fewer than ζ, thus one fewer than σ ($j' = j - 1$).

This case by case analysis shows that, provided our construction does define a generation tree (which we shall prove next), the tree does have all the claimed properties. We summarize them in the following lemma:

Lemma 21. *The parent of each special permutation is special. The number of k-cycles of a permutation is different from that of its parent if and only if both are special, and the former is critical; the difference in the number of k-cycles is ± 1, depending on whether rule VIa or VIb is used to obtain the child from the parent.*

We now turn to proving that our construction is indeed a generation tree, that is, we have defined, for each n, a bijection between $\mathcal{S}_{[n-1]} \times [n]$ and $\mathcal{S}_{[n]}$. Since both sets obviously have the same cardinality, all we need to do is prove that the transformation is one-to-one by exhibiting necessary conditions on a preimage for each permutation.

Let σ' be any permutation of size n, $\gamma = \gamma(\sigma')$ and $\tau = \text{remove}(\sigma', \gamma)$ the permutation of size $n - 1$ where γ has been removed from its cycle.

If σ' is special but not critical with $\mathcal{L}_{\sigma'}(\gamma) = 2k$ and $\mathcal{L}_{\sigma'}(n) = k$: σ' is the i-th child by rule Vb of $\sigma = \text{remove}(\text{insert}(\sigma', \sigma'^{-1}(n), i), n)$, with $i = \sigma'(\delta(\sigma'))$. In σ the cycle of i is of length k (n has been removed, i added); σ is special with $|\text{sc}(\sigma)| < k - 1$ and $i > \delta(\sigma)$, thus rule Vb produces σ'.

We can now note that all other rules describe σ' as the image of ζ or (ζ, i) by a bijective transform (shift, shift^{-1}, pop, cut, merge, and the identity restricted to the application cases of rules I and VIa). Thus, if we check that the images of these transforms are pairwise disjoint, then we know that some ζ or pair (ζ, i) can be recovered from σ'. In those cases where only ζ is recovered this way, we also show how i can be independently recovered. Then, the proof is completed by checking that knowing ζ and i is enough to recover σ (that is, the mapping $(\sigma, i) \mapsto (\zeta, i)$ is injective).

The various images correspond to the following (pairwise incompatible) descriptions:

I.
- $\gamma \neq 0$ and $\mathcal{L}_{\sigma'}(\gamma) \notin \{k+1, 2k, 2k+1\}$, or
- σ' is non-special, $\mathcal{L}_{\sigma'}(\gamma) = 2k$, or

- $\mathcal{L}_{\sigma'}(\gamma) = 2k+1$ with either $k \mid n$, τ is non-special, or τ is special and $\sigma'^{-1}(\gamma) \leqslant \delta(\tau)$.

II. σ' is non-special with $\mathcal{L}_{\sigma'}(\gamma) = k+1$, but σ' is not quasi-special.

III. σ' is a valid input of shift operator, *i.e.*, $\mathcal{L}_{\sigma'}(\gamma) = 2k+1$, $k \nmid n$, τ is special and $\sigma'^{-1}(\gamma) > \delta(\tau)$.

IV. σ' is quasi-special.

V. (a) σ' is special, $\mathcal{L}_{\sigma'}(n) = 2k$, $|\mathrm{sc}(\sigma')| \geqslant 1$.

(b) σ' is special, $\mathcal{L}_{\sigma'}(n) = k$, $|\mathrm{sc}(\sigma')| \geqslant 1$, and $\gamma \notin \mathrm{sc}(\sigma')$.

VI. (a) σ' is special, $\mathcal{L}_{\sigma'}(n) = k$, $|\mathrm{sc}(\sigma')| = 0$.

(b) σ' is special, $\mathcal{L}_{\sigma'}(n) = 2k$, $|\mathrm{sc}(\sigma')| = 0$.

Let us give a few words of explanation as to why these rules cover all possible cases. The case $\gamma = 0$ is included in condition Vb (it corresponds to σ' having only k-cycles). When $\gamma \neq 0$, we have a disjunction based on the value of $\mathcal{L}_{\sigma'}(\gamma)$: value $k+1$ is covered exactly by rules II and IV; value $2k+1$, by rule III and the last case of rule I; value $2k$ is covered by the second case of rule I and rules V and VI; and all other values by the first case of rule I.

