A combinatorial proof of Aldous–Broder theorem for general Markov chains

Luis Fredes, Jean-François Marckert

A combinatorial proof of Aldous–Broder theorem for general Markov chains

Luis Fredes[†] **and Jean-François Marckert**[*]
[†]Université Paris-Saclay.
[*]CNRS, LaBRI, Université Bordeaux

Abstract

Aldous–Broder algorithm is a famous algorithm used to sample a uniform spanning tree of any finite connected graph G, but it is more general: given an irreducible and reversible Markov chain M on G started at r, the tree rooted at r formed by the first entrance steps in each node (different from the root) has a probability proportional to $\prod_{e=(e^-,e^+)\in \mathsf{Edges}(t,r)} M_{e^-,e^+}$, where the edges are directed toward r. In this paper we give proofs of Aldous–Broder theorem in the general case, where the kernel M is irreducible but not assumed to be reversible (this generalized version appeared recently in Hu, Lyons and Tang [5]).

We provide two new proofs: an adaptation of the classical argument, which is purely probabilistic, and a new proof based on combinatorial arguments. On the way we introduce a new combinatorial object that we call the golf sequences.

Acknowledgments

The first author acknowledge support from ERC 740943 GeoBrown.

1 Introduction

Consider $G = (V, E)$ a finite connected graph: V is the set of vertices, and E the set of undirected edges. A path is a sequence of vertices $w = (w_k, 0 \le k \le m)$, with the property that $\{w_k, w_{k+1}\} \in E$ for every k. This path is said to be covering if it visits all the vertices and if m is the first time with this property. More formally, for every $1 \le k \le |V|$ define

$$\tau_k(w) = \inf\{j, |\{w_0, \cdots, w_j\}| = k\}, \quad 1 \le k \le |V|,$$

the first time the path has visited k different points (we write τ_k instead of $\tau_k(w)$ when it is clear from the context). The path w is then called covering if $\tau_{|V|}(w) = m$.

Denote by $\mathsf{SpanningTrees}(G)$ the set of spanning trees t of G, and by $\mathsf{SpanningTrees}^\bullet(G)$ the set of rooted spanning trees (t, r), where r, the root, is a distinguished vertex of V. Each tree (t, r) is considered as a directed graph in which the edges are directed toward the root: for any (e^-, e^+) in the edge set $E(t, r)$, e^+ is the parent of e^- in t (we write $e^+ = p(e^-)$).

Definition 1. *For a covering path w, denote by* $\mathsf{FirstEntranceTree}(w)$ *the rooted spanning tree (t, w_0) whose $|V| - 1$ edges are $(w_{\tau_k}, w_{\tau_k - 1})$ (that is oriented towards the root w_0) for $2 \le k \le \tau_{|V|}$.*

Thus, each edge corresponds to the first edge used to visit each vertex different from w_0, turned around.

A Markov kernel $M = (M_{a,b}, a, b \in V)$ on V is said to be *positive on* G if $M_{a,b} > 0 \iff \{a, b\} \in E$. Any Markov chain driven by such a Markov kernel can traverse each edge of G both ways, and each step of such a chain corresponds to an edge of G. On a connected graph, as it is the case here, this chain is irreducible, therefore it has a unique invariant distribution, $\rho = (\rho_v, v \in V)$.

A model of random covering paths is obtained by killing a Markov chain W at the cover time.

Theorem 2. *[Aldous–Broder, [1] and [2]]Let W be a Markov chain with positive and **reversible** kernel M and invariant distribution ρ. Then, for any $(t,r) \in \mathsf{SpanningTrees}^\bullet(G)$*

$$\mathbb{P}\left[\mathsf{FirstEntranceTree}\left(W_0,\cdots,W_{\tau_{|V|}}\right)=(t,r)\mid W_0=r\right]=\mathsf{Const.}\left[\prod_{e\in E(t,r)}M_e\right]/\rho(r). \tag{1}$$

To be totally clear, each e is an edge of the type (e^-,e^+), where $e^+=p(e^-)$, and $M_e=M_{e^-,e^+}$.

Elements around Aldous–Broder algorithm can be found in [6].

As a consequence if M is the Markov kernel corresponding to the simple random walk on G, $M_{a,b}=1_{\{a,b\}\in E}/\mathsf{deg}_G(a)$, and since the invariant distribution of this simple random walk ρ_v is proportional to $\mathsf{deg}_G(v)$, the r.h.s. of (1) is proportional to $\prod_{u\in V}1/\mathsf{deg}_G(u)$, that is independent of t, so that $\mathsf{FirstEntranceTree}(W_0,\cdots,W_{\tau_{|V|}})$ is a uniform spanning tree[1], rooted at $W_0=r$.

As we will see, Aldous–Broder theorem **does not hold** if we remove the reversibility condition.

Recall that if $X=(X_i,i\in\mathbb{Z})$ is a Markov chain with positive kernel M with invariant distribution ρ, under its stationary regime, then the time reversal of X is also a Markov chain under its stationary regime, with same invariant distribution, and with kernel

$$\overleftarrow{M}_{x,y}:=\rho_y M_{y,x}/\rho_x,\ \text{for all } (x,y)\in V^2. \tag{2}$$

The next result extends Aldous–Broder theorem:

Theorem 3. *Let W be a Markov chain with positive kernel M (**reversible or not**) with invariant distribution ρ. Then, for any $(t,r) \in \mathsf{SpanningTrees}^\bullet(G)$*

$$\mathbb{P}\left[\mathsf{FirstEntranceTree}\left(W_0,\cdots,W_{\tau_{|V|}}\right)=(t,r)\mid W_0=r\right]=\mathsf{Const.}\left[\prod_{e\in E(t,r)}\overleftarrow{M}_e\right]/\rho(r). \tag{3}$$

The aim of this paper, is to give two new proofs of this result. This theorem has been proved by Hu, Lyons and Tang [5] shortly before our work; but, at the moment we were writing the present paper, this result was not present on the first version of [5], on the arxiv.

• This theorem implies Aldous–Broder result since in the reversible case, $\overleftarrow{M}=M$.

• We will prove that if W is taken **under its stationary regime** (meaning that $W_0\sim\rho$) then

$$\mathbb{P}\left[\mathsf{FirstEntranceTree}\left(W_0,\cdots,W_{\tau_{|V|}}\right)=(t,r)\right]=\mathsf{Const.}\left[\prod_{e\in E(t,r)}\overleftarrow{M}_e\right]. \tag{4}$$

This formula is equivalent to (3).

• From our point of view, Aldous–Broder theorem should be stated like Theorem 3, since this time reversal is not only a tool for the (original) proof of the Theorem but a characteristic of its conclusion. As a matter of fact, it is probably common for researchers in the field to try to find a coupling between Wilson construction [10, 8] of the spanning tree (which is faster) and Aldous–Broder's, but the fact that Wilson algorithm produces a tree with distribution $\mathsf{Const.}\prod_{e\in E(t,r)}M_e$ shows that it is likely not possible, unless ones use Wilson algorithm with $\overleftarrow{M}$ instead of M.

[1] To conclude here, another argument is needed: the fact that the support of $\mathsf{FirstEntranceTree}(W_0,\cdots,W_{\tau_{|V|}})$ is the set of all spanning trees, which is easy to see

• Both theorems above are valid on multigraph, which are the analogous of graph in which multiple edges between vertices are allowed, as well as loops (edges $\{a,a\}$ adjacent to a single node are allowed). All the proofs we give are valid in this settings too (even if tiny adjustments to treat multiple edges could be needed at some places).

Remark 4. *In general,* $\prod_{e\in E(t,r)} M_e$ *and* $\prod_{e\in E(t,r)} \overleftarrow{M}_e$ *are different. Let us write* $\prod_{u\in t}$ *for a product over the set of vertices of* t*. For* $(t,r)\in \mathsf{SpanningTrees}^\bullet(G)$*,*

$$\prod_{e\in E(t,r)} M_e = \prod_{u\in t\setminus\{r\}} M_{u,p(u)} = \mathsf{Const.}\ \rho_r \prod_{u\in t\setminus\{r\}} \rho_u M_{u,p(u)}$$

while by (2), and since in a tree all nodes but the root are children of some nodes,

$$\prod_{e\in E(t,r)} \overleftarrow{M_e} \;=\; \prod_{u\in t\setminus\{r\}} \left[M_{p(u),u}\,\rho_{p(u)}/\rho_u\right] = \mathsf{Const.}\ \rho_r \prod_{u\in t\setminus\{r\}} \rho_{p(u)} M_{p(u),u}.$$

so that these quantities are different when ρ *is not reversible with respect to* M*.*

Different formula does not imply different values, but a graph of size 3 is sufficient to observe that these formulas are indeed different: consider

$$M=\begin{bmatrix} 0 & 1/3 & 2/3\\ 1/5 & 0 & 4/5\\ 1/7 & 6/7 & 0\end{bmatrix},\quad \rho=\frac{1}{226}\begin{bmatrix}33,95,98\end{bmatrix}\quad \textit{and}\quad \overleftarrow{M}=\begin{bmatrix} 0 & 19/33 & 14/33\\ 11/95 & 0 & 84/95\\ 11/49 & 38/49 & 0\end{bmatrix}.$$

It can be verified that ρ *is the invariant measure of* M *and that* $\overleftarrow{M}$ *is the reversal kernel. The tree* (t,r) *defined by* $(2,1)$ *and* $(3,1)$ *has weights* $M_{2,1}M_{3,1}=1/35\neq \overleftarrow{M}_{2,1}\overleftarrow{M}_{3,1}=121/4655$*.*

Let us say some words on the paths $(w_k, 0\le k\le \tau_{|V|})$ that satisfy $\mathsf{FirstEntranceTree}\left(w_0,\cdots,w_{\tau_{|V|}}\right)=(t,r)$. Such a path starts at $w_0=r$ and eventually reaches each vertex u of $V\setminus\{r\}$ for the first time from $p(u)$ its parent in (t,r). In other words, $\mathsf{FirstEntranceTree}\left(w_0,\cdots,w_{\tau_{|V|}}\right)=(t,r)$ iff

$$\{(w_{\tau_k}, w_{\tau_k-1}), 2\le k\le |V|\} = \{(u,p(u)), u\in V\setminus\{r\}\}, \tag{5}$$

meaning that $k\mapsto w(\tau_k)$ induces a decreasing labelling towards the root.

