A Class of Graphs where Ranking Spanning Trees and Forests Takes Linear Time

Ömer Eğecioğlu
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA

Jeffrey B. Remmel
Department of Mathematics
University of California, San Diego, La Jolla, CA, 92093-0112

S. Gill Williamson
Department of Computer Science and Engineering
University of California, San Diego, La Jolla, CA, 92093-0114

Abstract

In [11], Remmel and Williamson defined a class of directed graphs, called filtered digraphs, and described a natural class of bijections between oriented spanning forests of these digraphs and associated classes of functions. Filtered digraphs include many specialized graphs such as complete k-partite graphs. Their bijection allowed them to give explicit formulas for various multivariate generating functions for the oriented spanning forests which arise in this context. Specializations of their generating functions include many classical formulas for the number of spanning forests that arise from the matrix tree theorem as well as new formulas for the number of spanning forests of certain directed graphs which do not follow from the matrix tree theorem.

In this paper, we prove another important property of their bijection, namely, that it allows one to develop linear time algorithms for ranking and unranking spanning trees or spanning forests of filtered digraphs. In particular, this allows one to generate random spanning trees or spanning forests of a filtered digraph G in linear time in the number of vertices, n, of G where as the best known algorithm for generating random spanning forests of an arbitrary graph is $n^{2.376}$. Moreover, our algorithm allows one to randomly generate spanning forests of the G which contain a certain class of pre-specified edges.

1 Introduction

The original motivation for this paper was the work of Eğecioğlu and Remmel [3] who gave a bijective proof of the formula n^{n-2} for the number of Cayley trees on n vertices, i.e. the number of spanning trees of the complete graph K_n. In particular, they showed that there is a natural bijection between the set of $C_{n,1}$ of all Cayley trees on n vertices where all edges are directed toward the root 1 and the class of functions $\mathcal{F} = \{f : \{2, \ldots, n-1\} \to \{1, \ldots, n\}\}$. Later in [4], Eğecioğlu and Remmel extended this idea to give a bijective proof for the number of spanning trees of the complete k-partite graph, $K_{n_1,\ldots,n_k}$. Again in [4], Eğecioğlu and Remmel showed that there was a natural bijection between a certain class of functions $f : \{2, \ldots, n-1\} \to \{1, \ldots, n\}$ and the set of spanning trees of $K_{n_1,\ldots,n_k}$ rooted at vertex 1.

It is well known that the formulas for the number of spanning trees of K_n and $K_{n_1,\ldots,n_k}$ follow from the matrix tree theorem [1]. One advantage of [3, 4] over the matrix tree theorem approach is that the resulting bijections give rise to natural multivariate generating functions which keep track of the descent and rise edges for the set of root-directed spanning trees, i.e., the spanning trees where all edges are directed toward the root. A second advantage of the bijective approach of [3, 4] is that there are well known techniques [9, 10, 12] for ranking and unranking function classes and hence the bijections provide ways to rank and unrank spanning trees of K_n and $K_{n_1,\dots,n_k}$.

Later, Remmel and Williamson [11] extended the work of Eğecioğlu and Remmel to a much larger class of graphs. In particular, Remmel and Williamson defined a class of directed graphs, called filtered digraphs, and described a natural class of bijections between oriented spanning forests of these digraphs and associated classes of functions. Remmel and Williamson derived multivariate generating functions for the oriented spanning forests which arise in this context and they linked basic properties of these spanning forests to properties of the functions to which they correspond. The class of filtered digraphs contains not only both K_n and $K_{n_1,\dots,n_k}$ but also many directed graphs to which the matrix tree theorem does not apply. In addition, Remmel and Williamson extended the results of Eğecioğlu and Remmel in two other ways. First, the methods of [11], applied to spanning forests rather than just to spanning trees. Second, the multivariate generating functions derived in [11] were finer than those considered by Eğecioğlu and Remmel and hence have a greater variety of specializations.

In this paper, we shall show the Remmel-Williamson bijections can be used to give linear time ranking and unranking algorithms for the set of spanning forests in filtered digraphs. The basic idea is that the bijection between the set of spanning forests in filtered digraphs and their associated function classes can be carried out in linear time. This reduces the problem of ranking and unranking spanning forests to the problem of ranking and unranking functions for which there are well known techniques [12]. In particular, we obtain linear time ranking and unranking algorithms to generate random spanning trees or spanning forests of a filtered digraph G. This is a considerable improvement over the best algorithms for generating random spanning forests of an arbitrary graph, which runs in time $O(n^{2.376})$. Moreover, since our bijection preserves the collections of ascent edges between the directed graph of the function and the rooted directed spanning tree of its image under our bijection, we can also randomly generate spanning forests or trees which contain a give collection of pre-specified ascent edges with essentially no extra cost.

This paper is organized as follows. In Section 2, we define the class of filtered digraphs and their corresponding function classes. We then define the bijection between the function class of a filtered digraph and the set of root-directed spanning trees of the filtered digraph. In Section 3, we shall prove that this bijection can be carried out in linear time so that the problem of ranking and unranking spanning forests or trees of a filtered digraph in linear time can be reduced to the problem of ranking and unranking functions in the corresponding function class. In Section 4, we shall give linear time algorithms for ranking and unranking functions in function classes associated with filtered digraphs. Finally in Section 5, we shall give three specific applications of our algorithms.

Example 5.1. We consider one of the simplest cases of a filtered digraph, namely, the case where $G = K_n$. In this case, we show explicitly how our basic unranking and ranking procedures can rank and unrank root-directed spanning forests of K_n which contain a collection of prespecified ascent edges or for which we specify a collection of vertices which must start an ascent edge.

Example 5.2. We consider the case of filtered digraphs for complete multipartite graphs $K_{n_1,\dots,n_k}$. In this case, we explicitly give the unranking procedure for root directed spanning forests of $K_{n_1,\dots,n_k}$.

Example 5.3. We consider multipartite cyclic digraphs $C_{n_1,\dots,n_k}$. Multipartite cyclic graphs are not covered by the classical matrix tree theorem. Again we shall explicitly give the unranking procedure for root directed spanning forests of $C_{n_1,\dots,n_k}$.

2 The Bijection between Filtered Digraphs and Function Classes

In this section, we shall give the definitions of a *filtered digraph* and their corresponding functions classes as described in [11]. We shall also define the Remmel-Williamson bijection and give some of its properties.

Let $[n] = \{1, 2, \dots, n\}$. Let $G = ([n], E)$ be a digraph with vertex set $[n]$ and edge set E. Let $\mathbf{F} = (c_1, c_2, \dots, c_k)$ be a composition of n. That is, assume c_i is a positive integer for each i and $\sum_{i=1}^{k} c_i = n$. Let $N_0 = 0$ and let $N_t = c_1 + \cdots + c_t$ for $t = 1, \dots, k$. We let $\mathcal{C}_t = \{1 + N_{t-1}, \dots, N_t\}$ for $t = 1, \dots, n$. Note that each $\mathcal{C}_t$ is an interval and the collection of nonempty sets $\{\mathcal{C}_i \mid i = 1, \dots, k\}$ forms a set partition of $[n]$. We call this set partition the *filtration* associated with the composition $\mathbf{F}$.

Definition 2.1 *Given a composition* $\mathbf{F} = (c_1, \ldots, c_k)$ *of* n*, we define a partial order relation* $\leq_{\mathbf{F}}$ *on* $[n]$ *by* $x \leq_{\mathbf{F}} y$ *if* $x = y$ *or if* $x \in \mathcal{C}_i$ *and* $y \in \mathcal{C}_j$ *where* $1 \leq i < j \leq k$*. We call* $\leq_{\mathbf{F}}$ *the filtration order on* $[n]$*.*

We write $x <_F y$ if $x \leq_F y$ but $x \neq y$. If $x <_F y$, then our definitions ensure that $x < y$ as integers. Note that $1 \in C_1$, $n \in C_k$, and each of the sets $\mathcal{C}_i$ is a set of incomparable elements (i.e., an antichain) with respect to $\leq_F$. In the standard terminology for posets, $\leq_{\mathbf{F}}$ is the ordinal sum of the antichains $\mathcal{C}_i$, $1 \leq i \leq k$.