Since these conditions cover all possibilities, each permutation σ' must fall in one condition, and we can deduce then a rule, the permutation ζ and the value of $i = \sigma'^{-1}(\gamma)$ for all cases (except the case of rule Vb which was treated above).

Once i and ζ are known, it is straightforward to reconstruct the permutation σ as $\sigma = \mathrm{remove}(\mathrm{insert}(\zeta, n, i), n)$. □

4.2 Generation tree for $k = 1$

When k is 1 (that is, we are interested in the number of *fixed points* in permutations), the values $2k$ and $k+1$ coincide, and though the general description of the generation tree is still valid, some of our arguments for the previous proof are not; and at the same time, some of the rules in the general description never apply. We now provide a more compact description of the same generation tree in this case; the reader should check that what follows is indeed equivalent, when $k = 1$, to what we gave earlier.

The authors gave in [6] a description of a generation tree with similar properties to the one described here for $k = 1$, but it should be stressed that the tree is not the same. The tree described here is almost identical to the one appearing in the second author's Ph.D. thesis [8], with slight changes introduced by taking images in place of preimages in rules (3.a) and (4.c), and reversing the 3-cycle of p in rule (4.b).

For the rest of this section, we assume $k = 1$. Recall that in this case, "special", "quasi-special" and "critical" permutations are the same. Special permutations of size n are made of an arbitrary set of ℓ fixed points such that $n - \ell$ is even, and the rest of the permutation forms the unique critical permutation over the remaining elements, that is,

Fixed-Point-Generation-Tree

The root of the tree is (1). Let $n \geqslant 2$, $\sigma \in \mathcal{S}_{[n-1]}$, $i \in [n]$, $\gamma = \gamma(\sigma)$, $p = p(\sigma)$, $p' = p'(\sigma)$ and $\tau = \mathrm{insert}(\sigma, n, n)$. The i-th child σ' of σ is defined as follows:

1. If σ is special and $i = n$: $\sigma' = \tau$.
2. If σ is special and $\gamma < i < n$: $\sigma' = \mathrm{insert}(\sigma, i, n)$.
3. If $i \leqslant \gamma$:
   (a) If $i = \sigma(i)$: $\sigma' = \mathrm{insert}(\tau, \sigma^{-1}(\gamma), i)$.
   (b) If $i \neq \sigma(i)$: $\sigma' = \mathrm{insert}(\sigma, i, n)$.
4. If σ is non-special and $i > \gamma$:
   (a) If $\mathcal{L}_\sigma(p) = 2$: $\tau' = \mathrm{insert}(\tau, p', \sigma(p))$, and $\sigma' = \mathrm{insert}(\tau', i, p)$.
   (b) If $\mathcal{L}_\sigma(p) = 3$ and $\sigma(p) = p'$: $\sigma' = \mathrm{insert}(\tau', i, p)$.
   (c) Otherwise: $\sigma' = \mathrm{insert}(\tau', i, \sigma^{-1}(p))$.

the non-fixed points are paired into 2-cycles in increasing order. This makes it easier to see that there are 2^{n-1} special permutations of size n; they are in bijection with subsets of $[n]$ of even size.

Let us reconsider the pair (ℓ, ζ) constructed as the beginning of the general description. We get the following pair by setting $k = 1$:

$$(\ell, \zeta) = \begin{cases} (\mathcal{L}_\sigma(i), \tau) & \text{if } i = n \text{ or } \sigma(i) \neq i, \\ (\mathcal{L}_\sigma(\gamma(\sigma)), \mathrm{insert}(\tau, \sigma^{-1}(\gamma(\sigma)), i)) & \text{if } \sigma(i) = i \text{ and } i < \gamma(\sigma), \\ (0, \mathrm{insert}(\tau, i, i)) & \text{if } \gamma(\sigma) < i < n, \end{cases}$$

since $\mathcal{L}_\sigma(i) = 1$ corresponds to i being a fixed point; note here $\ell = 0$ happens when $i > \gamma(\sigma)$ (including the case $i = n$), otherwise we have $\ell \geqslant 2$.