The analysis of $\mathbb{P}(\mathsf{FirstEntranceTree}\left(W_0,\cdots,W_{\tau_{|V|}}\right)=(t,r))$ from this kind of considerations leads to sum on all the paths according to the order at which the vertices are reached. Since this direct approach of Theorem 3 seems difficult to complete, the initial proofs by both Aldous and Broder of their theorems follow a tricky path which is purely probabilistic. In our new proof, we avoid also the direct treatment of the labellings discussed above, but produce instead a combinatorial argument, relying on a new object we introduce, the golf sequence, which, we believe is interesting in its own.

1.1 Some recalls from combinatorics

The new proof uses many ingredients of the combinatorics folklore, and we must say that we have been deeply inspired by Zeilberger short paper [11] where many results from linear algebra are proved by means of combinatorial tools, notably, Foata's proof of MacMahon's master theorem [11, 3] and the famous

Theorem 5 (Matrix Tree Theorem)**.**

$$\det(\mathrm{Id} - M^{(r)}) = \sum_{(t,r)\in \mathsf{SpanningTrees}^\bullet(G)} \prod_{e\in E(t,r)} M_e, \tag{6}$$

where $M^{(r)}$ is the matrix M deprived of the line r and column r, and Id *the identity matrix with the same size as that of $M^{(r)}$.*

We borrow the argument from Zeilberger [11] since it helps to present the arguments of our own proof.

Proof. First, a cycle is a (class of equivalence of) path $(w_0, \cdots, w_m)$ for some $m \geq 1$, such that $w_m = w_0$ and $|\{w_0, \cdots, w_m\}| = m$ (it is simple). Two cycles are equivalent if one can be obtained from the other by shifting its indices in $\mathbb{Z}/m\mathbb{Z}$.

Observe that

$$\mathrm{Id} - M^{(r)} = \left[\left(\sum_{v'\in V} M_{u,v'}\right)\mathbf{1}_{(u=v)} - M_{u,v}\right]_{(u,v)\in(V\setminus\{r\})^2}. \tag{7}$$

Denote by $\mathcal{B}$ the set of pairs (B, C) such that:
• B is a directed graph on V, where each vertex of $V \setminus \{r\}$ has either 0 or 1 outgoing edge ending in V (including $\{r\}$ this time). Denote by V_B the set of vertices from which there is an outgoing edge.
• C is a collection of directed disjoint cycles on $V_C = (V \setminus \{r\}) \setminus V_B$.

Set $\mathsf{Weight}(B) := \prod_{u\in V_B} M_{u,t(u)}$ where $t(u)$ is the target of the edge starting at u; $\mathsf{Weight}(C) := (-1)^{N(C)} \prod_{c \text{ cycles of C}} \prod_{e\in c} M_{e^-,e^+}$ the product of the weight of edges along the directed cycles of C, where $N(C)$ the number of cycles of C and finally define $\mathsf{Weight}(B, C) := \mathsf{Weight}(B)\mathsf{Weight}(C)$.

An expansion of (7) allows one to see that $\mathsf{Weight}(\mathcal{B}) := \sum_{(B,C)\in\mathcal{B}} \mathsf{Weight}(B, C) = \det(\mathrm{Id} - M^{(r)})$. Here is the argument: first, expand $\det(\mathrm{Id} - M^{(r)})$ using Leibniz formula:

$$\det(\mathrm{Id} - M^{(r)}) = \sum (-1)^{\mathsf{sign}(\sigma)} \prod_{i\neq r} (\mathrm{Id} - M^{(r)})_{i,\sigma(i)},$$

where the sum range on all permutations σ on $V \setminus \{r\}$. Now, consider the set $F(\sigma) = \{i : \sigma(i) = i\}$ of fix points of σ, and rewrite, by (7):

$$\prod_{i\neq r} (\mathrm{Id} - M^{(r)})_{i,\sigma(i)} = \left(\prod_{i\in F(\sigma)} \left(-M_{i,i} + \sum_{j\in V} M_{i,j}\right)\right)\left(\prod_{i\in V\setminus(\{r\}\cup F(\sigma))} -M_{i,\sigma(i)}\right).$$

The second parenthesis can be interpreted as the weight of cycles of σ with lengths at least 2. Now expand the first parenthesis (without simplifying $M_{i,i}$, so that $M_{i,i}$ comes with a sign $+$, and in another term, with a sign $-$). This first parenthesis can be rewritten as a sum over $A \subset F(\sigma)$ as follows:

$$\prod_{i\in F(\sigma)} \left(-M_{i,i} + \sum_{j\in V} M_{i,j}\right) = \sum_{A\subset F(\sigma)} \left(\prod_{i\in A} (-M_{i,i})\right)\left(\prod_{i\in F(\sigma)\setminus A} \sum_j M_{i,j}\right).$$

Each factor $-M_{i,i}$ can be seen to be the weight of a loop over i (that is a cycle of size 1), and by expansion, $\prod_{i\in F(\sigma)\setminus A} \sum_{j\in V} M_{i,j}$ can be interpreted as the weight of a directed graph where each vertex of $F(\sigma) \setminus A$ has a single outgoing edge, ending on any vertex of V. This ends the argument explaining why $\mathsf{Weight}(\mathcal{B}) = \det(\mathrm{Id} - M^{(r)})$.
– We claim now that $\mathsf{Weight}(\mathcal{B}) = \sum_{(t,r)\in \mathsf{SpanningTrees}^\bullet(G)} \prod_{e\in E(t,r)} M_e$. The graphs "$B$" are made of cycles and trees, and C is made of cycles (with a sign of the weight corresponding to parity). For any

pair (B, C) having (totally) at least one cycle one can define (B', C') as follows: for a total order on the set of oriented cycles, take the greatest cycle c in the union of B and C. Denote by (B', C') the pair obtained by moving c from the component containing it to the other. This map $(B, C) \to (B', C')$ is clearly an involution and satisfies $\mathsf{Weight}(B', C') = -\mathsf{Weight}(B, C)$. Hence, $\mathsf{Weight}(\mathcal{B})$ coincides with the sum of the $\mathsf{Weight}(B, C)$ taken on the set of pairs (B, C) which have no cycles: C is empty, and the graph B has no cycle, and since its number of edges is one less than its number of vertices, it is a spanning tree. □

The quantity $\det\left(\mathrm{Id} - M^{(r)}\right)$ **is the central algebraic object;** it is somehow the partition function of the family of weighted spanning trees under inspection, but not only. It will appear under the following forms at several places of the paper.

Notice that the direct expansion of the determinant (using permutations) gives

$$\det(\mathrm{Id} - M^{(r)}) = \sum_{C} (-1)^{N(C)} \prod_{c \text{ cycles of C}} \prod_{e \in c} M_e \tag{8}$$

where the sum ranges over all sets C of disjoint (oriented) cycles of length ≥ 1 on the set of vertices $V \setminus \{r\}$.
Now by considering the involution that changes the orientation of every cycle one gets

$$\det(\mathrm{Id} - M^{(r)}) = \det(\mathrm{Id} - \overleftarrow{M}^{(r)})$$

and therefore for a fixed $r \in V$

$$\sum_{(t,r) \in \mathsf{SpanningTrees}^{\bullet}(G)} \prod_{e \in E(t,r)} M_e = \sum_{(t,r) \in \mathsf{SpanningTrees}^{\bullet}(G)} \prod_{e \in E(t,r)} \overleftarrow{M}_e.$$

This together with Remark 4 say that even though the terms associated to each tree (t, r) may be different, these sums are equal (similar identities are present in Hu & al. [5]).