Definition 2.2 *Let* $\{\mathcal{C}_i : i = 1, \ldots, k\}$ *be the filtration associated with the composition* $\mathbf{F} = (c_1, \ldots, c_k)$ *of* n*. Let* I_B *and* I_S *be subsets of* $[k]$ *and let* $\mathcal{B} = \{C_i : i \in I_B\}$ *and* $\mathcal{S} = \{C_i : i \in I_S\}$*. We refer to the sets* $\mathcal{B}$ *and* $\mathcal{S}$ *as the bases and summits of* G *respectively. A set* $\mathcal{C}_i \in \mathcal{B}$ *is called a base of* G *and its elements are called base vertices of* G*. A set* $\mathcal{C}_i \in \mathcal{S}$ *is called a summit of* G *and its elements are called summit vertices of* G*. We say that a digraph* $G = ([n], E)$ *is a filtered digraph with respect to* $\mathbf{F}$*,* I_B*, and* I_S*, if the following conditions hold for all* $x, y \in [n]$*.*

1. $1 \in I_B$*,* $1 \notin I_S$*,* $k \notin I_B$*, and* $k \in I_S$*.*
2. *If* $x, y \in \mathcal{C}_i$ *for some* $1 \leq i \leq k$*, then* $(x, y) \notin E$*.*
3. *If* $y <_F x$*, then* $(x, y) \in E$ *if and only if there exist* $p < q$ *such that* $x \in C_q$*,* $q \in I_S$*,* $y \in C_p$ *and* $p \in I_B$*.*
4. *If* $x \in \mathcal{C}_i$*,* $1 \leq i < k$*, and* $\mathcal{C}_i$ *is not a summit, then there is some* y *such that* $(x, y) \in E$ *and* $x <_{\mathbf{F}} y$*.*

It is perhaps helpful to paraphrase conditions (1)-(4). Condition (1) states that $\mathcal{C}_1$ is a base but not a summit and $\mathcal{C}_k$ is a summit but not a base. Otherwise the bases and summits are arbitrary. A set $\mathcal{C}_i$ with $i \notin \{1, k\}$ may be both a base and a summit. Condition (2) states that the restrictions of G to the sets $\mathcal{C}_i$ are empty digraphs (no edges). Condition (3) states that all directed edges between summit vertices and "lower" base vertices are present and these are the only "downward" edges in E. That is, these are the only edges $(x, y) \in E$ with $y <_{\mathbf{F}} x$. Finally, condition (4) states that for any vertex x that does not belong to a summit, there is at least one upward edge out of x. That is, there is at least one edge $(x, y) \in E$ such that $x <_{\mathbf{F}} y$.

Given any digraph $G' = ([n], E')$, we can define the set of "root-directed" spanning forests of G' with roots $r_1, \ldots, r_q \in [n]$ as follows. First we regard the digraph G' as an undirected graph in the obvious manner. Next we consider the set of all spanning forests $T' = (T'_1, \ldots, T'_q)$ of this undirected version of G' with subtrees T'_i, $i = 1, \ldots, q$, such that $r_i \in T'_i$ for $i = 1, \ldots, q$. We can then think of each T'_i as a directed graph by considering r_i as the root of T'_i and directing all edges back toward the root. That is, we direct all edges in T'_i so that there is a directed path from each vertex v in T'_i to r_i. If for each i, all these directed edges are in fact in E, then we say that T' is a *root-directed spanning forest of* G' *with roots* $r_1, \ldots, r_q$. Alternatively, such spanning forests are called "oriented," but we shall stick to the former terminology.

We denote the set of all root-directed spanning forests of G' with roots $r_1, \ldots, r_q$ by $\mathcal{T}^{G'}_{\{r_1, \ldots, r_q\}}$. If $n \notin \{r_1, \ldots, r_q\}$, then we use the notation $\mathcal{T}^{G'}_{\{r_1, \ldots, r_q\}; r_j}$ to designate all root-directed spanning forests of G' with n in the component tree rooted at r_j.

Returning to the case $G = ([n], E)$, suppose we are given a directed edge (i, j) where $1 \leq i, j \leq n$. Following [11], we define the weight of (i, j), $W((i, j))$, by

$$W((i,j)) = \begin{cases} p_i s_j \text{ if } i < j, \\ q_i t_j \text{ if } i \geq j \end{cases} \tag{1}$$

where p_i, q_i, s_i, t_i are variables for $i = 1, \ldots, n$. We shall call a directed edge (i, j) a descent edge if $i \geq j$ and an ascent edge if $i < j$. We then define the weight of any digraph $G = ([n], E)$ by

$$W(G) = \prod_{(i,j) \in E} W((i,j)). \tag{2}$$

Definition 2.3 *Let* $G = ([n], E)$ *be a filtered digraph with respect to the composition* $\mathbf{F} = (c_1, \ldots, c_k)$*, the set of bases indexed by* I_B *and the set of summits indexed by* I_S*. Let* m *be such that* $1 \leq m < n - 1$ *and assume that* $1, \ldots, m$ *are base vertices of* G*. Suppose that* $m \in \mathcal{C}_t$*. Let* $\mathcal{F}_n(G, \mathbf{F}, m)$ *be the set of all functions* $f : \{m + 1, \ldots, n - 1\} \to [n]$ *that satisfy the following conditions.*

1. *If $f(i) \neq i$ then $(i, f(i)) \in E$.*

2. *If $f(i) = i$ and $i \in \mathcal{C}_p$, then $p \in I_S \cap I_B$ and $p \geq t$.*

3. *If $p \in I_S \cap I_B$ and $p \geq t$, then there is at most one $i \in \mathcal{C}_p$ such that $f(i) = i$.*

We call $\mathcal{F}_n(G, \mathbf{F}, m)$ the m-canonical function class for the filtered digraph G with respect to $\mathcal{F}$, I_B and I_S.

We note that our conditions ensure that $\mathcal{F}_n(G, \mathbf{F}, m)$ is not empty. That is, suppose that $v \in [n] - \{n, 1, \ldots m\}$ is in $\mathcal{C}_i$. Now if $\mathcal{C}_i$ is a summit, then we know that $i > 1$. Moreover, we know that $1 \in \mathcal{C}_1$ and that $\mathcal{C}_1$ is a base so that $(v, 1)$ in E. Thus there is at least one choice for $f(v)$. Similarly, if $\mathcal{C}_i$ is not a summit, then $i < k$ and we know that there is at least one upward edge out of v in G. Thus again, there is at least one choice for $f(v)$.

We can think of each $f \in \mathcal{F}_n(G, \mathbf{F}, m)$ as a directed graph on the vertex set $\{1, \ldots, n\}$. That is, if $f(i) = j$, then there is a directed edge from i to j. A moment's thought will convince one that, in general, the digraph corresponding to a function $f \in \mathcal{F}_n(G, \mathbf{F}, m)$ will consists of $m + 1$ root-directed trees rooted at vertices $1, \ldots, m$ and n respectively, with all edges directed toward their roots, plus a number of directed cycles of length ≥ 1. For each vertex v on a given cycle, there is possibly a root-directed tree attached to v with v as the root and all edges directed toward v. Note the fact that there are trees rooted at vertices $n, 1, \ldots, m$ is due to the fact that these elements are not in the domain of f. Thus there can be no directed edges out of any of these vertices. We let the weight of f, $W(f)$, be the weight of the digraph associated with f.

Suppose that we are given a filtered digraph $G = ([n], E)$ with respect to the composition $\mathbf{F} = (c_1, \ldots, c_k)$. Suppose that the summits are indexed by I_S and the bases are indexed by I_B. Suppose also that $1, \ldots, m$ are fixed base elements of G. Let $\{\mathcal{C}_i \mid i = 1, \ldots, k\}$ be the filtration partition for $\mathbf{F}$ and suppose that $m \in \mathcal{C}_t$. Thus $N_{t-1} < m \leq N_t$. Let $\mathcal{T}^G_{[m];j}$ denote all root-directed spanning forests of G with roots in $[m] = \{1, \ldots, m\}$ and for which n is a vertex of the tree (component) rooted at j. We shall show that in this situation, if the root $j \notin \mathcal{C}_t$, then there is a natural bijection Θ_j between the m-canonical function class $\mathcal{F}_n(G, \mathbf{F}, m)$ for G and the set $\mathcal{T}^G_{[m];j}$. If $j \in \mathcal{C}_t$, then there is a corresponding bijection Θ^*_j from the subset $\mathcal{F}^*_n(G, \mathbf{F}, m)$ of $\mathcal{F}_n(G, \mathbf{F}, m)$ to $\mathcal{T}^G_{[m];j}$ where $\mathcal{F}^*_n(G, \mathbf{F}, m)$ consists of all $f \in \mathcal{F}_n(G, \mathbf{F}, m)$ such that $f(i) \neq i$ for all $i \in \mathcal{C}_t$ such that $i > m$. The fact that Θ^*_j, which is simply the restriction of Θ_j to $\mathcal{F}^*_n(G, \mathbf{F}, m)$, is a bijection will easily follow from our proof that Θ_j is a bijection.

To define the bijection Θ_j, $1 \leq j \leq N_{t-1}$, we first imagine that the directed graph corresponding to $f \in \mathcal{F}_n(G, \mathbf{F}, m)$ is drawn in two parts (see Figure 1). The first part of the graph consists of the rooted-trees at roots $1, \ldots, j-1, j+1, \ldots m$. The second part of the graph is drawn so that

(a) the trees rooted at n and j are drawn on the extreme left and the extreme right respectively with their edges directed upwards,

(b) the cycles are drawn so that their vertices form a directed path on the line between n and j, with one back edge above the line, and the root-directed tree attached to any vertex on a cycle is drawn below the line between n and 1 with its edges directed upwards,

(c) each cycle is arranged so that its maximum element is on the right, and

(d) the cycles are arranged so that if the maximum element m_c of a cycle c is in $\mathcal{C}_i$ and the maximum element $m_{c'}$ of a cycle c' in $\mathcal{C}_j$, then c is to the left of c' if either (i) $i > j$, (ii) $i = j$ and c is a one cycle or (iii) $i = j$, neither c nor c' are one cycles and $m_c > m_{c'}$.