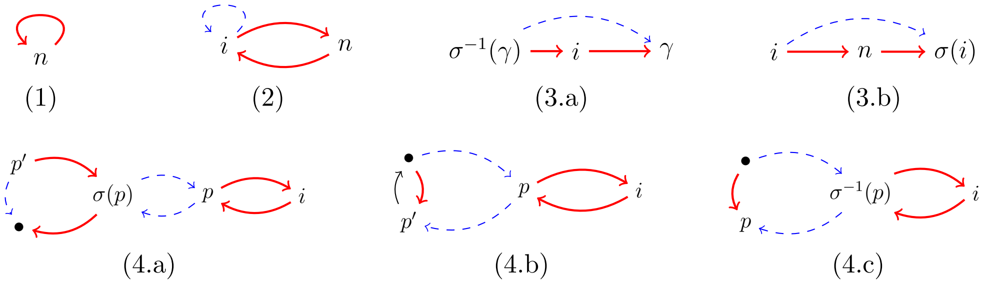
Now reconsidering the rules of the general description: rules III, IV and V never apply since $1 \mid n$ and $\ell \neq 1$.

Rule II applies when σ is non-special and $\ell = 0$, *i.e.*, when $i > \gamma(\sigma)$. In the new description, this is covered by rule 4 which is just the description of pop for $k = 1$.

Rule VIa applies when σ is special and $\mathcal{L}_\sigma(i) = 0$, *i.e.*, $i = n$. In this case, it corresponds to the insertion of n as a fixed point in σ. This is covered by rule 1 in the new description.

Rule VIb applies when σ is special, $\gamma(\sigma) < i < n$ (hence i is a fixed point). In this case, merge adds a new 2-cycle containing i and n, as described in the new rule 2.

The remaining original rule, rule number I, corresponds to the remaining cases, that is, $i \leqslant \gamma(\sigma)$. In this case, there are two possibilities, either i is not a fixed point in σ (and n is inserted after i), or i is a fixed point (and it is inserted just before $\gamma(\sigma)$). This is covered by the new rule 3.



(1) (2) (3.a) (3.b)

(4.a) (4.b) (4.c)

Figure 6: Illustration of our generation tree for $k = 1$ and $\sigma \in \mathcal{S}_{[n-1]}$; dashed edges represent relations in σ, whereas thick edges represent relations in σ', the i-th child of σ in the tree. Only differences between σ and σ' are shown.

The remaining constructions are illustrated in Figure 6. It should be clear that this construction defines a uniform generation tree, since from any permutation σ' we can identify which rule has been applied and easily reverse the operation:

- rules 1 and 2 build special permutations with n either as a fixed point or in a 2-cycle, and $i = \sigma'^{-1}(n)$.
- rule 3 builds only permutations where $\gamma(\sigma')$ is in a 3-cycle, and we have $i = \sigma'^{-1}(\gamma(\sigma'))$.
- rule 4 builds non-special permutations where $\gamma(\sigma')$ is in a 2-cycle, and we have $i = \gamma(\sigma')$.

5 Probabilistic consequences

In this section, we describe a few applications of our tree construction in a probabilistic setting.

5.1 Stabilization time for the number of k-cycles

Special permutations make up a vanishing proportion of all permutations of a given size as size goes to infinity; thus, the probability that a random descent reaches a changing permutation at a given level goes to zero. Thanks to the heredity property of our generation tree, *almost surely, the number of k-cycles in the permutations of a random descent is ultimately constant.* More precisely, almost all descents will reach a non-special permutation, and all permutations reached from then on will have the same number of k-cycles.

Let T' denote the size of the first non-special permutation in a random descent; T' is obviously a stopping time[2]. If we define time T as the stabilization time for the number

[2] with respect to the natural filtration defined by the sequence of permutations.

of k-cycles, that is, the first index n such that for all $m \geqslant n$ we have $c_k(\sigma_m) = c_k(\sigma_n)$, T is not a stopping time; for instance, in the generation tree for $k = 2$ as shown in Figure 1, all sequences containing the permutation (1)(2) have $T = 1$ (and $T' = 2$), but just looking at the first permutation (1) (which is a special permutation) does not let one know the value of T. But, because each permutation of size a multiple of k either is non-special or has a different number of k-cycles from its parent, T' cannot be larger than $T + k$.