Even more, the invariant distribution ρ of the Markov kernels M and $\overleftarrow{M}$ is related to these quantities by:

$$\rho_w = \mathsf{Const.} \det(\mathrm{Id} - M^{(w)}) = \mathsf{Const.} \det(\mathrm{Id} - \overleftarrow{M}^{(w)}) \tag{9}$$

so that, from the matrix tree theorem, this provides a connection between ρ_w and the total mass of spanning trees rooted at w. To prove (9), there are several methods, one of them being the use of Theorem 2 or 3, but direct arguments exist (and are classical). One wants to solve the vector system $\left\{ \rho(\mathrm{Id} - M) = 0, \rho \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = 1 \right\}$. Assume here that the vertex are labeled from 1 to n. Since the sum by rows of $\mathrm{Id} - M$ is zero, the system is over-determined, so we discard the equation $(\rho(\mathrm{Id} - M))_{1,n} = 0$. Then, the row vector ρ is solution of

$$\rho Q = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \text{ where } Q = \left[\begin{array}{c|c} \left[(\mathrm{Id} - M)_{i,j}\right]_{i \leq n, j \leq n-1} & \begin{matrix} 1 \\ \vdots \\ 1 \end{matrix} \end{array}\right]. \tag{10}$$

By symmetry, it suffices to prove that ρ_n satisfies (9). For this use the Cramer rule to get (9): notice that replacing the n-th row of Q by $[0, \cdots, 0, 1]$ we obtain the wanted determinant in the numerator (as

a cofactor). It remains to show for the denominator that $\det(Q) = \sum_w \det(\mathrm{Id} - M^{(w)})$; the following argument is classical. It suffices to expand $\det(Q)$ according to the last column and show that

$$\det((\mathrm{Id} - M)_{(i,j):i\neq a, j\neq n})(-1)^{n-a} = \det((\mathrm{Id} - M)_{(i,j):i\neq a, j\neq a}) = \det(\mathrm{Id} - M^{(a)}). \tag{11}$$

This first equality is more general: in fact, $\det((\mathrm{Id} - M)_{(i,j):i\neq a, j\neq b})(-1)^{b-a} = \det((\mathrm{Id} - M)_{(i,j):i\neq a, j\neq a})$ for all $b \in \{1, \cdots, n\}$, and this independence to the deprived column b (up to the sign) is a consequence of the fact that the sum of the columns of $\mathrm{Id} - M$ is 0 (this property allows to play with the lacking column b: take the a-th column C_a and replace it by $C_a + \sum_{k\neq a,b} C_k = \sum_{k\neq b} C_k = -C_b$. This replacement does not alter the determinant, and now, we have $-C_b$ at position a).

2 New combinatorial proof of Theorem 3

We will assume all along the proof that $|V| \geq 2$, otherwise the statement is trivial.
Theorem 3 expresses $\mathbb{P}(\mathsf{FirstEntranceTree}(W_i, 0 \leq i \leq \tau_{|V|}(W)) = (t, r))$ in terms of $\overleftarrow{M}$, but since W is a Markov chain with kernel M, we need also to consider reverse paths in the proof. For a path $w = (w_0, \cdots, w_m)$ on G denote by $|w| = m$ its number of steps, $\overline{w} = w_m$ its last position, and $\underline{w} = w_0$ its first position. Denote by $\overleftarrow{w} = (w_m, w_{m-1}, \cdots, w_0)$ the reverse time path and set

$$\mathsf{Weight}(w) := \prod_{i=0}^{|w|-1} \overleftarrow{M}_{w_{i+1}, w_i}, \tag{12}$$

which is the weight of $\overleftarrow{w}$ starting at $\overline{w}$ taken under the kernel $\overleftarrow{M}$.

A *heap h of directed outgoing edges of a vertex u* is a finite sequence

$$h = [(u, t_i), 1 \leq i \leq \ell]$$

for some $\ell \geq 0$, where, for each i, $\{u, t_i\} \in E$. The set of such heaps is $\cup_{k\geq 0} O_u^k$ where $O_u = \{(u, v) : \{u, v\} \in E\}$ is the set of oriented outgoing edges out of u (in G). A *collection of heaps* is a sequence $H := (H_u, u \in V)$ of heaps, where $H_u \in \cup_{k\geq 0} O_u^k$ is a heap of directed edges out of u (see Fig. 1, picture 2). The set of collections of heaps is the product set

$$\mathsf{HeapCol} = \prod_{u\in V} \Big(\bigcup_k O_u^k \Big).$$

Each element of $\mathsf{HeapCol}$ can be seen as a multigraph (repetition of edges permitted), in which the sequence of outgoing edges out of each node is equipped with a total order. We will use collections of heaps to encode various sorts of family of connected paths. As a combinatorial tool, they allow some rearrangements of steps, and then provide some ways to construct bijections efficiently.

The concatenation operation $\oplus$ of heaps is defined as usual:

$$[e_1, \cdots, e_m] \oplus [e'_1, \cdots, e'_{m'}] = [e_1, \cdots, e_m, e'_1, \cdots, e'_{m'}]$$

which extends to collection of heaps as expected : $H \oplus H' := (H_u \oplus H'_u, u \in V)$.
We say that h is a prefix of $h \oplus h'$, notion which extends again to collections.

The weight of a heap $h = [(s_i, t_i), 1 \leq i \leq m]$ and of a collection $H = (H_u, u \in V)$ are

$$\begin{aligned} \mathsf{Weight}(h) &:= \prod_{i=1}^{m} \overleftarrow{M}_{s_i, t_i}, \\ \mathsf{Weight}(H) &:= \prod_{u\in V} \mathsf{Weight}(H_u). \end{aligned}$$

The weight of an empty heap is set to 1.

Prefix removal (PR). For $h = h' \oplus h''$, the "prefix removal" of h' in h is possible. We write this operation $h \ominus_{PR} h' = h''$. This extends to collections: if $H = H' \oplus H''$, we set $H \ominus_{PR} H' = H''$.

Trees. The number of children $\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u)$ of a node u in (t, r) which is often called out-degree or simply degree in the literature in the case of trees, coincides with the number of incoming edges at u in (t, r) (the notation $\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u)$ is introduced to avoid any confusion with $\mathsf{deg}_G(u)$, the degree of the graph at u which comes into play too, and with the total degree of u in t). A node u with $\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) = 0$ is called a leaf, and the set of leaves is denoted ∂t. Notice that since $|V| \geq 2$, r is not a leaf. The elements in the set $t^o = V \setminus \partial t$ are called internal nodes.

2.1 Step 1. Collection of heaps encoding of a path

Definition 6. *The passport of a collection of heaps $H = (H_u, u \in V) \in \mathsf{HeapCol}$ is the pair*

$$\mathsf{OUTIN}(H) := [(\mathsf{Out}_u(H), u \in V), (\mathsf{In}_u(H), u \in V)]$$

where, for all $u \in V$, $\mathsf{Out}_u(H) := |H_u|$ records the number of outgoing edges of u and

$$\mathsf{In}_u(H) \quad := \quad \sum_{w \in V} |(w, u) \text{ in } H_w|$$

records the number of incoming edges to u (with multiplicities).

As a standard passport in real life, the passport is used to record the number of entries/exits at each node, with multiplicities. Different collections of heaps can have the same passport.

Denote by $\mathsf{Paths}(r)$ the set of paths w starting at $w_0 = r$ (with any finite length). With any path $w \in \mathsf{Paths}(r)$ associate the collection of heaps $\mathsf{Heap}(w) := (\mathsf{Heap}_u(w), u \in V) \in \mathsf{HeapCol}$, where

$$\mathsf{Heap}_u(w) := [(w_{j+1}, w_j) \ : 0 \leq j \leq |w| - 1, w_{j+1} = u] \, ;$$

these edges are then taken according to their order in w, and they correspond to edges that reach u, that are turned around. It can be useful to consider them as steps outgoing from u in $\overleftarrow{w}$, still taken according to their order in w (see Figure 1).

As warm up, let us prove the following classical fact.

Lemma 7. *The map*

$$\begin{array}{rccc} \mathsf{Heap}: & \mathsf{Paths}(r) & \longrightarrow & \mathsf{HeapCol} \\ & w & \longmapsto & \mathsf{Heap}(w) \end{array}$$

is a weight preserving injection (meaning that $\mathsf{Weight}(w) = \mathsf{Weight}(\mathsf{Heap}(w))$.

This is not a bijection since some elements of $\mathsf{HeapCol}$ do not encode any paths.

Proof. Let us take an element $h = (h_u, u \in V)$ in the image, and let us reconstruct its unique preimage w by the map Heap (in fact, we build $\overleftarrow{w}$). The total number of edges in h is the number of steps m of any walk w such that $\mathsf{Heap}(w) = h$. If $m = 0$, the statement is clear. Recall that the collection of heaps gives the steps of $\overleftarrow{w}$. Assume $m > 0$. This allows to recover the only possible value for the final point w_m too: this is the unique node w_m having one more outgoing edge than incoming (if such a node does not exist, the path ends and starts at r, and then $w_m = r$). Now, knowing the last position of the walk w_m, take (w_m, u) the rightmost (top see Figure 1) edge in Heap_{w_m}, record the identity of the node u, and remove the edge (w_m, u) from Heap_{w_m}. Now, set $w_{m-1} := u$. Iterate the construction until all the heaps are empty.
The statement about the weights easily follows. □

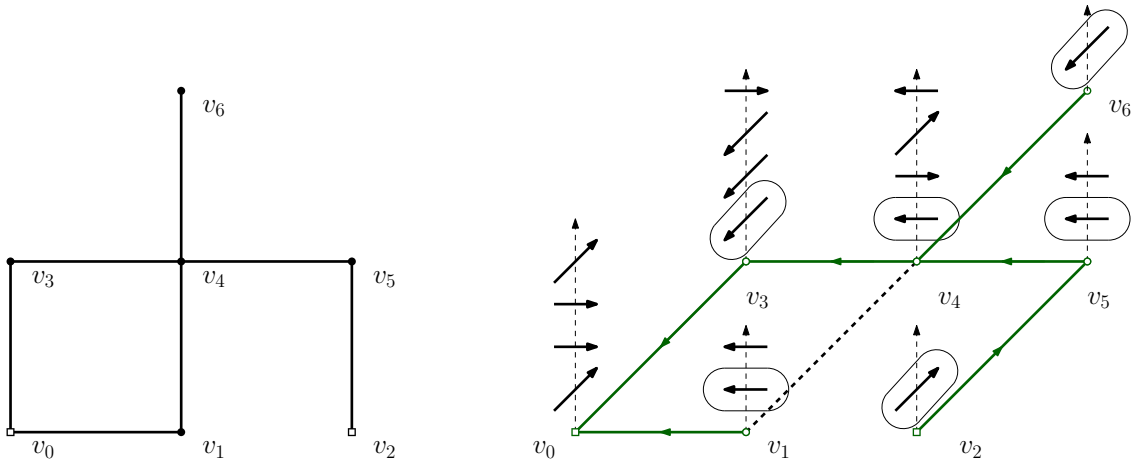

Figure 1: Left: graph G, where the walk $w = v_0v_3v_0v_1v_0v_1v_0v_3v_0v_3v_4v_5v_4v_6v_4v_3v_4v_5v_2$ is performed. Right: Heap associated to the path w; for example, $H_{v_3} = [(v_3, v_0), (v_3, v_0), (v_3, v_0), (v_3, v_4)]$. To read the steps of w in the heap, follow the edges from top to bottom to form $\overleftarrow{w}$, then transform $\overleftarrow{w}$ into w. The steps used to construct the tree are circled.