Figure 1 pictures a function f drawn according to the rules (a)-(d) where $n = 27$, $\mathbf{F} = (5, 5, 3, 6, 8)$, and $G = ([n], E)$ is the filtered digraph defined as follows. Since $\mathbf{F} = (5, 5, 3, 6, 8)$, $\mathcal{C}_1 = \{1, \ldots, 5\}, \mathcal{C}_2 = \{6, \ldots, 10\}, \mathcal{C}_3 = \{11, \ldots, 13\}, \mathcal{C}_4 = \{14, \ldots, 19\}$ and $\mathcal{C}_5 = \{20, \ldots, 27\}$. We let $I_B = \{1, 2, 4\}$ and $I_S = \{2, 4, 5\}$ so that the sets $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_4$ are bases and the sets $\mathcal{C}_2, \mathcal{C}_4$ and $\mathcal{C}_5$ are summits. Finally, we specify the edges of G as follows.

$\mathcal{C}_1 : \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$
$\mathcal{C}_2 : \mathcal{C}_1, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5$

i	f(i)
4	9
5	12
6	11
7	27
8	16
9	4
10	10
11	20
12	19
13	17
14	14
15	1
16	8
17	5
18	10
19	5
20	6
21	3
22	8
23	8
24	1
25	11
26	2

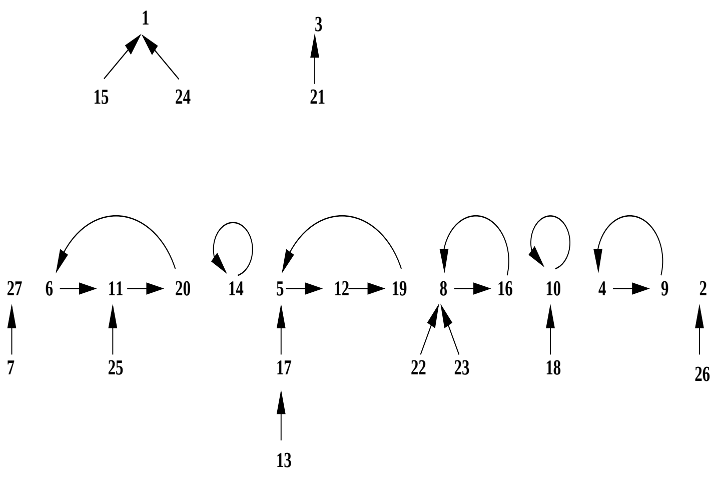


Figure 1: The digraph of a function

$\mathcal{C}_3 : \mathcal{C}_4, \mathcal{C}_5$
$\mathcal{C}_4 : \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_5$
$\mathcal{C}_5 : \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_4$

In the above specification of the edges of G, we interpret $\mathcal{C}_i : \mathcal{C}_{j_1}, \ldots \mathcal{C}_{j_s}$ to mean that there is a directed edge from every vertex $v \in \mathcal{C}_i$ to every vertex w in $\mathcal{C}_{j_k}$ for $k = 1, \ldots, s$.

This given, suppose that the digraph of f is drawn as described above and the cycles of $f$are $c_1(f), \ldots, c_a(f)$ reading from left to right. We let $r_{c_i(f)}$ and $l_{c_i(f)}$ denote the right and left endpoints of the cycle $c_i(f)$ for $i = 1, \ldots, a$. Note that if $c_i(f)$ is a 1-cycle, then we let $r_{c_i(f)} = l_{c_i(f)}$ be the element in the 1-cycle. $\Theta_j(f)$ is obtained from f by simply deleting the back edges $(r_{c_i(f)}, l_{c_i(f)})$ for $i = 1, \ldots, a$ and adding the directed edges $(r_{c_i(f)}, l_{c_{i+1}(f)})$ for $i = 1, \ldots, a-1$ plus the directed edges $(n, l_{c_1(f)})$ and $(r_{c_a(f)}, j)$. That is, we remove the all the back edges that are above the line, and then we connect n to the lefthand endpoint of the first cycle, the righthand endpoint of each cycle to the lefthand endpoint of the cycle following it, and we connect the righthand endpoint of the last cycle to j. For example, $\Theta_2(f)$ is pictured in Figure 2 for the f given in Figure 1. If there are no cycles in f, then $\Theta_j(f)$ is simply the result of adding the directed edge (n, j) to the digraph of f.

To see that $\Theta_j(f) \in \mathcal{T}^G_{[m];j}$ for all $f \in \mathcal{F}_n(G, \mathbf{F}, m)$, first observe that for all $i = 1, \ldots, a$, $l_{c_i(f)}$ must be a base element and $r_{c_i(f)}$ must be a summit element. That is, if c_i is a cycle of length one, then $f(r_{c_i(f)}) = l_{c_i(f)} = r_{c_i(f)}$ so that $l_{c_i(f)}$ must be both a summit and base element, since by our definition of $\mathcal{F}_n(G, \mathbf{F}, m)$, the only fixed points of f are elements of $\mathcal{C}_p$ for some $p \in I_B \cap I_S$. If c_i is not a one cycle, then $f(r_{c_i(f)}) = l_{c_i(f)} < r_{c_i(f)}$ since $r_{c_i(f)}$ is the largest element of c_i. By our definition of $\mathcal{F}_n(G, \mathbf{F}, m)$, $(r_{c_i(f)}, l_{c_i(f)})$ must be a downward edge of G so that $r_{c_i(f)}$ must be a summit element and $l_{c_i(f)}$ must be a base element. In particular, this means that the edges $(n, l_{c_1(f)})$ and $(r_{c_a(f)}, j)$ are elements of G. Recall here that $n \in \mathcal{C}_k$, which is the highest summit, and hence n is connected to all base elements in G. On the other hand, by assumption, $j \in [N_{t-1}] \subseteq [m]$. By definition, $1, \ldots, m$ are base elements and $m \in \mathcal{C}_t$. Thus $\{1, \ldots, m\} \subseteq \cup_{b=1}^t \mathcal{C}_b$ and $\mathcal{C}_1, \ldots, \mathcal{C}_t$ are all bases. Now suppose that $r_{c_a(f)} \in \mathcal{C}_s$. There are two possibilities. First it could be that c_a is a one cycle. Then in this case, our definition of $\mathcal{F}_n(G, \mathbf{F}, m)$ ensures that $s \in I_B \cap I_S$ and $s \geq t$. Thus the edge $(r_{c_a(f)}, j)$ connects a summit vertex to an element in a lower base and hence is in G. If c_a is not a one cycle, $l_{c_a(f)} \in \mathcal{C}_u$ where $u \in I_B$ and $u < s$. However, since $l_{c_a(f)}$ is part of a cycle of f (hence in the domain of f), we know that $l_{c_a(f)} > m$. Thus, by our definition of a filtration, we know that $u \geq t$ since $m \in \mathcal{C}_t$. Since $j \in \mathcal{C}_r$ for some $r < t$ and j is a base element, $\mathcal{C}_r$ is a base below the summit $\mathcal{C}_s$ so

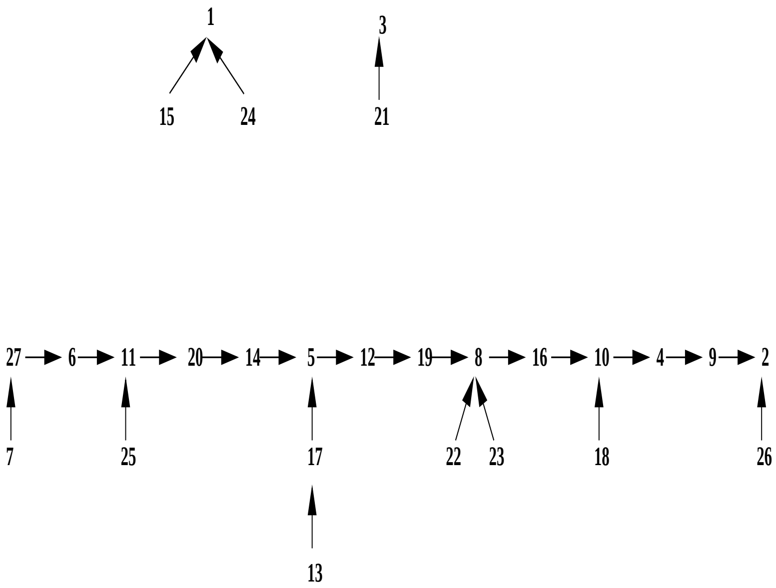


Figure 2: $\Theta_2(f)$

that again we can conclude that $(r_{c_a(f)}, j)$ is in G.

Now consider the other edges $(r_{c_i(f)}, l_{c_{i+1}(f)})$ that we added to the digraph of f. There are two cases to consider.

Case 1 c_{i+1} is a one cycle.
Then $l_{c_{i+1}(f)} = r_{c_{i+1}(f)} \in \mathcal{C}_p$ for some $p \in I_B \cap I_S$. However, by our convention for drawing the digraph of f, c_{i+1} must be the leftmost cycle of the form c_k such that $r_{c_k(f)} \in \mathcal{C}_p$. Hence $r_{c_i(f)}$ must be in some $\mathcal{C}_w$ where $w > p$. Since $r_{c_i(f)}$ is a summit element, the edge $(r_{c_i(f)}, l_{c_{i+1}(f)})$ goes from a summit vertex to a vertex which is in a lower base and hence is in G.