For the applications we wish to describe in the rest of this section, it is useful to compute the expected value of T'.

Proposition 22. *Let T' denote the size of the first non-special permutation in a random descent into our tree. Then T' has finite expectation*

$$\mathbb{E}(T') = (1 + H_{k-1})\frac{e^{2/k} + 1}{2},$$

where $H_n = \sum_{i=1}^{n} 1/i$ denotes the n-th harmonic number.

Proof. The expected time can be expressed as $\mathbb{E}(T') = \sum_n \mathbb{P}(T' \geqslant n)$, and the probability that T' is at least n is exactly the probability that the permutation reached at level $n-1$ of the tree is special, *i.e.*, $S(n-1)/(n-1)!$.

Thus, we have

$$\begin{aligned}\mathbb{E}(T') &= \sum_{n \geqslant 0} \frac{S(n)}{n!} \\ &= 1 + \sum_{h=1}^{k-1} \frac{S(h)}{h!} + \sum_{m=1}^{\infty} \left(\frac{S(mk)}{(mk)!} + \sum_{h=1}^{k-1} \frac{S(mk+h)}{(mk+h)!} \right).\end{aligned}$$

Using the counting formulae of Proposition 7, we get

$$\begin{aligned}\mathbb{E}(T') &= 1 + \sum_{h=1}^{k-1} \frac{(h-1)!}{h!} + \sum_{m=1}^{\infty} \left(\frac{1}{2} \frac{(2/k)^m}{m!} + \sum_{h=1}^{k-1} \frac{1}{2h} \frac{(2/k)^m}{m!} \right) \\ &= 1 + H_{k-1} + \frac{1 + H_{k-1}}{2} \sum_{m=1}^{\infty} \frac{(2/k)^m}{m!} \\ &= (1 + H_{k-1})\, \frac{e^{2/k} + 1}{2}.\end{aligned}$$

□

5.2 Uniform random generation of permutations with a fixed number of k-cycles

In this section, we show how our generation tree can be used for random generation.

The fact that non-special permutations have the same number of k-cycles as each of their children, and by induction as any of their descendants, allows us to design algorithms that make use of this property for sampling a uniform permutation conditioned on having a prescribed number of k-cycles. The algorithm we get is efficient both in time/space complexity and in *randomness* complexity. The randomness cost can be estimated by two different parameters, and we analyze our sampling algorithm for both: the number of random bits used by the algorithm (which we call the *random bit* complexity), and the number of calls to a unit-cost $\texttt{Random}(m)$ primitive that returns a uniform number in $[m]$ (which we call the *random integer* complexity; all calls to such a Random primitive would be with parameters bounded by n).

In the discussion that follows, we consider both k and ℓ to be constants (the asymptotics is only for large n), though they may be large constants.

A very basic rejection algorithm to sample uniformly a (n, ℓ)-permutation, that is a permutation of size n with exactely ℓ k-cycles, is to generate uniform permutations of size n, until one is obtained that has exactly ℓ cycles of length k. Since the probability of success $f_k(n, \ell)/n!$ converges (quickly, see [2]) to $e^{-1/k}/(k^\ell \ell!)$ in the large n limit, this simple *rejection method* needs to generate, in expectation, $e^{1/k}k^\ell \ell! + O(1/n!)$ permutations of size n. Thus its expected random integer complexity is $e^{1/k}k^\ell \ell! n + O(1/(n-1)!)$ when using standard methods for generating permutations, such as the Fisher-Yates shuffle, also known as Knuth shuffle (see [9, 7, 10]).