Definition 8. *(i) A collection of heaps $H \in \mathsf{HeapCol}$ with a balanced passport i.e.*

$$\mathsf{OUTIN}\,(H) = [(N_u, u \in V), (N_u, u \in V)], \tag{13}$$

for some $N_u \geq 0$, is called a heap of cycles. Denote by $\mathsf{HCycles}$ the set of heaps of cycles and by $\mathsf{HCycles}(\mathsf{free} \ni v)$ the heaps of cycles where $N_v = 0$, meaning that v is not an element of any cycle.

(ii) A heap of cycles $H \in \mathsf{HCycles}$ satisfying $N_u \leq 1$ for all u, is called trivial (it is a set of non intersecting cycles). We denote by $\mathsf{HCycles}^{\mathsf{Trivial}}(\mathsf{free} \ni v)$ the set of trivial heap of cycles for which $N_v = 0$ (that is, non incident to v).

Remark 9. *It is folklore to build "classical heap of cycles" from "our collection of heaps with balanced passport" see Viennot [9]. Classical heaps of cycles are equivalence classes of sequence of cycles $(c_1, \cdots, c_p)$ (were $p \geq 0$), where the relation is the transitive closure of the following relation: two sequences $(c_1, \cdots, c_p) \sim_R (c'_1, \cdots, c'_{p'})$ are equivalent if they coincide except for some non intersecting and consecutive cycles c_i and c_{i+1} and $(c_i, c_{i+1}) = (c'_{i+1}, c'_i)$. In words: non intersecting cycles commute, and intersecting cycles do not. Following Cartier & Foata [3] terminology, sequences of cycles with the partial commutation formula form a partially commutative monoid (for the concatenation operation), which can be geometrically represented using heap of "real" cycles (see Viennot [9] and Krattenthaler [7]).*

Giving these classical heap of cycles one can form the collection of heaps $\mathsf{Heap}(c_1) \oplus \cdots \oplus \mathsf{Heap}(c_p)$. To define the converse we need to introduce first, the $\mathsf{PopCycle}$ function. Assume that a collection of heaps $(H_u, u \in V)$ is given. For a vertex w_0 we define the procedure $\mathsf{PopCycle}(w_0)$ as: follow the path starting at w_0 and stop it the first time it forms a cycle c, that is the path $w_0, w_1, \cdots, w_j$ satisfies $w_i = w_j$ for some $i < j$ and j is the first index with this property; then return c the cycle formed by $(w_i, \cdots, w_j)$ and suppress the cycle from the heaps (that is all the edges $(w_i, w_{i+1}), \cdots, (w_{j-1}, w_j)$).

Now, to define the converse do the following: if all the heaps of the collection are empty, then associate the empty heap of cycles; otherwise, consider $w_0 = u$ a vertex with non-empty heap. Starting

from $i = 1$ repeat the following procedure until the heaps are all empty: define $c_i := \mathsf{PopCycle}(w_0)$, increment i and define $w_0 = u$, for a u chosen among those for which $H_u \neq \varnothing$ (see Figure 2 for a representation of this map). This procedure is well-defined since passports are balanced, i.e. the number of incoming edges is equal to the number of out-coming edges; and this property is preserved each time PopCycle *is used. At the end, the sequence $(c_1, \cdots, c_m)$ obtained depends on the order of the nodes u chosen, but it can be proved that for all choices of u, the final sequences of cycles are equal as heap of cycles.*

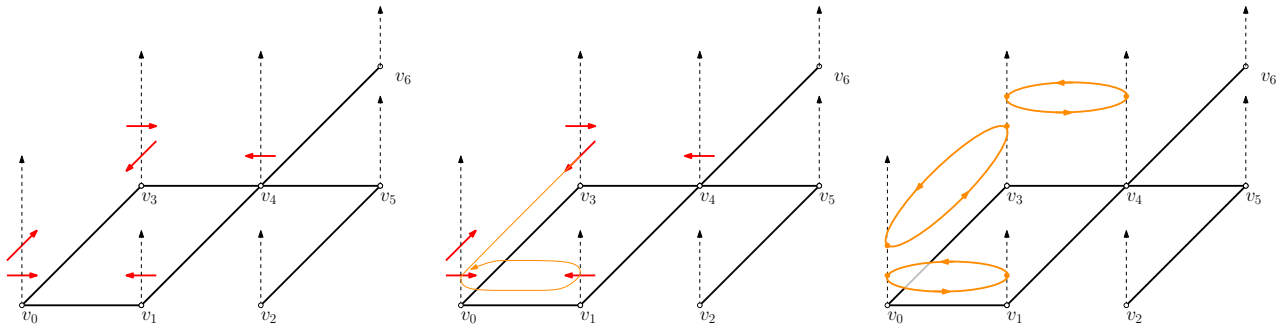

Figure 2: Left: representation of a heap of cycles as in Definition 8. Center: algorithm to form the classical heap of cycles started from v_3. Right: representation of a classical heap of cycles (there are some cycles on top of the others). Cycles here have length 2, but they may be of any length in general.

For a trivial heap of cycles $H = (H_u, u \in V)$, define

$$\mathsf{SignedWeight}(H) = \mathsf{Weight}(H)(-1)^{|\mathsf{Cycles}(H)|}.$$

We have the following inversion formula

$$\sum_{H \in \mathsf{HCycles}(\mathsf{free} \ni f)} \mathsf{Weight}(H) = \left(\sum_{H \in \mathsf{HCycles}^{\mathsf{Trivial}}(\mathsf{free} \ni f)} \mathsf{SignedWeight}(H) \right)^{-1} \overset{(8)}{=} \det\left(\mathrm{Id} - M^{(f)}\right)^{-1}. \quad (14)$$

This formula is famous in combinatorics (see [9, 7]). A simple proof : we will prove that the product of the sums of the two first members of (14) gives 1. For this associate to each pair (h, t) of heap of cycles and trivial heap of cycles (an element of $\mathsf{HCycles}(\mathsf{free} \ni f) \times \mathsf{HCycles}^{\mathsf{Trivial}}(\mathsf{free} \ni f)$), a pair (h', t') of the same type obtained as follows: Take the greatest cycle c in the union of t and of the set of cycles of h having no cycle above them and which moreover do not intersect the cycles of t. Now, to obtain (h', t') from (h, t), remove c from the component which contains it, and move it to the other (if it is move on h, it is placed on top of it). This operation is an involution which changes the sign of the weight, so that the total weight of the pairs (h, t) coincides with the weight of the single pair $(\varnothing, \varnothing)$ which is 1.

2.2 Step 2. Collection of heaps of a covering path corresponding to a given tree

Consider the subset of $\mathsf{Paths}(r)$,

$$\mathsf{Paths}(t, r, f) := \{w \text{ covering } : \mathsf{FirstEntranceTree}(w) = (t, r), \underline{w} = r, \overline{w} = f\}.$$

The passport of any path $w \in \mathsf{Paths}(t, r, f)$, satisfies,

$$\mathsf{OUTIN}(\mathsf{Heap}(w)) = [(N_u + 1_{u=f}, u \in V), (N_u + 1_{u=r}, u \in V)], \quad (15)$$

for $N_f = 0$, some $N_r \geq 0$, and some $N_u \geq 1$ for all $u \notin \{f, r\}$.

From the FirstEntranceTree construction, the last visited vertex is a leaf:

Lemma 10. *If f is an internal node of t, then* $\mathsf{Paths}(t, r, f) = \varnothing$.