Case 2 c_{i+1} is not a one cycle.
Then $l_{c_{i+1}(f)} < r_{c_{i+1}(f)}$. Thus $r_{c_{i+1}(f)} \in \mathcal{C}_p$ for some $p \in I_S$ and $l_{c_{i+1}(f)} \in \mathcal{C}_s$ for some $s \in I_B$ where $s < p$ since $(r_{c_{i+1}(f)}, l_{c_{i+1}(f)})$ is a downward edge in G. By our convention for ordering the cycles, we know that $r_{c_i(f)} \in \mathcal{C}_u$ where $u \geq p$. Thus $\mathcal{C}_u$ must be a summit which lies above the base $\mathcal{C}_s$ so again the edge $(r_{c_i(f)}, l_{c_{i+1}(f)})$ must be in G.

In the special case where f has no cycles, we add only the edge (n, j) which must be in G since j is in a base which lies below the top summit $\mathcal{C}_k$. It follows that all the new edges that we add to the digraph of f are in G. Note that since we remove all the backedges and these are the only possible loops in the digraph of f, all the remaining edges are of the form (i, j) where $f(i) = j$ and $i \neq j$ and hence are in G by our definition of $\mathcal{F}_n(G, \mathbf{F}, m)$. Thus $\Theta_j(f) \in \mathcal{T}^G_{[m];j}$ for all $f \in \mathcal{F}_n(G, \mathbf{F}, m)$.

The weight preserving property of Θ_j is also easy to verify. That is, by our conventions, any backedges $(r_{c_i(f)}, l_{c_i(f)})$ are descent edges so that its weight is $q_{r_{c_i(f)}} t_{l_{c_i(f)}}$. Thus the total weight of the backedges is

$$\prod_{i=1}^{a} q_{r_{c_i(f)}} t_{l_{c_i(f)}}. \tag{3}$$

Our argument above shows that all the new edges that we add are also descent edges so that the weight of the new edges is

$$q_n t_{l_{c_1(f)}} \left(\prod_{i=1}^{a-1} q_{r_{c_i(f)}} t_{l_{c_{i+1}(f)}}\right) q_{r_{c_a(f)}} t_j = q_n t_j \prod_{i=1}^{a} q_{r_{c_i(f)}} t_{l_{c_i(f)}}. \tag{4}$$

Since all the remaining edges have the same weight in both the digraph of f and in the digraph $\Theta_j(f)$, it follows that $q_n t_j W(f) = W(\Theta_j(f))$ as claimed.

To see that Θ_j is a bijection, we shall describe how to define Θ_j^{-1}. Given a forest $T \in \mathcal{T}_{[m]:j}^G$, consider the path

$$m_0 = n, x_1, \ldots, m_1, x_2, \ldots m_2, \ldots, x_t, \ldots, m_t, j$$

where m_i is the maximum interior vertex on the path from m_{i-1} to j, $1 \leq i \leq t$. If (m_{i-1}, m_i) is an edge on this path, then it is understood that $x_i, \ldots, m_i = m_i$ consists of just one vertex and we define $x_i = m_i$. Note that by definition $m_0 = n > m_1 > \ldots > m_t$.

We obtain the digraph $\Theta_j^{-1}(T)$ from T via the following procedure.

Procedure for computing $\Theta_j^{-1}(T)$:

(1) First we declare that any edge e of T which is not an edge of the path from n to j is an edge of $\Theta_j^{-1}(T)$.

(2) Next we remove all edges of the form (m_t, j) or (m_{i-1}, x_i) for $1 \leq i \leq t$.

Finally for each i with $1 \leq i \leq t$, we consider the subpath $x_i, \ldots, m_i$.

(3) If $m_i = x_i$, create a directed loop (m_i, m_i).

(4) If $m_i \in C_s$ for some s, but $x_i \notin C_s$, convert the subpath $x_i, \ldots, m_i$ into the directed cycle $x_i, \ldots, m_i, x_i$.

(5) If $x_i, m_i \in C_s$ for some s, but $x_i \neq m_i$, then convert the subpath $x_i, x_i', \ldots, m_i$ to the directed cycle $x_i', \ldots, m_i, x_i'$ and the directed loop (x_i, x_i).

We refer the reader to [11] for the details that $\Theta_j^{-1}(T)$ is actually the inverse map of Θ_j. The key observation that ensures that we can reverse the bijection Θ_j^{-1} is that a loop (y, y) with $y \in \mathcal{C}_p$ that appears in $\Theta_j^{-1}(T)$ must occur as the *leftmost* cycle of all cycles whose maximum element is also in $\mathcal{C}_p$. Thus, if we apply the map Θ_j to $\Theta_j^{-1}(T)$ then we will just reconstruct T. Thus Θ_j is a bijection from $\mathcal{F}_n(G, \mathbf{F}, q)$ to $\mathcal{T}_{[m];j}^G$ for each $1 \leq j \leq m$.

Next we consider the case were $N_{t-1} + 1 \leq j \leq m$ so that $j \in \mathcal{C}_t$. Given a function $f^* \in \mathcal{F}_n^*(G, \mathbf{F}, m)$, the forest $\Theta_j^*(f^*)$ is created by exactly the same procedure that we used to create $\Theta_j(f)$ for $f \in \mathcal{F}_n(G, \mathbf{F}, m)$. To show that $\Theta_j^*(f^*)$ is always in $\mathcal{T}_{[m];j}^G$, one can use essentially the same argument that we used to show that $\Theta_j(f)$ is always in $\mathcal{T}_{[m];j}^G$ for an $f \in \mathcal{F}_n(G, \mathbf{F}, m)$ when $j \leq N_{t-1}$ with one exception. That is, consider the edge $(r_{c_a}(f^*), j)$. There are two possibilities. First it could be the c_a is a one cycle so that $l_{c_a}(f^*) = r_{c_a}(f^*)$ and $r_{c_a}(f^*)$ is a fixed point of f^*. Then by definition of $\mathcal{F}_n^*(G, \mathbf{F}, m)$, it must be that $r_{c_a}(f^*) \in \mathcal{C}_p$ for $p > t$ and $r_{c_a}(f^*)$ is a summit vertex. Since j is a base vertex in $\mathcal{C}_t$ and $p > t$, it follows that $(r_{c_a}(f^*), j)$ is an edge in G. The only other possibility is that $l_{c_a}(f^*) \neq r_{c_a}(f^*)$ in which case $(r_{c_a}(f^*), l_{c_a}(f^*))$ is a downward edge. But then once again we can conclude that $r_{c_a}(f^*)$ is a summit vertex in some $\mathcal{C}_p$ for $p > t$. Thus once again in this case, $(r_{c_a}(f^*), j)$ is an edge in G since j is a base vertex in $\mathcal{C}_t$ and $p > t$.

Finally given a $T^* \in \mathcal{T}_{[m];j}^G$, we define $(\Theta_j^*)^{-1}(T^*)$ in exactly the same way that we defined $\Theta_j^{-1}(T)$ for $T \in \mathcal{T}_{[m];j'}^G$ when $j' \leq N_{t-1}$. The only thing that we have to check to prove that $(\Theta_j^*)^{-1}(T^*) \in \mathcal{F}_n^*(G, \mathbf{F}, m)$ is that $(\Theta_j^*)^{-1}(T^*)$ never has a fixed point in $\mathcal{C}_t$. That is, exactly the same argument will show that $\Theta_j^*((\Theta_j^*)^{-1}(T^*)) = T^*$. It then follows from our ordering of the cycles that the only way that $f^* = (\Theta_j^*)^{-1}(T^*)$ could have a fixed point in $\mathcal{C}_t$ is if the rightmost cycle of $(\Theta_j^*)^{-1}(T^*)$ has its maximal element $r_{c_a}(F^*)$ in $\mathcal{C}_p$. But this is impossible since $j \in \mathcal{C}_t$ and $(r_{c_a}(f^*), j)$ is an edge in G.

In fact, Remmel and Williamson [11] proved the following result.

Theorem 2.4 *Let $G = ([n], E)$ be a filtered digraph with respect to the composition $\mathbf{F} = (c_1, \ldots, c_k)$, the set of summits indexed by I_S, and the set of bases indexed by I_B. Assume that $1, \ldots, m$ are base elements* *of G, that $m \in \mathcal{C}_t$, and that $N_{t-1} = c_1 + \cdots + c_{t-1}$. Then, for each $1 \leq j \leq N_{t-1}$, there is a bijection $\Theta_j : \mathcal{F}_n(G, \mathbf{F}, m) \to \mathcal{T}^G_{[m];j}$ and, for each $N_{t-1}+1 \leq j \leq m$, there is a bijection $\Theta^*_j : \mathcal{F}^*_n(G, \mathbf{F}, m) \to \mathcal{T}^G_{[m];j}$ such that*

$$\begin{aligned} q_n t_j W(f) &= W(\Theta_j(f)),\ 1 \leq j \leq N_{t-1} \ \text{ and} &(5)\\ q_n t_j W(f) &= W(\Theta^*_j(f)),\ N_{t-1}+1 \leq j \leq m. &(6)\end{aligned}$$

Hence

$$q_n(t_1 + \cdots + t_{N_{t-1}}) \sum_{f \in \mathcal{F}_n} W(f) \ + \ q_n(t_{N_{t-1}+1} + \cdots + t_m) \sum_{f \in \mathcal{F}^*_n} W(f) = \sum_{T \in \mathcal{T}^G_{[m]}} W(T). \qquad (7)$$

Here if $t = 1$, then $N_{t-1} = N_0 = 0$ and we take $t_1 + \cdots + t_{N_{t-1}} = 0$.