If ℓ is large, this cost is impractical. There is an easy trick to avoid this large multiplicative constant in front of n: we can generate separately the k-cycles and the rest of the permutation. To do so, we can sample ℓk elements from $[n]$ by ℓk successive calls to a uniform sampler, the i-th call using $\texttt{Random}(n-i+1)$ in order to get one of the remaining elements – this can be done easily in $\mathcal{O}(1)$ per generation by just swapping some elements in a table, in a Fisher-Yates fashion. At the same time as selecting these elements, we can build k-cycles in the same order as their selection: each $(\ell k, \ell)$-permutation of the selected elements is equally likely this way. Then the second part asks for a uniform $(n - \ell k, 0)$-permutation of the remaining elements. In order to generate such a permutation of size $n - \ell k$, a naive rejection algorithm in the same spirit as in the previous paragraph needs in average $e^{1/k}(n-\ell k) + O(1/(n-\ell k-1)!)$ calls to a random sampler. Hence this method yields an algorithm with random integer complexity $e^{1/k}n + \ell k(1 - e^{1/k}) + o(1)$.

For any value of n, ℓ and k the above method is satisfactory in a practical sense. However, we can improve the multiplicative constant and obtain, thanks to our generation tree, a new random generation algorithm with random integer cost $n + \mathcal{O}(1)$ calls to $\texttt{Random}()$. This algorithm is a rejection algorithm heavily based on our generation tree.

Since $e^{1/k}$ is rather close to 1 for values of k larger than a few units, the most interesting case in such a generation process remains the uniform generation of derangements ($k = 1$, $\ell = 0$).

In the literature, [12] proposes an equivalent of the Fisher-Yates shuffle for derangements, resulting in a bounded time algorithm with an expected random integer complexity of $2n + o(n)$. In [1], experimental studies are performed comparing this method and the anticipated rejection method, resulting in comparable results; an extension to permutations having only cycles of length at least $m \geqslant 3$ is also given. In [6, 8], a generator using $n + \mathcal{O}(1)$ calls is given, matching the cost of an algorithm sketched at the end of [3], which is based on Lehmer encoding and a specific bijection of permutations, and not directly on derangements.

We propose the following natural algorithm UNIFORMPERM0(n,k) in order to sample uniformly a permutation of size n with *no* k-cycles:

1. Perform a random descent in our generation tree until reaching a non-special permutation σ or level n, whichever comes first.

2. If the reached permutation has no k-cycles, continue the descent until reaching level n and return the permutation obtained at the end; otherwise, repeat Step 1.

Proposition 23. UNIFORMPERM0*(n,k) returns a uniform permutation of size n that does not have any k-cycles; it has expected random integer complexity $n + \mathcal{O}(1)$.*

Proof. The algorithm works as an anticipated rejection method: it is equivalent to performing a uniform sampling of a permutation of size n, and as soon as the generated permutation is known to have cycles of length k, the algorithm aborts and retries; otherwise it returns the sampled permutation. Thus the returned permutations are uniform $(n, 0)$-permutations. The properties of the generation tree are used to make anticipated rejection efficient.

Let $A = A_n$ be the random variable describing the number of calls to `Random()` used by the algorithm on input n; recall that k is assumed to be a constant. We separate the cost into two contributions corresponding to the two phases of the algorithm, such that $A = C_1 + C_2$, where C_1 counts the calls to `Random()` made (including rejections) until a $(n, 0)$-permutation is obtained that either is non-special, or has size n; and C_2 counts only calls made by step 2 of the algorithm.

We have $\mathbb{E}(C_1) = \mathbb{E}(C)/\alpha$, where C is the number of calls during one trial of the first phase, and $\alpha = q_k(n, 0)$ the probability of success for each trial.

From the exponential generating function $G_k(x, 0) = \exp(-x^k/k)/(1 - x)$ counting the number of $(n, 0)$-permutations (see proof of Proposition 2), we get $q_k(n, 0)$ as the coefficient of x^n in $G_k(x, 0)$:

$$q_k(n,0) = [x^n]\left(\sum_{j\geqslant 0} x^j\right)\left(\sum_{j\geqslant 0}\frac{(-1)^j x^{jk}}{k^j j!}\right) = \sum_{j=0}^{\lfloor n/k\rfloor}\frac{(-1)^j}{k^j j!},$$

thus

$$\alpha = \sum_{j=0}^{\lfloor n/k\rfloor}\frac{(-1)^j}{k^j j!} = e^{-1/k} - \sum_{j=\lfloor n/k\rfloor+1}^{\infty}\frac{(-1)^j}{k^j j!}.$$