All paths $w \in \mathsf{Paths}(t, r, f)$ have a step from $p(u)$ to u, for each node $u \neq r$. The oriented edges of (t, r) can be seen as a collection of heaps $H^t = (H^t_u, u \in V)$ defined by:

$$\left\{ \begin{array}{rcl} H^t_u & = & [(u, p(u)], \text{ for } u \neq r, \\ H^t_r & = & \varnothing, \end{array} \right.$$

whose passport is

$$\mathsf{OUTIN}(H^t) = \left[(1 - 1_{u=r}, u \in V), \left(\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u), u \in V \right) \right]. \tag{16}$$

By construction, the edges of H^t are the first edges of the collection $\mathsf{Heap}(w)$ (we put them according to their order in w), and then can be removed using the remove prefix operation; it produces the collection of "truncated heaps" defined by removing the edges of t from $\mathsf{Heap}(w)$

$$\mathsf{Heap}^{\mathsf{Trunc}}(w) := \mathsf{Heap}(w) \ominus_{PR} H^t(w). \tag{17}$$

If (15) holds, in view of (16), the passport of $\mathsf{Heap}^{\mathsf{Trunc}}(w)$ is

$$\mathsf{OUTIN}\left(\mathsf{Heap}^{\mathsf{Trunc}}(w) \right) = [(N_u + 1_{u=f} - (1 - 1_{u=r}), u \in V), (N_u + 1_{u=r} - \mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u), u \in V)]. \tag{18}$$

One of the main point of the proof is the following Lemma, which says that we completely know the set of collections of **truncated** heaps $\mathsf{Heap}^{\mathsf{Trunc}}[\mathsf{Paths}(t, r, f)]$. In words, they correspond, prior to truncation, to all collections $\mathsf{Heap}(w)$ satisfying (15) and large enough so that H^t can be removed. Set

$$\mathsf{HeapCol}[(o_u, u \in V), (i_u, u \in V)] := \{h \in \mathsf{HeapCol} : \mathsf{OUTIN}(h) = [(o_u, u \in V), (i_u, u \in V)]\}$$

Lemma 11. *If $f \in \partial t$, the set* $\mathsf{Heap}^{\mathsf{Trunc}}[\mathsf{Paths}(t, r, f)]$ *is equals to*

$$\Xi(t, r, f) := \bigcup \mathsf{HeapCol}\left[(n_u + (\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) - 1)\mathbf{1}_{u \in t^o}, u \in V), (n_u + \mathbf{1}_{u \in \partial t}(1 - 1_{u=f}), u \in V) \right] \tag{19}$$

where the union is taken on all $n_u \geq 0, \forall u \neq f$, and $n_f = 0$ (no incoming or outgoing edge at f, while $n_u + \mathbf{1}_{u \in \partial t}(1 - 1_{u=f}) > 0$ for the other leaves).

Lemma 11 describes the set of heaps that are not circled in Figure 1 (in all generality).

Proof. ■ Proof of $\mathsf{Heap}^{\mathsf{Trunc}}[\mathsf{Paths}(t, r, f)] \subset \Xi(t, r, f)$. This inclusion can be seen thanks to a change of variables from (18). Indeed, from (18) we must have $N_u - \mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) + 1_{u=r} \geq 0$ and $N_u + 1_{u=f} - 1 + 1_{u=r} \geq 0$, which implies

$$N_u \geq \left(\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) - 1_{u=r} \right) \mathbf{1}_{u \in t^o} + \mathbf{1}_{u \in \partial t}(1 - 1_{u=f})$$

since $\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) \geq 1$ when $u \in t^o$ and $\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) = 0$ when $u \in \partial t$. We then set

$$n_u := N_u - \left[\left(\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) - 1_{u=r} \right) \mathbf{1}_{u \in t^o} + \mathbf{1}_{u \in \partial t}(1 - 1_{u=f}) \right]$$

and compute the passport (18) after this change of variable to conclude this inclusion.
■ Now, take $h = (h_u, u \in V) \in \Xi(t, r, f)$ and consider $H := H^t \oplus h$ the potential unique collection

of heaps such that $\mathsf{Heap}^{\mathsf{Trunc}}[H] = h$. Let us prove that there exists a path $w \in \mathsf{Paths}(t, r, f)$ such that $\mathsf{Heap}(w) = H$. As explained in the proof of Lemma 7, the collection of heaps H can be used to construct a path: denote by p the current point and $\overleftarrow{w^\star}$ the current path which are initialized as $p = f$ and $\overleftarrow{w^\star} = (f)$.
(a) If H_p is empty then stop the construction and "return $\overleftarrow{w^\star}$". If H_p is not empty, consider the last edge (p, u) of the heap H_p.
(b) Remove (p, u) from H_p, add (p, u) as last step in $\overleftarrow{w^\star}$.
(c) Set $p = u$ and go to (a).

This construction stops when a vertex p with an empty heap H_p is reached. To conclude the proof, taking into account Lemma 7, it suffices to prove that all the heaps are empty when this event occurs (since it is necessary and sufficient for H to encode a path).

We claim that indeed, all the heaps are empty and moreover $p = r$. Observe first that the global passport of H is obtained by doing $\mathsf{OUTIN}(H^t) + \mathsf{OUTIN}(h)$, which gives for some $(n_u, u \in V)$,

$$\begin{aligned} & \Big[(n_u + (\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) - 1)\mathbf{1}_{u \in t^o} + 1 - 1_{u=r}, u \in V), (n_u + \mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) + \mathbf{1}_{u \in \partial t}(1 - 1_{u=f}), u \in V)\Big] \\ = & \Big[(n_u + \mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) + 1_{u \in \partial t} - 1_{u=r}, u \in V), (n_u + \mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) + \mathbf{1}_{u \in \partial t} - 1_{u=f}, u \in V)\Big]. \end{aligned}$$

Up to a change of variables, this matches the passport of a path (see (15)); however (15) does not characterize collection of heaps corresponding to paths, but collection of heaps of paths concatenated with heap of cycles (see Defi. 8 and (13)). We then need to discard the possibility of cycles. Notice that each vertex different from r has as many outgoing as incoming edges, so that necessarily $p = r$. We will prove that if the heap H_v of an internal node v is empty, then it is also the case of the heaps of its children. Indeed, if H_v is empty, then, all the outgoing edges out of v have been used, and therefore all the incoming edges to v must have also been used too! (one always needs to enter before exiting). Therefore when H_v is empty all the incoming edges arriving at v have been used in their respective heaps. As a matter of fact, the leftmost edge of the heap H_u for each child u of v, is the edge (u, v), and this edge is incoming at v: as a conclusion when H_v is empty, H_u is empty too. □

By Lemma 11,

Corollary 12. *For any* $(t, r) \in \mathit{SpanningTrees}^\bullet(G)$,

$$\begin{aligned} \mathbb{P}(\mathsf{FirstEntranceTree}(w_0, \cdots, w_{\tau_{|V|}(w)}) = (t, r)) &= \mathsf{Weight}(H^t) \sum_{f \in \partial t} \rho_f \mathsf{Weight}(\Xi(t, r, f)) \\ &= \Big(\prod_{e \in E(t,r)} \overleftarrow{M_e}\Big) \sum_{f \in \partial t} \rho_f \mathsf{Weight}(\Xi(t, r, f)). \end{aligned}$$

2.3 The combinatorial trick

In the preceding sections we use collections of heaps to present some operations that could have been presented without them, since withdrawing the steps of the first entrance trees could have been done directly on w. From here, collections of heaps becomes the central objects: just consider a moment the collections of heaps $\mathsf{Heap}^{\mathsf{Trunc}}(w)$ which comes from w, and have been defined using t. From now on, the economic way to think about them is to forget where they come from ! They are just elements $(H_u, u \in V)$ of some sets of collections of heaps whose passports satisfy some constraints (see Proposition 16). When seen as coming from w and $\overleftarrow{w}$, it is tempting to equip the edges of $(H_u, u \in V)$ with a global order inherited from w, but this way of thinking drove us in some dead ends.

Collections of heaps are simple object, devoid of global ordering, even if each heap comes with a fixed order. In the sequel, in order to make clear that we don't try to follow in any way the order induced by w, we still call "first edge" of H_u the leftmost one (the bottom on the figures in the representation of heaps, for example, Figure 1). All our algorithms will use "first edges first": hence since the edges of t have been removed, following paths constructed using the first edges of the heaps produce paths that do not correspond to portions of w or of $\overleftarrow{w}$ in general.
The general principle is that one can do many things with collections of heaps. Including playing golf? Yes.

2.4 Combinatorial golf sequences decomposition

In the sport called *golf*, players hit balls into a series of holes. We introduce **the combinatorial golf sequence**, a new combinatorial object: it consists in a sequence of paths on G, modeling the moves of some balls (placed on some vertices) on a golf court (the graph G), until each of them finds a different *empty hole* (some special vertices). It is parameterized by the pair $\Big[\mathsf{HoleSet}, S\Big]$ as follows:
- $S = (S_1, \cdots, S_{\mathsf{Nb}})$ is the *sequence of starting vertices* from which balls are to be played, with S_j, being the starting position of the j-th. We denote by $\mathsf{Nb}_u := |\{j : S_j = u\}|$ the number of balls starting at u, $\mathsf{Nb} = \sum_u \mathsf{Nb}_u$, the total number of balls; $(\mathsf{Nb}_u, u \in V)$ is called the *counting sequence* of S.
- $\mathsf{HoleSet}$ is a subset of V, the set of empty holes, and we require $u \in \mathsf{HoleSet} \Rightarrow \mathsf{Nb}_u = 0$ so that a node which is the ending point of a path, is not the starting of any paths.

Definition 13. *A sequence of paths $(w^{(1)}, \cdots, w^{(|\mathsf{Nb}|)})$ is an element of $\mathsf{GolfSequence}\Big[\mathsf{HoleSet}, S\Big]$ iff:*
– for any j, $w^{(j)} = \Big(w_i^{(j)}, 0 \le i \le \big|w^{(j)}\big|\Big)$ is a finite path starting at S_j, i.e. $w^{(j)} \in \mathsf{Paths}(S_j)$,
– the final positions of each path is a hole, meaning that

$$\mathsf{Final}(i) := w^{(i)}(|w^{(i)}|) \in \mathsf{HoleSet}, \text{ for all } 1 \le i \le \mathsf{Nb}.$$

– each hole captures the first ball visiting it, and looses after that the property of being a hole. It becomes a standard vertex for subsequent paths. This means that the vertices $\mathsf{Final}(i)$ are distinct, and:
- *the time $|w^{(1)}|$ is the hitting time of $\mathsf{HoleSet}$ by $w^{(1)}$,*
- *... the time $|w^{(k)}|$ is the hitting time of $\mathsf{HoleSet} \setminus \{\mathsf{Final}(1), \cdots, \mathsf{Final}(k-1)\}$ by $w^{(k)}$, for each k.*

Remark 14. $\mathsf{GolfSequence}\Big[\mathsf{HoleSet}, S\Big]$ *is empty if* $\mathsf{Nb} > |\mathsf{HoleSet}|$, *and non-empty if* $\mathsf{Nb} \le |\mathsf{HoleSet}|$.