We end this section with some general remarks about how to modify the proof of Theorem 2.4 to deal with root-directed spanning forests at other roots. For example, suppose that we want to find a formula for the number of root-directed spanning forests which are rooted at the m largest vertices instead of the m smallest vertices. One can easily modify the machinery that we developed to prove Theorem 2.4 to handle this case. That is, we need only make the following observation. Suppose that we replace each vertex label i with the vertex label $n+1-i$. It is easy to see that we interchange ascent edges with descent edges and conversely. That is, suppose that $i \neq j$. Then an ascent edge (i,j) with weight $p_i s_j$ will become a descent edge $(n+1-i, n+1-j)$ with weight $q_{n+1-i} t_{n+1-i}$. Similarly a descent edge (i,j) with weight $q_i t_j$ becomes ascent edge $(n+1-i, n+1-j)$ with weight $p_{n+1-i} s_{n+1-i}$. In the case when $i = j$, we declared a loop (i,i) as a descent edge with weight $q_i t_i$ which means that we want $(n+1-i, n+1-i)$ to be an ascent edge with weight $p_{n+1-i} s_{n+1-i}$. Thus the new weights of edges should be as follows. We define the weight of (i,j), $\overline{W}((i,j))$, by

$$\overline{W}((i,j)) = \begin{cases} p_i s_j \text{ if } i \leq j, \\ q_i t_j \text{ if } i > j \end{cases}$$

Similarly, we define the weight of any digraph $G = ([n], E)$ by

$$\overline{W}(G) = \prod_{(i,j) \in E} \overline{W}((i,j)).$$

If we do this, we can then get a formula for the sum of the weights $\overline{W}(F)$ of the root-directed forests F rooted at $n+1-1, \ldots, n+1-m$ by simply taking equation (4) and systematically replacing q_i by p_{n+1-i}, t_i by s_{n+1-i}, p_i by q_{n+1-i} and s_i by t_{n+1-i}.

One can also develop a machinery to deal with root-directed forests rooted at an arbitrary set of roots. This was done in the case of a single root for the set of spanning trees of K_n by Eğecioğlu and Remmel [3]. In that case, if one picks a root i other than 1 or n, one must use a coarser set of weights than we used in Theorem 2.4. A similar situation holds for the case of root-directed forests of filtered digraphs that are not rooted at the m largest or the m smallest vertices. However we shall not pursue these types of results in this paper.

3 Construction of the spanning forests from the the function table in time $O(n)$

In this section, we shall briefly outline the proof that one can compute the bijections Θ_j and Θ^*_j in linear time. That is, we shall prove the following.

Theorem 3.1 *Let $G = ([n], E)$ be a filtered digraph with respect to the composition $\mathbf{F} = (c_1, \ldots, c_k)$, the set of summits indexed by I_S, and the set of bases indexed by I_B. Assume that $1, \ldots, m$ are base elements of G, that $m \in \mathcal{C}_t$, and that $N_{t-1} = c_1 + \cdots + c_{t-1}$. Then, for each $1 \leq j \leq N_{t-1}$, we can compute the bijection $\Theta_j : \mathcal{F}_n(G, \mathbf{F}, m) \to \mathcal{T}^G_{[m];j}$ and its inverse in linear time and, for each $N_{t-1}+1 \leq j \leq m$, we can compute the bijection $\Theta^*_j : \mathcal{F}^*_n(G, \mathbf{F}, m) \to \mathcal{T}^G_{[m];j}$ and its inverse in linear time.*

Proof. We need only prove this result for the bijection Θ_j since the bijection Θ_j^* is just the restriction of Θ_j to $\mathcal{F}_n^*(G, \mathbf{F}, m)$.

Suppose that we are given $f \in \mathcal{F}_n(G, \mathbf{F}, m)$. Our basic data structure for the function f is a list of pairs $\langle i, f(i), j_i, j_{f(i)} \rangle$ where $i \in \mathcal{C}_{j_i}$ and $f(i) \in \mathcal{C}_{j_{f(i)}}$. Our goal is to construct the directed graph of $\Theta_j(f)$ from our data structure for f, that is, for $i = m+1, \dots, n$, we want to find the set of four-tuples, $\langle i, t_i, j_i, j_{t_i} \rangle$, such that there is directed edge from i to t_i in $\Theta_j(f)$ and $i \in \mathcal{C}_{j_i}$ and $t_i \in \mathcal{C}_{j_{t_i}}$.

We shall not try to give the most efficient algorithm to construct $\Theta_j(f)$ from f. Instead, we shall give an outline the basic proceedure which shows that one can construct $\Theta_j(f)$ from f in linear time. For ease of presentation, we shall organize our procedure so that it makes four linear time passes through the basic data structure for f to produce the data structure for $\Theta_j(f)$.

Pass 1. *Goal: Find, in linear time in n, a set of representatives $t_1, \dots, t_r$ of the cycles of the directed graph of the function f.*

To help us find $t_1, \dots, t_r$, we shall maintain an array

$$A[m+1], A[m+2], \cdots A[n-1],$$

where for each i, $A[i] = (c_i, p_i, q_i)$ is a triple of integers such $c_i \in \{0, \dots, n-m\}$ and $\{p_i, q_i\} \subseteq \{-1, m+1, \cdots, n-1\}$. The c_i's will help us keep track of what loop we are in relative to the sequence of operations described below. Then our idea is to maintain, through the p_i and q_i, a doubly linked list of the locations i in A where $c_i = 0$, and we obtain pointers to the first and last elements of this doubly linked list. It is a standard exercise that these data structures can be maintained in linear time.

Initially, all the c_i's will be zero. In general, if $c_i = 0$, then p_i will be the largest such integer $m+1 \leq j < i$ for which $c_j = 0$ if there is such a j and $p_i = -1$ otherwise. Similarly, $q_i > i$ to be the smallest such integer $n_1 \geq k > i$ for which $c_k = 0$ if there is such a k and $q_i = -1$ if there is no such k. If $c_{m+1} > 0$, then q_{m+1} is the smallest integer $j > m+1$ such that $c_j = 0$ and $q_{m+1} = -1$ if there is no such integer j. If $c_{n-1} > 0$, then p_{n-1} is the largest integer $k < n_1$ such that $c_k = 0$ and $p_{n-1} = -1$ if there is no such integer k.

We initialize A by setting $A[m+1] = (0, -1, q_{m+1})$, by setting $A[i] = (0, i-1, i+1)$ for $m+1 < i < n-1$, and by setting $A[n-1] = (0, p_{n-1}, -1)$. If $m+1 < n-1$ then $q_{m+1} = m+2$ and $p_{n-1} = n-2$. Otherwise ($m+1 = n-1$), these quantities are both -1.

LOOP(1): Start with $i_1 = m+1$, setting $c_{m+1} = 1$.

Compute $f^0(m+1), f^1(m+1), f^2(m+1), \dots, f^{k_1}(m+1)$, each time updating A by setting $c_{f^j(m+1)} = 1$ and adjusting pointers, until, prior to setting $c_{f^{k_1}(m+1)} = 1$, we discover that either

(1) $f^{k_1}(m+1) \in \{1, \dots, m, n\}$. In this case, we have reached a node in *graph*(f) which is not in the domain of f and we start over again with the $m+1$ replaced by the smallest i for which $c_i = 0$ or

(2) $x = f^{k_1}(m+1)$ already satisfies $c_x = 1$. This condition indicates that the value x has already occurred in the sequence $m+1, f(m+1), f^2(m+1), \dots, f^{k_1}(m+1)$. Then we set $t_1 = f^{k_1}(m+1)$.

LOOP(2): Start with $i_2 = q_{m+1}$ which is the location of the first i such that $c_i = 0$, and repeat the calculation of LOOP1 with i_2 instead of $i_1 = m+1$. In this manner, generate $f^0(i_2), f^1(i_2), f^2(i_2), \dots, f^{k_2}(i_2)$, each time updating A by setting $c_{f^j(i_2)} = 2$ and adjusting pointers, until either

(1) $f^{k_1}(m+1) \in \{1, \dots, m, n\}$. In this case, we have reach a node in *graph*(f) which is not in the domain of f and we start over again with the i_2 replaced by the smallest i for which $c_i = 0$ or

(2) $x = f^{k_1}(m+1)$ already satisfies $c_x = 2$. (This condition indicates that the value x has already occurred in the sequence $i_2, f(i_2), f^2(i_2), \dots, f^{k_1}(i_2)$.) Then we set $t_2 = f^{k_1}(i_2)$.

We continue this process until $q_{m+1} = -1$. At this point, we will have generated $t_1, \ldots, t_r$, where the last loop was LOOP(r). The array A will be such that, for all $m+1 \le i \le n-1$, $1 \le c_i \le r$ identifies the LOOP in which that particular domain value i occurred in our computation described above.