The last term is an alternating series, so that the limit differs from $e^{-1/k}$ by at most the first term; independently of the parity of $\lfloor n/k\rfloor$, we get the valid bounds

$$e^{-1/k} - \frac{1}{k^{1+\lfloor n/k\rfloor}(\lfloor n/k\rfloor+1)!} \leqslant \alpha \leqslant e^{-1/k} + \frac{1}{k^{1+\lfloor n/k\rfloor}(\lfloor n/k\rfloor+1)!},$$

so we have $1/\alpha = e^{1/k} + \mathcal{O}(1/(\lfloor n/k\rfloor + 1)!)$.

We now compute an upper bound on $\mathbb{E}(C)$. The number of calls to our random integer generator is exactly one less than the level at which the descent ends; that is, $\mathbb{E}(C) = \mathbb{E}(\min(n, T')) - 1 \leqslant \mathbb{E}(T') - 1$.

Finally, C_2 is at most $n-2$, since the smallest non-special permutations are of size 2. This carries over to $\mathbb{E}(C_2) \leqslant n-2$. By linearity of expectation, we obtain a bound on the expected total number $\mathbb{E}(A) = \mathbb{E}(C_1) + \mathbb{E}(C_2)$ of calls,

$$\mathbb{E}(A) \leqslant n - 2 + e^{1/k}\left(H_{k-1} + \frac{1 + H_{k-1}}{2}\left(e^{2/k} - 1\right)\right) + \mathcal{O}((n/k)!^{-1}). \qquad \square$$

To estimate the random bit complexity, we assume that the primitive `Random(j)` is optimal in the sense described in [11]. This implies that, for any $j > 2$, the expected random bit cost of the call `Random(j)` is at most $2 + \log_2(j)$. This is bounded above by $2 + \log_2(n)$, since j is never larger than n. Multiplying by the previous bound for the expected random integer complexity, we get an upper bound of $n \log_2(n) + \mathcal{O}(n)$. This is asymptotically tight because the expected number of random bits cannot be less than $\log_2(f_k(n, 0))$, which is also $n \log_2(n) + \mathcal{O}(n)$.

Proposition 24. *The expected random bit complexity of algorithm* UniformPerm0*(n,k) is* $n \log_2(n) + \mathcal{O}(n)$*.*

Finally, from the discussion of the beginning of this section, we deduce directly the following corollary:

Corollary 25. *It is possible to generate a uniform random permutation of size* n *with exactly* ℓ k*-cycles with expected random integer complexity* $n + \mathcal{O}(1)$*.*

5.3 Sampling Poisson random variates

A second consequence of our generation tree is a combinatorial algorithm to sample from the Poisson distribution with parameter $1/k$, that is, obtaining a random variate that takes each integer value $\ell \geqslant 0$ with probability $1/(e^{1/k} k^{\ell} \ell!)$.

Many sampling methods already exist for the Poisson distribution (see [4] for most of them). These algorithms are efficient but use rational and irrational numbers. By contrast, our algorithm is purely combinatorial in nature and only uses small integer numbers.

Our algorithm Poisson-$1/k$ is pretty simple once the generation tree is known: perform a random descent in the generation tree (by which we mean, start from the root, and at each level pick a uniform random child of the current node), until a non-special permutation is reached; then output its number of fixed points.

We have already computed the expected random integer complexity of this algorithm in the proof of Proposition 22, as

$$\mathbb{E}(T' - 1) = H_{k-1}\frac{e^{2/k} + 1}{2} + \frac{e^{2/k} - 1}{2}.$$

Algorithm 1 Poisson1

```
$n \leftarrow 1$, $g \leftarrow 0$, $k \leftarrow 1$
loop
    $i \leftarrow$ Random$(n+1)$
    if $i = n+1$ then
        $k \leftarrow k+1$
    else if $i > g$ then
        $k \leftarrow k-1$, $g \leftarrow n+1$
    else
        return $k$
    $n \leftarrow n+1$
```

As stated in the introduction, since we know that the limit distribution of the number of k-cycles of uniform permutations of size n is the Poisson distribution with expectation $1/k$, we deduce the correctness of the algorithm from the fact that it returns precisely the limit of such a vector $(\sigma_n)_{n\geqslant 1}$ – recall that after reaching a non special permutation, further descents in the tree yield permutations with the same number of k-cycles.