The stochastic golf sequence. We call $\mathsf{StochasticGolfSequence}\Big[\mathsf{HoleSet}, S\Big]$, a random variable $X := (X^{(1)}, \cdots, X^{(\mathsf{Nb})})$ taking its values in $\mathsf{GolfSequence}\Big[\mathsf{HoleSet}, S\Big]$ distributed as follows. For any $j \in \{1, \cdots, \mathsf{Nb}\}$, conditionally on $\{X^{(1)}, \cdots, X^{(j-1)}\}$, the trajectory of the jth ball $X^{(j)} := (X_t^{(j)}, t \ge 0)$ is a Markov chain with kernel $\overleftarrow{M}$ started at S_j and killed at the first element of $\mathsf{HoleSet}$ not present in the already killed trajectories $(X^{(k)}, k \le j-1)$.

The weight of a combinatorial golf sequence $(w^{(1)}, \cdots, w^{(\mathsf{Nb})})$ is set as

$$\mathsf{Weight}\left[w^{(1)}, \cdots, w^{(\mathsf{Nb})}\right] := \prod_{j=1}^{\mathsf{Nb}} \mathsf{Weight}\left(w^{(j)}\right) = \prod_{j=1}^{\mathsf{Nb}} \prod_{i=0}^{|w^{(j)}|-1} \overleftarrow{M}_{w_i^{(j)}, w_{i+1}^{(j)}}$$

which then corresponds to the probability of independent trajectories in the stochastic golf sequence using the kernel $\overleftarrow{M}$, killed at the first time it visits a still available hole.

Denote by $\mathsf{GolfSequence}\Big[\mathsf{HoleSet}, S, \mathsf{free} \ni f\Big]$ the subset of $\mathsf{GolfSequence}\Big[\mathsf{HoleSet}, S\Big]$ leaving the hole f empty, meaning that $f \notin \{\mathsf{Final}(i), 1 \leq i \leq \mathsf{Nb}\}$).

Proposition 15. *[Trivial stochastic golf sequence property]*
Let $X = (X^{(1)}, \cdots, X^{(\mathsf{Nb})})$ *be a* $\mathsf{StochasticGolfSequence}\Big[\mathsf{HoleSet}, S\Big]$ *with* $\mathsf{Nb} \leq |\mathsf{HoleSet}|$.

(i) We have $\mathbb{P}\Big(X \in \mathsf{GolfSequence}\Big[\mathsf{HoleSet}, S\Big]\Big) = 1$.

(ii) If $\mathsf{Nb} = |\mathsf{HoleSet}| - 1$ *(eventually a single hole will remains free with probability 1) then*

$$\sum_{f \in \mathsf{HoleSet}} \mathbb{P}\Big(X \in \mathsf{GolfSequence}\Big[\mathsf{HoleSet}, S, \mathsf{free} \in f\Big]\Big) = 1.$$

Proof. When $\mathsf{Nb} \leq |\mathsf{HoleSet}|$, the stochastic golf sequence is well defined and takes its value in $\mathsf{GolfSequence}\Big[\mathsf{HoleSet}, S\Big]$: the paths are a.s. finite by irreducibility of the Markov chains. The second statement follows since a single hole must remain free at the end. □

Proposition 16. *For any parameters* $(\mathsf{HoleSet}, S)$ *such that* $u \in \mathsf{HoleSet} \Rightarrow \mathsf{Nb}_u = 0$*, the map*

$$\begin{array}{rccc} \mathsf{Heap}: & \mathsf{GolfSequence}\Big[\mathsf{HoleSet}, S\Big] & \longrightarrow & \mathsf{HeapCol} \\ & \left(w^{(1)}, \cdots, w^{(\mathsf{Nb})}\right) & \longmapsto & \mathsf{Heap}\left(w^{(1)}\right) \oplus \cdots \oplus \mathsf{Heap}\left(w^{(\mathsf{Nb})}\right) \end{array} \tag{20}$$

is a weight preserving injection, and the image is included in

$$\bigcup_{\substack{A:|A|=\mathsf{Nb} \\ A \subset \mathsf{HoleSet}}} \bigcup_{n_u \geq 0} \mathsf{HeapCol}\left[(n_u + \mathsf{Nb}_u, u \in V), (n_u + \mathbf{1}_{u \in A}), u \in V)\right] \tag{21}$$

Remark 17. *This Proposition is also a key point of our approach. The map* $\left(w^{(1)}, \cdots, w^{(N)}\right) \mapsto \mathsf{Heap}\left(w^{(1)}\right) \oplus \cdots \oplus \mathsf{Heap}\left(w^{(N)}\right)$ *is not an injection when defined on a more general set of* Nb*-tuple of paths; but for golf sequences, it is.*

Proof. The fact that the image of any golf sequence is included in (21) comes from the fact that for each u, Nb_u paths start at u. The result is that, eventually, a subset A of the HoleSet with cardinality Nb is eventually occupied. This gives a collection of heaps in the set (21).
Take $H := \mathsf{Heap}\left(w^{(1)}\right) \oplus \cdots \oplus \mathsf{Heap}\left(w^{(\mathsf{Nb})}\right)$ an image under this map. It suffices to explain how to recover $w^{(1)}$ from H (and iterate). Recall that S_1 and HoleSet are known data. To recover $w^{(1)}$ follow steps (a), (b) and (c) in Lemma 11 started at $p = S_1$ with the difference that we stop and return the path the first time $p \in \mathsf{HoleSet}$ and then we define $\mathsf{Final}(1) := p$. Do the same for $w^{(k)}$, except that the stopping time is now the first time the constructed path hits $\mathsf{HoleSet} \setminus \{\mathsf{Final}(1), \cdots, \mathsf{Final}(k-1)\}$. At the end, the set of final points A is the set of final points of the golf paths. □

2.5 Decomposition of Truncated heaps

Proposition 18. *Consider* $(\mathsf{Nb}_u, u \in V)$ *some non negative integers,* HoleSet *a subset of* V*, such that*

$$|\mathsf{HoleSet}| = \mathsf{Nb} + 1, \text{ and such that } u \in \mathsf{HoleSet} \Rightarrow \mathsf{Nb}_u = 0,$$

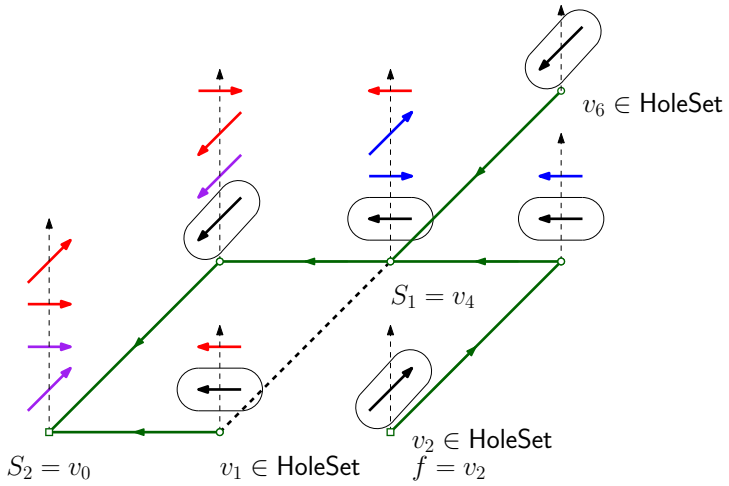

Figure 3: An application of Proposition 18 to our example in Figure 1. The bijection takes as input a pair of heaps: the heap of cycles is in red, and the golf sequence heap is colored blue and purple (the blue steps shows the path starting at $S_1 = v_4$ and the purple steps the path starting at $S_2 = v_0$; in this case $S = (S_1, S_2) = (v_4, v_0)$). The bijection outputs the heap obtained by putting the heap of cycles above the golf sequence heap. The corresponding HoleSet is $\{v_1, v_2, v_6\}$, and $f = v_2$ (it is in free, because it belongs to HoleSet, and there are not incoming edges at v_2). We kept the steps used to construct the tree, which are circled, in order to show the full decomposition of Figure 1.

i.e. after the golf sequence a single vertex remains free. Finally let $S = (S_1, \cdots, S_{\mathsf{Nb}})$ *be a sequence of starting points with counting vector* $(\mathsf{Nb}_u, u \in V)$. *Assume that* $f \in \mathsf{HoleSet}$. *There is a weight preserving bijection between* $\mathcal{N} := \mathsf{HeapGolfSequence}\Big[\mathsf{HoleSet}, S, \mathsf{free} \ni f\Big] \times \mathsf{HCycles}(\mathsf{free} \ni f)$ *and*

$$\mathcal{M} := \bigcup_{\substack{n_u \geq 0 \\ n_f = 0}} \mathsf{HeapCol}\left[(n_u + \mathsf{Nb}_u, u \in V), (n_u + \mathbf{1}_{u \in \mathsf{HoleSet} \setminus \{f\}}), u \in V)\right].$$

See Figure 3 for an application of this proposition.