Pass 2. *Goal: For $i = 1, \ldots, r$, find the largest element l_i in the cycle determined by t_i.*
It is easy to see that this computation can be done in linear time by one pass through the array A computed in Pass 1 above. At the end of Pass 2, we $s_i = f(l_i)$. Thus when we draw the cycle containing t_i according to our definition of $\Theta_j(f)$, l_i will be right most element in the and s_i will be the left most element of the cycle containing t_i. However, at this point, we have not ordered the cycles appropriately. This ordering will be done in the next pass.

Pass 3. *Goal: Sort $(l_1, s_1), \ldots, (l_k, s_k)$ so that they are appropriately ordered according the criterion for the bijection $\Theta_j(f)$ as described in by condition (a) -(d).*

Note that according to condition (d) of our criterion for drawing the graph of f so that we can construct $\Theta_j(f)$, the cycles are arranged so that if the maximum element m_c of a cycle c is in $\mathcal{C}_i$ and the maximum element $m_{c'}$ of a cycle c' in $\mathcal{C}_j$, then c is to the left of c' if either (i) $i > j$, (ii) $i = j$ and c is a one cycle or (iii) $i = j$, neither c nor c' are one cycles and $m_c > m_{c'}$. It is then easy to see that our desired ordering can be constructed in linear time by constructing a triple $\langle j_{l_i}, \chi(l_i \text{ is a fixed point}), l_i \rangle$ where $l_i \in \mathcal{C}_{j_{l_i}}$ and then doing a lexicographic bucket sort. Here for any statement A, $\chi(A) = 1$ if A is true and $\chi(A) = 0$ if A is false. (See Williamson's book [12] for details on the lexicographic bucket sort.)

Pass 4. *Goal: Construct the directed graph of $\Theta_j(f)$ from the directed graph of f.*

Given the root j whose subtree is to contain n, modify the table for f to produce the table for $\theta(f)$. That is, assume that $(l_1, s_1), \ldots, (l_k, s_k)$ is the sorted list coming out of Pass 3. Then we modify the table for f so that we add entries for the directed edges $\langle n, s_1 \rangle$ and $\langle l_k, j \rangle$ and modify entries of the four tuples starting with $l_1, \ldots, l_k$ so that there corresponding second elements are $s_2, \ldots, s_k, j$ respectively. This can be done in linear time using our data structures.

Next, consider the problem of computing the inverse of either Θ_j or Θ_j^*. Suppose that we are given the data structure of the forest $F \in T^G_{[m]}$, i.e. we are given a set of four tuples, $\langle i, t_i, j_i, j_{t_i} \rangle$, such that there is a directed edge from i to t_i in F and $i \in \mathcal{C}_{j_i}$ and $t_i \in \mathcal{C}_{j_{t_i}}$. Recall the computation of the inverse consists of two basic steps.

Step 1. Given a forest $F \in \mathcal{T}^G_{[m]:j}$, consider the path

$$m_0 = n, x_1, \ldots, m_1, x_2, \ldots m_2, \ldots, x_t, \ldots, m_t, j$$

where m_i is the maximum interior vertex on the path from m_{i-1} to j, $1 \le i \le t$. If (m_{i-1}, m_i) is an edge on this path, then it is understood that $x_i, \ldots, m_i = m_i$ consists of just one vertex and we define $x_i = m_i$. Note that by definition $m_0 = n > m_1 > \ldots > m_t$.

First it is easy to see that by making one pass through the data structure for F, we can construct the directed path $n \to a_1 \to \ldots \to a_r$ where $a_r \in \{1, \ldots, m\}$. Thus $j = a_r$ in this case. In fact, we can construct a doubly linked list $(n, a_1, \ldots, a_{r-1}, j)$ with pointers to the first and last elements in linear time. If we traverse the list in reverse order, $(j, a_{r-1}, \ldots, a_1, n)$, then it easy to see that $m_t = a_{r-1}$, m_{t_1} is the next element in the list $(a_{r-2}, \ldots, a_1)$ which is greater than m_t and, in general, having found $m_i = a_s$, then m_{i-1} is the first element in the list $(a_{s-1}, \ldots, a_1)$ which is greater than m_i. Thus it is not difficult to see that we can use our doubly linked list to produce the factorization

$$m_0 = n, x_1, \ldots, m_1, x_2, \ldots m_2, \ldots, x_t, \ldots, m_t, j$$

in linear time.

Step 2. We obtain the digraph $\Theta_j^{-1}(F)$ from F via the following procedure.

Procedure for computing $\Theta_j^{-1}(F)$:

(1) First we declare that any edge e of F which is not an edge of the path from n to j is an edge of $\Theta_j^{-1}(F)$.

(2) Next we remove all edges of the form (m_t, j) or (m_{i-1}, x_i) for $1 \leq i \leq t$.

Finally for each i with $1 \leq i \leq t$, we consider the subpath $x_i, \ldots, m_i$.

(3) If $m_i = x_i$, create a directed loop (m_i, m_i).

(4) If $m_i \in C_s$ for some s, but $x_i \notin C_s$, convert the subpath $x_i, \ldots, m_i$ into the directed cycle $x_i, \ldots, m_i, x_i$.

(5) If $x_i, m_i \in C_s$ for some s, but $x_i \neq m_i$, then convert the subpath $x_i, x_i', \ldots, m_i$ to the directed cycle $x_i', \ldots, m_i, x_i'$ and the directed loop (x_i, x_i).

Again it is easy to see that we can use the data structure for F, our doubly linked list, and our path factorization, $m_0 = n, x_1, \ldots, m_1, x_2, \ldots m_2, \ldots, x_t, \ldots, m_t, j$ to construct the data structure for f in linear time.

Since we use exactly the same procedure to construct the inverse of Θ_j^*, it follows that we can construct the inverses of Θ_j and Θ_j^* in linear time. $\Box$.

Given that we can carry out the bijections Θ_j and Θ_j^* and their inverses in linear time, it follows that we can reduce the problem of constructing linear time ranking and unranking algorithms for $\mathcal{T}^G_{[m];j}$ to the problem of constructing linear time ranking and unranking algorithms for the corresponding function classes. We will construct linear time ranking and unranking algorithms for these function classes in the next section.

4 Linear Time Ranking and Unranking Algorithms for Function Classes

The main goal of this section is to construct linear time algorithms for ranking and unranking the function classes $\mathcal{F}_n(G, \mathbf{F}, m)$ or $\mathcal{F}_n^*(G, \mathbf{F}, m)$. Our idea is to reduce this problem to the problem of ranking and unranking paths through a planar tree, see [12]. Thus before we present our ranking and unranking algorithms for $\mathcal{F}_n(G, \mathbf{F}, m)$ or $\mathcal{F}_n^*(G, \mathbf{F}, m)$, we need to develop some notation and terminology for ranking and unranking paths through rooted planar trees.

Given a rooted planar tree T, let $L(T)$ be the numbers of leaves of T and $Path(T)$ be the set of paths which go from the root to a leaf. Then for any path $p \in Path(T)$, we define the rank of p relative to T, $rank_T(p)$, to be the number of leaves of T that lie to the left of p.

Given two rooted planar trees T_1 and T_2, $T_1 \otimes T_2$ is the tree that results from T_1 by replacing each leaf of T_1 by a copy of T_2, see Figure 3. If the vertices of T_1 and T_2 are labeled, then we shall label the vertices of $T_1 \otimes T_2$ according to the convention that each vertex v in T_1 have the same label in $T_1 \otimes T_2$ that it has in T_1 and each vertex w in a copy of T_2 that is decendent from a leaf labeled l in T_1 has a label (l, s) where s is the label of w in T_2. Given rooted planar trees $T_1, \ldots, T_k$ where $k \geq 3$, we can define $T_1 \otimes T_2 \otimes \cdots \otimes T_k$ by induction as $(T_1 \otimes \cdots \otimes T_{k-1}) \otimes T_k$. Similarly if $T_1 \ldots, T_k$ are labeled rooted planar trees, we can define the labeling of $T_1 \otimes T_2 \otimes \cdots \otimes T_k$ by the same inductive process.

Now suppose that we are given two rooted planar trees T_1 and T_2 and suppose that $p_1 \in Path(T_1)$ and $p_2 \in Path(T_2)$. Then we define the path $p_1 \otimes p_2$ in $T_1 \otimes T_2$ which follows p_1 to its leaf l in T_1 and then follows p_2 in the copy of T_2 that sits on top of leaf l to a leaf l' in $T_1 \otimes T_2$. Similarly, given paths $p_i \in T_i$ for $i = 1, \ldots k$, we can define a path $p = p_1 \otimes \cdots \otimes p_k \in Path(T_1 \otimes T_2 \otimes \cdots \otimes T_k)$ by induction as $(p_1 \otimes \cdots \otimes p_{k-1}) \otimes p_k$.

Next we give two simple lemmas that tell us how to rank and unrank the set of paths in such trees.

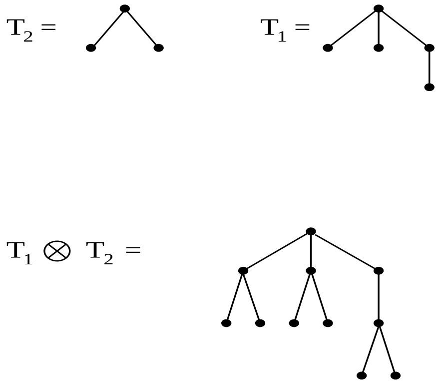


Figure 3: The operation $T_1 \otimes T_2$.