For the case $k = 1$, special permutations have a particularly simple form, and the algorithm Poisson1 is even simpler. It only needs to keep track of 3 integers representing the size of the permutation, the largest non-fixed point and the current number of fixed points.

From a Poisson distributed variable X_1 of expectation 1, it is easy to deduce another random variable $X_{1/k}$, which follows this time the Poisson distribution with mean $1/k$: $X_{1/k}$ is the number of heads in X_1 biased coin flips with parameter $1/k$. For any k, the number of times the `Random()` primitive has to be used in such an algorithm is given by $1 + (e^2 - 1)/2 = (e^2 + 1)/2 \approx 4.20$; comparing this with the expected random integer complexity of Poisson-$1/k$ shows that the latter is more efficient for $k \leqslant 32$.

We now turn to a short analysis of the random bit complexity of our Poisson1 algorithm.

Proposition 26. *The* Poisson1 *algorithm uses, in expectation, between* 6.89 *and* 6.9 *random bits, assuming the* `Random` *primitive is optimal in terms of random bits used.*

Proof. Let B be the number of random bits used by the algorithm and B_n the number of random bits (if any) consumed at level n of the tree, so that $C = \sum_{n=2}^{\infty} C_n$.

We have $\mathbb{E}(B_n) = \frac{2^{n-2}}{(n-1)!}\mathbb{E}(U_n)$, where U_n is the number of bits used by an optimal uniform sampler `Random`(n) in the sense of [11]; the $2^{n-2}/(n-1)!$ factor is simply the probability of reaching level n of the tree. The inequality $\log_2(n) \leqslant \mathbb{E}(U_n) \leqslant \log_2(n) + 2$ is valid for all n, but using it blindly to bound the complexity of our algorithm gives an upper bound of 9.59, significantly more bits than we want.

Instead, we compute exactly $D_m = \sum_{n=2}^{m} \frac{2^{n-2}}{(n-1)!}\mathbb{E}(U_n)$ for some constant m. The lower and upper bounds for $\mathbb{E}(C)$ then become $D_m + \sum_{n\geqslant m} \frac{2^{n-1}\log_2(n+1)}{n!}$ and $D_m +$

$\sum_{n\geqslant m}\frac{2^{n-1}(2+\log_2(n+1))}{n!}$, respectively. For fixed n, $\mathbb{E}(U_n)$ can be simply obtained from the binary expansion of $1/n$, as described in [11].

To obtain the estimates given in the proposition, we computed the above bounds for $m = 10$. Values of $\mathbb{E}(U_n)$ for n from 2 to 10 are $1, 8/3, 2, 18/5, 11/3$, $24/7, 3, 14/3, 23/5$, yielding $D_{10} = \frac{97771}{14175} \simeq 6.897$. This is a lower bound for the expected random bit complexity.

For the upper bound, we need an upper bound for

$$M = \sum_{n\geqslant 10} \frac{2^{n-1}(2+\log_2(n+1))}{n!}.$$

We only use the inequality $2 + \log_2(n + 1) \leqslant n$ (which is certainly valid for $n \geqslant 10$); we get

$$M \leqslant \sum_{n\geqslant 9} \frac{2^n}{n!} = e^2 - \sum_{n=0}^{8} \frac{2^n}{n!} = e^2 - \frac{2327}{315} \simeq 0.00175.$$

Indeed, both the lower and upper bound are between 6.89 and 6.9. □

Remark 27. In algorithm Poisson1, we can replace the call to the random sampler in the algorithm by an optimal ternary generator choosing the first option with probability $1/(n+1)$, the second one with probability $(n-g)/(n+1)$ and the third one with probability $g/(n + 1)$. In such a situation, analogous bound computations as in the previous proof gives an expected bit complexity of 5.12.