Proof. The map $\phi : (H_1, H_2) \mapsto H_1 \oplus H_2$ is an injection from $\mathcal{N}$ to $\mathcal{M}$ since passports are compatible, and as following the idea in the proof of Proposition 16 (in the case where $|A| = \mathsf{Nb} - 1$), it is possible to recover the golf sequence from the concatenation.
Now, ϕ is also a surjection: take an element $H \in \mathcal{M}$ and let us build a pair $(h, h') \in \mathcal{N}$ such that $h \oplus h' = H$. For that, we need to show that we can extract from H all the golf trajectories. Consider S_1, and follow the path coming out of S_1 using the edges given by the collection H, as usual. Eventually, this path will reach an element z of HoleSet: indeed, for all nodes which are not in HoleSet, the number of outgoing edges is $\geq$ the number of incoming edges. It is then not possible that the ball arriving at an internal node $u \in t^o$, find an empty heap H_u and is trapped there.

This gives the first golf sequence, and the removal of all the edges used to define it provides an element of $\bigcup_{\substack{n'_u \geq 0 \\ n'_f = 0}} \mathsf{HeapCol}\left[(n'_u + \mathsf{Nb}_u - 1_{u=S_1}, u \in V), (n'_u + \mathbf{1}_{u \in \mathsf{HoleSet} \setminus \{f,z\}}), u \in V)\right]$ since the balance at each vertex of this first path is 0, except at the starting and ending position. We are in the same situation as before with one less path in the golf sequence. When all the paths of the golf sequence have been extracted, the possibly remaining heaps are balanced, and still $n_f = 0$, so that the remaining steps form a heap of piece of $\mathsf{HCycles}(\mathsf{free} \ni f)$. □

2.6 Conclusion: proof of Theorem 3

From Proposition 18 and Lemma 11, one identifies that

$$\mathsf{Weight}(\Xi(t,r,f)) = \mathsf{Weight}\left(\mathsf{HeapGolfSequence}\Big[\mathsf{HoleSet}, S, \mathsf{free} \ni f\Big]\right) \mathsf{Weight}(\mathsf{HCycles}(\mathsf{free} \ni f))$$

where $\mathsf{HoleSet} = \partial t$ and S is any starting sequence with counting sequence $(\mathsf{Nb}_u, u \in V) = ((\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) - 1)\mathbf{1}_{u \in t^o}, u \in V)$: in words $\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u) - 1$ balls start at the internal node u. Such a starting sequence $(S^i, 1 \le i \le \mathsf{Nb})$ is easy to build: take any order on t^o, and sort the nodes $u_1, \cdots, u_{|t^o|}$, and take as sequence S: the node u_1 repeated $\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u_1) - 1$ times, followed by u_2 repeated $\mathsf{Deg}^{\mathsf{in}}{}_{(t,r)}(u_2) - 1$ times, etc.

Recall Corollary (12). We want to prove that

$$\sum_{f \in \partial t} \rho_f \mathsf{Weight}(\Xi(t,r,f)) = \frac{1}{\sum_x \det(\mathrm{Id} - M^{(x)})}.$$

Taking into account that $\rho_f = \frac{\det(\mathrm{Id} - M^{(f)})}{\sum_x \det(\mathrm{Id} - M^{(x)})}$ and (14) we just need to prove that

$$\sum_{f \in \partial t} \mathsf{Weight}(\mathsf{HeapGolfSequence}\Big[\mathsf{HoleSet}, S, \mathsf{free} \ni f]) = 1$$

but this is precisely what says the trivial stochastic golf sequence property (Proposition 15(ii)).

Remark 19. *In Section 2.3 we said that we would use the collection of heaps* $\mathsf{Heap}^{\mathsf{Trunc}}(w)$ *while forgetting the initial order of edges coming from* w *or* $\overleftarrow{w}$*: we used these collections of heap to define the golf sequences, and the additional heap of cycles, and none of them have been defined using the order of the original steps of* w *or* $\overleftarrow{w}$*.*

However, using portions of w *(or* $\overleftarrow{w}$*) to define golf trajectories seem possible but brings many complications: because somehow, "the next ball" to be played depends on the arrival position of the previous. In terms of heaps of cycles everything become messy; for example the global heap of cycles produced finally in our construction, needs to be partially piled after each golf path, if one wants to use the chunks of consecutive steps of* $\overleftarrow{w}$ *in their initial order. These considerations make this ensemble quite difficult to follow and would produce a much longer and involved argument.*

Remark 20. *In the proof, we have rearranged the steps of the random walk with kernel* $\overleftarrow{M}$*. What we did – notably around the notion of golf sequences – is reminiscent of Diaconis & Fulton paper [4] and the commutative property they proved (this property being also at the core of the study of internal DLA). When one studies a sequence of stopped random paths, where the stopping time of each path depends on the ending points of the preceding paths (the starting points being fixed, and possibly different), under some various conditions, the distribution of the set of final points of the trajectories is independent from the order of the sequence of paths and the distribution of the vector giving the total number of traversals of each edge by the sequence of paths, is also independent from the order of sequence of path.*

3 Proof of Theorem 3 by adaptation of the classical argument

The proof follows mainly, Aldous–Broder's ideas, except that $M = \overleftarrow{M}$ is not an hypothesis; we proceed here and there to some small changes with respect to classical arguments.

The LastExitTree**.** Any path $(z_0, \cdots, z_n)$ on G can be used to define a rooted tree

$$(\Gamma_n, z_n) := \mathsf{LastExitTree}(z_0, \cdots, z_n),$$

rooted at z_n, as follows: set (Γ_0, z_0) as the tree reduced to its root z_0; from $k = 0$ to $n-1$, construct (Γ_{k+1}, z_{k+1}) from (Γ_k, z_k) by the addition of the edge (z_k, z_{k+1}) and the suppression of the outgoing edge from z_{k+1} (if any). The set of nodes of (Γ_n, z_n) is $\{z_0, \cdots, z_n\}$. For any $k \in \{1, \cdots, |\{z_0, \cdots, z_n\}|\}$, denote by

$$\nu_k = \max\{j : |\{z_j, \cdots, z_n\}| = k\},$$

the last time k nodes are to be visited "in the future", which is also a last visit time for a node. It follows that the tree $\mathsf{LastExitTree}(z_0, \cdots, z_n)$ has root z_n and has for edges

$$(z_{\nu_k}, z_{\nu_k+1}), \text{ for } k = |\{z_0, \cdots, z_n\}| \text{ to } 2.$$

In Definition 1, we defined the FirstEntranceTree associated with a covering path; this definition can be extended to any path, covering or not, killed at the covering time or not. The following lemma is proved by a straightforward checking:

Lemma 21. *For any path $(w_0, \ldots, w_n)$ on G:*

$$\mathsf{FirstEntranceTree}(w_0, \cdots, w_n) = \mathsf{LastExitTree}(w_n, \cdots, w_0). \tag{22}$$

Several remarks follow:

• The sequence of trees $(\mathsf{FirstEntranceTree}(w_0, \cdots, w_k), 0 \le k \le n)$ is non decreasing, and the jumps of this sequence correspond to the $\tau_m = \inf\{j : |\{w_0, \cdots, w_j\}| = m\}$ where a new node is attached to the tree. Moreover, for a Markov chain $(W_i, i \ge 0)$,

$$\mathsf{FirstEntranceTree}(W_i, i \ge 0) = \mathsf{FirstEntranceTree}(W_i, 0 \le i \le \tau_{|V|}).$$

• The right hand side in (22) is somehow not natural since we reverse the time (time is decreasing). The following Lemma is a direct consequence of the previous one

Lemma 22. *Assume that $(X_k, k \ge 0)$ and $(Y_k, k \le 0)$ are two processes such that,*

$$(X_0, X_1, \cdots) \overset{(d)}{=} (Y_0, Y_{-1}, \cdots). \tag{23}$$

For

$$\begin{aligned} \tau_m &= \inf\{j : |\{X_0, X_1, \cdots, X_j\}| = m\}, \\ L_m &= \inf\{j : |\{Y_0, Y_{-1}, \cdots, Y_{-j}\}| = m\}, \end{aligned}$$

we have

$$(X_0, \cdots, X_{\tau_m}) \overset{(d)}{=} (Y_0, \cdots, Y_{-L_m}) \tag{24}$$

and then

$$\mathsf{FirstEntranceTree}(X_0, \cdots, X_{\tau_m}) = \mathsf{LastExitTree}(X_{\tau_m}, \cdots, X_0) \overset{(d)}{=} \mathsf{LastExitTree}(Y_{-L_m}, \cdots, Y_0). \tag{25}$$

The equality in law (23) is crucial: assume that $(X_k, k \geq 0)$ is a Markov chain with kernel M; in this case the initial value X_0, play a role for $(X_k, k \geq 0)$, which is really different from the final value Y_0 in $(Y_{-k}, Y_{-k+1}, Y_{-k+2}, \cdots, Y_0)$. Requiring (23) for Markov chains can be done in two natural ways:

Lemma 23. *(a) If $(X_k, k \in \mathbb{Z})$ and $(Y_k, k \in \mathbb{Z})$ are two Markov chains with respective positive Markov kernels M and $\overleftarrow{M}$ on G, with invariant distribution ρ. Assume that both $(X_k, k \in \mathbb{Z})$ and $(Y_k, k \in \mathbb{Z})$ are taken under their invariant distributions, then*

$$(X_0, X_1, X_2, \cdots) \stackrel{(d)}{=} (Y_0, Y_{-1}, Y_{-2}, \cdots)$$

(b) If M is reversible (that is $M = \overleftarrow{M}$) then, if $(X_k, k \in \mathbb{Z})$ and $(Y_k, k \in \mathbb{Z})$ are two Markov chains with respective Markov kernel M, then for any x_0,

$$\mathcal{L}\left[(X_0, X_1, X_2, \cdots) \mid X_0 = x_0\right] = \mathcal{L}\left[(Y_0, Y_{-1}, Y_{-2}, \cdots) \mid Y_0 = x_0\right]$$

(so that, somehow, the situation of (a) occurs, conditionally, "outside the stationary regime").