Lemma 4.1 *Suppose that $T_1, \dots, T_k$ are rooted planar trees and*

$$T = T_1 \otimes T_2 \otimes \cdots \otimes T_k.$$

Then for any path $p = p_1 \otimes \cdots \otimes p_k \in Path(T)$,

$$rank_T(p) = \sum_{j=1}^{k} rank_{T_j}(p_j) \prod_{l=j+1}^{k} L(T_l) \tag{8}$$

Proof. We proceed by induction on k. Let us assume that $T_1, \dots T_k$ are labeled rooted planar trees.

First suppose that $k = 2$ and that p_1 is a path that goes from the root of T_1 to a leaf labeled 1_1 and p_2 goes from the root of T_2 to a leaf labeled l_2. Thus $p_1 \times p_2$ goes from the root of $T_1 \otimes T_2$ to the leaf l_1 in T_1 and then proceeds to the leaf (l_1, l_2) in $T_1 \otimes T_2$. Now for each leaf l' to the left of l_1 in T_2, there are $L(T_2)$ leaves of $T_1 \otimes T_2$ that lie to left of (l_1, l_2) coming from the leaves of the copy of T_2 that sits on top of l'. Thus there are a total of $L(T_2) \cdots rank_{T_1}(p_1)$ such leaves. The only other leaves of $T_1 \otimes T_2$ that lie to left of $p_1 \otimes p_2$ are the leaves of the form (l_1, l'') where l'' is to left of p_2 in T_2. There are $rank_{T_2}(p_2)$ such leaves. Thus there are a total of $rank_{T_2}(p_2) + L(T_2) \cdots rank_{T_1}(p_1)$ leaves to left of $p_1 \otimes p_2$ and hence

$$rank_{T_1 \otimes T_2}(p_1 \otimes p_2) = rank_{T_2}(p_2) + L(T_2) \cdot rank_{T_1}(p_1)$$

as desired.

Next assume that (8) holds for $k < n$ and that $n \geq 3$. Then

$$\begin{aligned}
& rank_{T_1 \otimes \cdots \otimes T_n}(p_1 \otimes \cdots \otimes p_n) = \\
& rank_{(T_1 \otimes \cdots \otimes T_{n-1}) \otimes T_n}(p_1 \otimes \cdots \otimes p_{n-1}) \otimes p_n) = \\
& rank_{T_n}(p_n) + L(T_n)(\sum_{j=1}^{n-1} rank_{T_j}(p_j) \prod_{l=j+1}^{n-1} L(T_l)) = \\
& \sum_{j=1}^{n} rank_{T_j}(p_j) \prod_{l=j+1}^{n} L(T_l).
\end{aligned}$$

$\square$.

This given, it is easy to develop an algorithm for unranking in a product of trees. The proof of this lemma can be found in [12].

Lemma 4.2 *Suppose that $T_1,\ldots,T_k$ are rooted planar trees and $T = T_1 \otimes T_2 \otimes \cdots \otimes T_k$. Then given a $p \in Path(T)$ such that $rank_T(p) = r_0$, $p = p_1 \otimes \cdots \otimes p_k \in Path(T)$ where $rank_{T_i}(p_i) = q_i$ and*

$$\begin{aligned}
r_0 &= q_1 \prod_{l=2}^{k} L(T_l) + r_1 \ \textit{where}\ 0 \le r_1 < \textstyle\prod_{l=2}^{k} L(T_l) && (9)\\
r_1 &= q_2 \prod_{l=3}^{k} L(T_l) + r_2 \ \textit{where}\ 0 \le r_2 < \textstyle\prod_{l=3}^{k} L(T_l) && (10)\\
&\vdots && (11)\\
r_{k-2} &= q_{k-1} L(T_k) + r_{k-1} \ \textit{where}\ 0 \le r_{k-1} < L(T_k) && (12)\\
r_{k-1} &= q_k && (13)
\end{aligned}$$

Next we shall construct the appropriate rooted planar trees for the function classes associated with filtered digraphs. Let $G = ([n], E)$ be a filtered digraph with respect to the composition $\mathbf{F} = (c_1,\ldots,c_k)$, the set of summits indexed by I_S, and the set of bases indexed by I_B. Assume that $1,\ldots,m$ are base elements of G, that $m \in \mathcal{C}_t$, and that $N_{t-1} = c_1 + \cdots + c_{t-1}$.

Essentially we have two basic cases. That is, if we are trying to rank and unrank relative to the bijection $\Theta_j : \mathcal{F}_n(G,\mathbf{F},m) \to \mathcal{T}^G_{[m]}$, when $j \notin \mathcal{C}_t$, we will define a rooted planar tree $\mathcal{T}_j$ such that the paths in $P(\mathcal{T}_j)$ encode the functions in $\mathcal{F}_n(G,\mathbf{F},m)$. Similarly, in the case where $j \le m$ and $j \in \mathcal{C}_t$, then when we want to rank and unrank according the bijection $\Theta^*_j : \mathcal{F}^*_n(G,\mathbf{F},m) \to \mathcal{T}^G_{[m]}$, we will define a rooted planar tree $\mathcal{T}^*_j$ such that the paths in $P(\mathcal{T}_j)$ encode the functions in $\mathcal{F}^*_n(G,\mathbf{F},m)$.

The final tree $\mathcal{T}$ for ranking and unranking spanning forests in $\mathcal{T}^G_{[m]}$ is then constructed as follows. First we start with a tree S_m of height 1 with m leaves labeled from left to right with $1,\ldots,m$. Then for each j we will attach the tree $\mathcal{T}_j$ below node j if j is not in $\mathcal{C}_t$ and we will attach the tree $\mathcal{T}^*_j$ below node j if $j \in \mathcal{C}_t$. Our idea is that if our path goes through the node labeled j at level 1 in $\mathcal{T}$ and $j \notin \mathcal{C}_t$, then the remainder of the path through $\mathcal{T}_j$ encodes a function f in $\mathcal{F}_n(G,\mathbf{F},m)$ so that, under the bijection $\Theta_j(f)$, n will be in the tree of the forest with root j. Similarly, if our path goes through the node labeled j at level 1 in $\mathcal{T}$ and $j \in \mathcal{C}_t$, then the remainder of the path through $\mathcal{T}^*_j$ encodes a function f in $\mathcal{F}^*_n(G,\mathbf{F},m)$ so that, under the bijection $\Theta^*_j(f)$, n will be in the tree of the forest with root j. For example, if $1,\ldots u \in \bigcup_{i=1}^{t-1} \mathcal{C}_i$ and $u+1,\ldots,m \in \mathcal{C}_t$, then $\mathcal{T}$ will be as pictured in Figure 4. We denote this tree as

$$\mathcal{T} = \mathcal{T}_1 \oplus \cdots \oplus \mathcal{T}_u \oplus \mathcal{T}^*_{u+1} \oplus \cdots \oplus \mathcal{T}^*_m.$$

First we need to establish some notation. For $i = t,\ldots,k$, let $\mathcal{D}_i = \mathcal{C}_i - \{1,\ldots,m,n\}$. Let $\mathcal{D}_i = \{d_{1,i} < \ldots < d_{t_i,i}\}$. For each i such that $m+1 \le i \le n-1$, let $A_i = \{j : (i,j) \in E\}$ be the set of vertices that are adjacent to i in G. Let T_{A_i} be the rooted planar tree of height 1 with $|A_i|$ leaves. Let q_s denote the path of rank s in T_{A_i}. We shall then think of q_s as encoding that $f(i) = w_s$ where $A_i = \{w_{0,i} < \ldots w_{|A_i|-1,i}\}$. That is, if we follow the path of rank s in T_{A_i}, then f maps i to the s-th element of its adjacency set.

Now suppose that $j \in [m]$ and $j \notin \mathcal{C}_t$. Then $\mathcal{T}_j$ will be of the form $T^j_{\mathcal{D}_t} \otimes \cdots \otimes T^j_{\mathcal{D}_k}$ where for $i = t,\ldots,k$, $T^j_{\mathcal{D}_i}$ is a tree which encodes the values of $f \in \mathcal{F}_n(G,\mathbf{F},m)$ restricted to the set $\mathcal{D}_i$. We then have two cases for the definition of $T^j_{\mathcal{D}_i}$.

Case 1. $i \notin I_B \cap I_S$.
In this case,

$$T^j_{\mathcal{D}_i} = T_{A_{d_{1,i}}} \otimes T_{A_{d_{2,i}}} \otimes \cdots \otimes T_{A_{d_{t_i,i}}}.$$

Note that in this case each path p in $Path(T^j_{\mathcal{D}_i})$ is of the form $p = q_{s_1} \otimes \cdots \otimes q_{s_{t_i}}$ where $q_{s_r} \in Path(T_{A_{d_{r,i}}})$ for some $s_1,\ldots,s_{t_i}$. Thus we can think of p as specifying the values of a function $f \in \mathcal{F}_n(G,\mathbf{F},m)$ restricted to $\mathcal{D}_i$ by letting $f(d_{r,i}) = w_{s_r,d_{r,i}}$.

Case 2. $i \in I_B \cap I_S$.