Our Poisson algorithm is not an optimal algorithm regarding its bit consumption; again applying the results in [11], an optimal algorithm would use, in expectation, fewer than $2 + E$ bits, where $E = 1 + e^{-1}\sum_{j\geqslant 0}\log_2(j)/j! \simeq 1.89$ is the binary entropy of the Poisson distribution; note, however, that such an algorithm would need the binary expansions of all individual probabilities of the Poisson distribution, $p_j = 1/(j!e)$.

Once again, computing $\sum_{i=0}^{B} i/2^i \sum_{j=0}^{A}[p_j]_i$ for $[p_j]_i$ the i-th bit of the binary expansion of probability p_j, and some constants A and B, gives a lower bound on the random bit complexity of an optimal algorithm for the Poisson distribution. For $A = 10$ and $B = 32$, we get a lower bound of 3.66, indicating our combinatorial algorithm is within a factor 2 of the optimal cost.

6 Concluding remarks

We have described a new uniform generation tree for permutations, with the property of preserving as much as possible the number of k-cycles between a permutation and its parent. This generation tree gives us a new and efficient algorithm for the uniform random generation of permutations with exactly ℓ k-cycles, as well as a new method for sampling from the Poisson distribution with parameter $1/k$, both for any $k \geqslant 1$. Both algorithms use only small integer numbers.

We believe other applications to random generation can be devised for the generation tree, such as sampling permutations having some predefined condition on their number of

k-cycles: for example, an even number. We expect our tree to make many rejection-based algorithms for this kind of task particularly efficient.

More generally, we are eager to see if such a construction can be adapted to other families of combinatorial objects, keeping in mind the potential gain for new efficient generation algorithms for such objects. Indeed, the generation process described in Section 5.2 replaces a multiplicative constant by an additive constant, in the expected cost of the algorithm. For combinatorial objects representing an exponentially (in k) small part of an easy to sample set, using an adaptation of our method may lead to still generate those objects in time proportional to $n + \mathcal{O}(1)$ instead of the exponential multiplicative cost induced by a classic rejection algorithm.

References

[1] Jörg Arndt. *Generating Random Permutations.* PhD thesis, Australian National University, 2009.

[2] Richard Arratia and Simon Tavare. The cycle structure of random permutations. *The Annals of Probability*, pages 1567–1591, 1992.

[3] Jacques Désarménien. Une autre interprétation du nombre de dérangements. In *Actes 8e Sém. Lothar. Combin.*, pages 11–16. IRMA, Strasbourg, 1984.

[4] Luc Devroye. *Non-Uniform Random Variate Generation.* Springer Verlag, 1986.

[5] Persi Diaconis and Mehrdad Shahshahani. On the eigenvalues of random matrices. *Journal of Applied Probability*, pages 49–62, 1994.

[6] Philippe Duchon and Romaric Duvignau. A new generation tree for permutations, preserving the number of fixed points. In *26th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2014)*, pages 679–690. DMTCS Proceedings, 2014.

[7] Richard Durstenfeld. Algorithm 235: Random permutation. *Commun. ACM*, 7(7):420, July 1964.

[8] Romaric Duvignau. *Maintenance et simulation de graphes aléatoires dynamiques (Maintenance and simulation of dynamic random graphs).* PhD thesis, University of Bordeaux, 2015.

[9] R. A. Fisher and F. Yates. *Statistical Tables for Biological, Agricultural and Medical Research.* Edinburgh: Oliver and Boyd, 1938.

[10] Donald E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[11] Donald E Knuth and Andrew C Yao. The complexity of nonuniform random number generation. In *Proceedings of Symposium on Algorithms and Complexity*, pages 357–428. Academic Press, New York, 1976.

[12] Conrado Martínez, Alois Panholzer, and Helmut Prodinger. Generating random derangements. In *I. Munro, R. Sedgewick, W. Szpankowski, and D. Wagner, editors, Proc. of the 10th ACM-SIAM Workshop on Algorithm Engineering and Experiments (ALENEX) and the 5th ACM-SIAM Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 234–240. SIAM, 2008.

[13] Fanja Rakotondrajao. k-fixed-points-permutations. In *INTEGERS: Electronic Journal of Combinatorial Number Theory*, volume 7, 2007. A36.

[14] Herbert S Wilf. *generatingfunctionology*. Elsevier, 2013.