Aldous–Broder argument relies on (b) while (a) seems to us more natural and leads to Theorem 3. The following Lemma is not stated by Aldous [1] or Broder [2], but it is a consequence of their proofs.

Lemma 24. *Assume $(Y_{k,k\in\mathbb{Z}})$ is a Markov chain with kernel $\overleftarrow{M}$ under its stationary regime, then for every $(t, r) \in \mathsf{SpanningTrees}^\bullet(G)$ one has*

$$\mathbb{P}(\mathsf{LastExitTree}(Y_k, k \leq 0) = (t, r)) = \mathsf{Const.} \prod_{e \in E(t,r)} \overleftarrow{M}_e. \tag{26}$$

Before proving the Lemma, let us discuss a bit the consequences of the two last lemmas :
■ As discussed in Aldous [1] and Broder [2], when $\overleftarrow{M} = M$ (reversible case) and r fixed,

$$\mathcal{L}((X_0, \cdots, X_{\tau_m}) \mid X_0 = r) = \mathcal{L}((Y_0, \cdots, Y_{-L_m}) \mid Y_0 = r)$$

so that by Lemma 22,

$$\begin{aligned} \mathbb{P}(\mathsf{FirstEntranceTree}(X_i, 0 \leq i \leq \tau_{|V|}) = (t, r) \mid X_0 = r) &= \mathbb{P}(\mathsf{LastExitTree}(Y_{\leq 0}) = (t, r) | Y_0 = r) \\ &= \mathsf{Const.} 1_{x_0 = r} \Big(\prod_{e \in E(t,r)} M_e \Big) / \rho(r). \end{aligned}$$

■ If we do not assume $\overleftarrow{M} = M$, then for X a Markov chain with kernel M, starting under its invariant distribution, we have

$$\begin{aligned} \mathbb{P}(\mathsf{FirstEntranceTree}(X_i, 0 \leq i \leq \tau_{|V|}) = (t, r)) &= \mathbb{P}(\mathsf{LastExitTree}(Y_{\leq 0}) = (t, r)) \\ &= \mathsf{Const.} \prod_{e \in E(t,r)} \overleftarrow{M}_e, \end{aligned}$$

and conditional on $X_0 = r$ in the left hand side and $Y_0 = r$ in the right hand side, it gives Theorem 3.

Proof. Equation (26) comes from the fact that one can view the sequence $(\mathsf{LastExitTree}(Y_k, k \leq m), m \geq 0)$ as a Markov chain with state space $\mathsf{SpanningTrees}^\bullet(G)$ running from $-\infty$[2] and therefore under its stationary regime. The kernel $Q((t, r), (t', r'))$ of this chain provides the probability to

[2]When the chain is not aperiodic, it has to be seen as the limit when $m \to +\infty$ of a Markov chain starting at $-m$ under its invariant distribution

pass from (t,r) to (t',r'); this weight is $\overleftarrow{M}_{r,r'}$ or 0 depending on whether the root displacement is enough to transform t into t' or not (together with the removal of the outgoing edge from r' in t). The trees (t,r) which may lead to (t',r') do not have the (oriented) edge (r,r') but an edge from r' to one of the neighbors u of r' in G; this is the only difference between t and t'. The trees (t,r) having (t',r') as neighbour (see Fig . 4) are then parameterized by these neighbours u of r', and the balance equation (if one assumes $P(t,r) = \prod_{e\in E(t,r)} \overleftarrow{M}_e$),

$$\sum_{\substack{(t,r):(t',r')\\ \text{is a neighbour}}} P((t,r))Q((t,r),(t',r')) = \left(\sum_{u \text{ neighbour of } r'} \overleftarrow{M}_{r',u} \frac{\left[\prod_{e\in E(t',r')} \overleftarrow{M}_e\right]}{\overleftarrow{M}_{r,r'}} \right) \overleftarrow{M}_{r,r'} = \prod_{e\in E(t',r')} \overleftarrow{M}_e$$

is satisfied.

As mentioned by Aldous [1], the irreducibility follows the fact that the LastExitTree of the sequence of steps that forms the depth first traversal of a tree is the tree itself. □

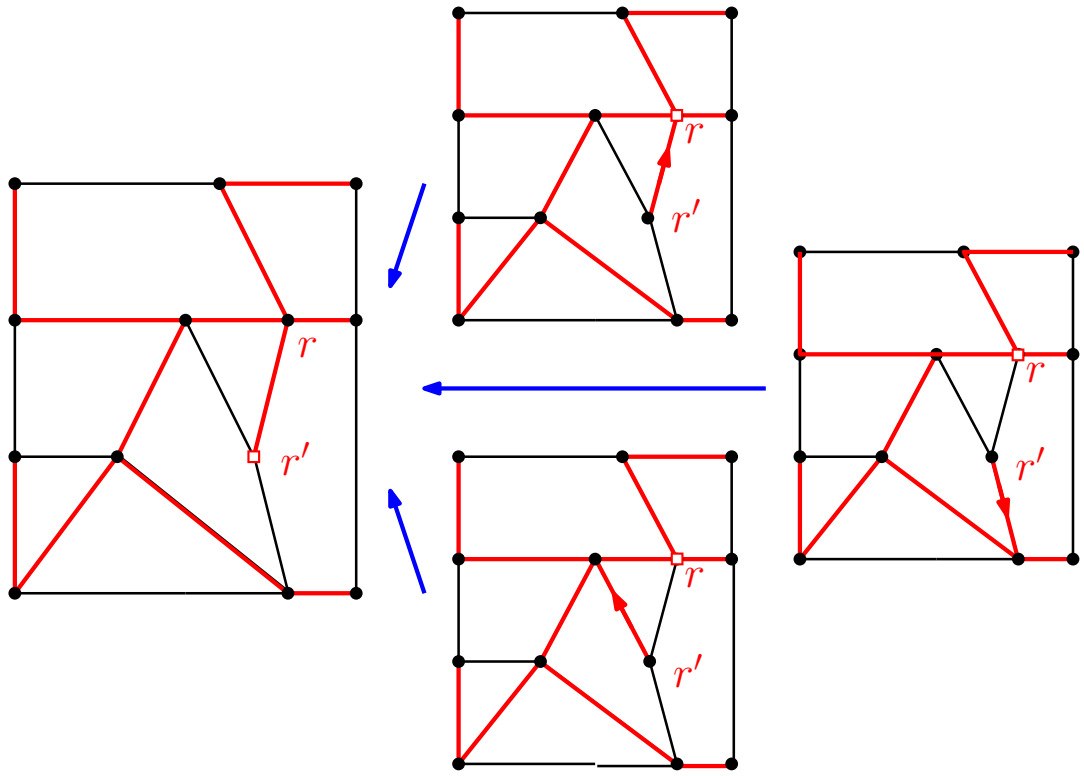

Figure 4: The antecedent of a given tree for the tree Markov chain.

Remark 25. *In Hu & al. [5], it is established that the first entrance tree and the last exit tree taken at the cover time of the Markov chain, have same distribution.*

Acknowledgments

We thank the anonymous referee for her/his corrections and comments.

References

[1] D. J. Aldous. The random walk construction of uniform spanning trees and uniform labelled trees. SIAM Journal on Discrete Mathematics, 3(4):450–465, 1990.

[2] A. Z. Broder. Generating random spanning trees. In FOCS, vol. 89, pages 442–447, 1989.

[3] P. Cartier and D. Foata. Problèmes combinatoires de commutation et réarrangements. Lecture notes in mathematics, 1969.

[4] P. Diaconis and W. Fulton. A growth model, a game, an algebra, Lagrange inversion, and characteristic classes. volume 49, pages 95–119 (1993). 1991. Commutative algebra and algebraic geometry, II (Italian) (Turin, 1990).

[5] Y. Hu, R. Lyons, and P. Tang. A reverse Aldous–Broder algorithm. Annales de l'Institut Henri Poincaré, Probabilités et Statistiques, 57(2):890 – 900, 2021.

[6] A. A. Járai. The uniform spanning tree and related models, 2009. `http://www.maths.bath.ac.uk/%7Eaj276/teaching/USF/USFnotes.pdf`.

[7] C. Krattenthaler. The theory of heaps and the cartier–foata monoid. In Appendix of the electronic edition of Problèmes combinatoires de commutation et réarrangements. 2006.

[8] J. G. Propp and D. B. Wilson. How to get a perfectly random sample from a generic markov chain and generate a random spanning tree of a directed graph. J Algorithms, 27(2):170–217, 1998.

[9] G. X. Viennot. Heaps of pieces, i : Basic definitions and combinatorial lemmas. In G. Labelle and P. Leroux, editors, Combinatoire énumérative, pages 321–350, Berlin, Heidelberg, 1986. Springer Berlin Heidelberg.

[10] D. B. Wilson. Generating random spanning trees more quickly than the cover time. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 296–303, 1996.

[11] D. Zeilberger. A combinatorial approach to matrix algebra. Discrete Mathematics, 56(1):61 – 72, 1985.