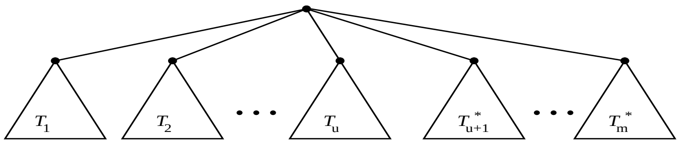


Figure 4: The tree $\mathcal{T}$.

In this case, each $f \in \mathcal{F}_n(G, \mathbf{F}, m)$ has at most one fixed point in $\mathcal{D}_i$. Our idea is to encode the possibility of a fixed point as the branching at level 0 of our tree $T^j_{\mathcal{D}_i}$. That is, the root of $T^j_{\mathcal{D}_i}$ will be of degree $1+t_i$ where t_i is the number of elements of $\mathcal{D}_i$. If we follow the left most branch out of level 0 to a node labeled (0), then f will have no fixed point in $\mathcal{D}_i$. If we follow the t-th branch from the left where $t \geq 2$ to a node labeled $(d_{t-1,i})$, then we will regard this as encoding that $d_{t-1,i}$ is the unique fixed point of f in $\mathcal{D}_i$. We can then use the same idea as in Case 1 to encode the rest of our choices for f on $\mathcal{D}_i$ given our branching at the first level. That is, if we take the left most branch out of level 0 so that there is no fixed point of f in $\mathcal{D}_i$, then we need to attach a copy of $T_{A_{d_{1,i}}} \otimes T_{A_{d_{2,i}}} \otimes \cdots \otimes T_{A_{d_{t_i,i}}}$ descendent from that vertex to code the function values of f on $\mathcal{D}_i$ just as in Case 1. If we take any other branch out of level 0, then we are specifying that some $d_{r,i}$ is the unique fixed point of f on $\mathcal{D}_i$. In that case, the tree below that node should encode the function values of f on $\mathcal{D}_i - \{d_{r,i}\}$. Thus we need only attach a copy of

$$T_{A_{d_{1,i}}} \otimes \cdots T_{A_{d_{r-1,i}}} \otimes T_{A_{d_{r+1,i}}} \cdots \otimes T_{A_{d_{t_i,i}}}$$

descendent from that vertex to code the function values of f on $\mathcal{D}_j - \{d_{r,j}\}$ using the same reasoning that we applied in Case 1.

Now suppose that $j \in [m]$ and $j \in \mathcal{C}_t$. Then $\mathcal{T}^*_j$ will be of the form $T^{j,*}_{\mathcal{D}_t} \otimes \cdots \otimes T^{j,*}_{\mathcal{D}_k}$ where for $i =, t, \ldots, k$, $T^{j,*}_{\mathcal{D}_i}$ is a tree which encodes the values of $f \in \mathcal{F}^*_n(G, \mathbf{F}, m)$ restricted to the set $\mathcal{D}_i$. Note that $\mathcal{F}_n(G, \mathbf{F}, m)$ differs from $\mathcal{F}^*_n(G, \mathbf{F}, m)$ only if $t \in I_S \cap I_B$. In that case, there can be no fixed points in f restricted to $\mathcal{D}_t$. Thus we must define $T^{j,*}_{\mathcal{D}_t}$ according to Case 1 above, that is,

$$T^{j,*}_{\mathcal{D}_t} = T_{A_{d_{1,t}}} \otimes T_{A_{d_{2,t}}} \otimes \cdots \otimes T_{A_{d_{t_i,t}}}.$$

If $i \notin C_t$, then we define $T^{j,*}_{\mathcal{D}_i}$ via Cases 1 and 2 in the exact same way that we defined $T^j_{\mathcal{D}_i}$ in the case when $j \notin \mathcal{C}_t$.

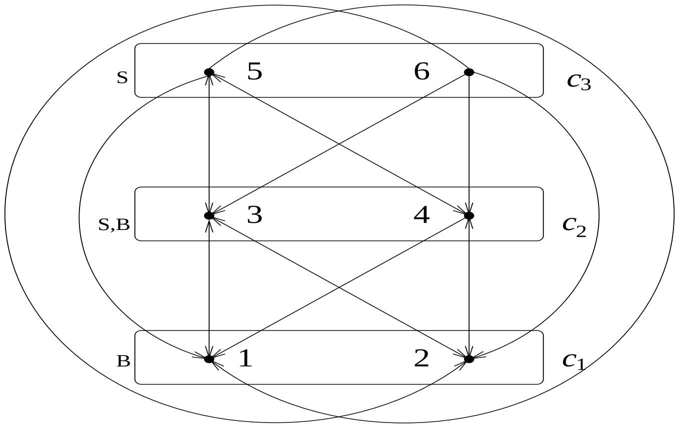


Figure 5: The filtered digraph G

For example consider the filtered digraph G pictured in Figure 5. In this case $\mathcal{C}_1 = \{1,2\}$, $\mathcal{C}_2 = \{3,4\}$ and $\mathcal{C}_3 = \{5,6\}$. We let $I_B = \{1,2\}$ and $I_S = \{2,3\}$ and G have the following directed edges:

(1,3),(2,3),(2,4)
(3,1),(3,2),(3,5), (4,1),(4,2),(4,5)
(5,1),(5,2),(5,3),(5,4),(6,1),(,6,2),(6,3),(6,4).

We let $m = 2$ so that $\mathcal{D}_1 = \emptyset$, $\mathcal{D}_2 = \{3,4\}$ and $\mathcal{D}_3 = \{5\}$. Then $A_3 = \{1,2,5\}$, $A_4 = \{1,2,5\}$ and $A_5 = \{1,2,3,4\}$. For $i = 1,2$, the tree $T^{i,*}_{\mathcal{D}_2}$ is constructed according to Case 2 since $2 \in I_S \cap I_B$ and the tree $T^{i,*}_{\mathcal{D}_2}$ is constructed according to Case 2. The trees $T_{A_3}, T_{A_4}, T_{A_5}, T_{\mathcal{D}_2}$ and $T_{\mathcal{D}_3}$ are pictured in Figure 6. We note that in Figure 6 we have left off the super scripts on the tree since $T^{1,*}_{\mathcal{D}_s} = T^{2,*}_{\mathcal{D}_s}$ for both $s = 2$ and $s = 3$.

This given, we have the following ranking lemma for trees of the form $T^{j,}_{\mathcal{D}_i}$ and $T^{j,*}_{\mathcal{D}_t}$ which easily follows from Lemma 8 and our conventions for coding the functions values on $\mathcal{D}_i$.

Lemma 4.3 *For $i = t, \ldots, k$, let $\mathcal{D}_i = \mathcal{C}_i - \{1, \ldots, m, n\} = \{d_{1,i} < \ldots < d_{t_i,i}\}$. For each r such that $m+1 \le r \le n-1$, let $A_r = \{j : (r,j) \in E\} = \{w_{0,r} < \ldots < w_{|A_i|-1,r}\}$.*

1. *Let $t \le i \le m$ and suppose that $T^j_{\mathcal{D}_i}$ and/or $T^{j,*}_{\mathcal{D}_i}$ are defined via Case 1 so that*

$$T = T_{A_{d_{1,i}}} \otimes T_{A_{d_{2,i}}} \otimes \cdots \otimes T_{A_{d_{t_i,i}}}.$$

Then if T is either $T^j_{\mathcal{D}_i}$ or $T^{j,*}_{\mathcal{D}_i}$,

1.a $L(T) = |A_{d_{1,i}}| \cdots |A_{d_{t_i,i}}|$,

1.b *Suppose $f(d_{r,i}) = w_{s_i,d_{r,i}}$ for $r = 1, \ldots, t_i$. Then f corresponds to the path*

$$p = p_1 \otimes \cdots \otimes p_{t_i} \in Path(T)$$

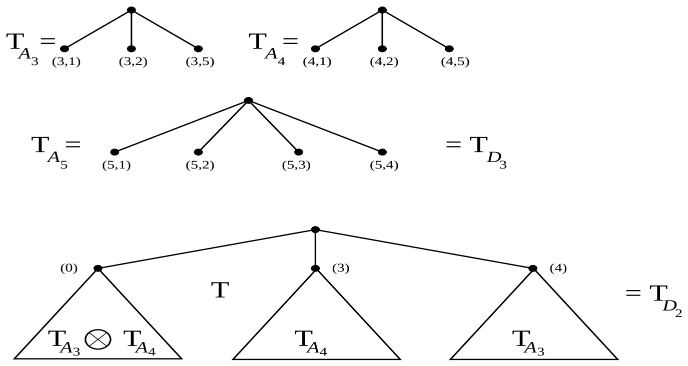


Figure 6: The rooted planar trees for G

where $rank_{T_{A_{d_{r,i}}}}(p_r) = s_r$ *and hence*

$$rank_{\mathcal{D}_i}(f) := rank_T(p) = \sum_{r=1}^{t_i} s_r \prod_{l=r+1}^{t_i} |A_{d_{l,i}}| \tag{14}$$

2. *Let* $t \le i \le m$ *and suppose that* $T^j_{\mathcal{D}_i}$ *and/or* $T^{j,*}_{\mathcal{D}_i}$ *are defined via Case 2. Then if* T *is either* $T^j_{\mathcal{D}_i}$ *or* $T^{j,*}_{\mathcal{D}_i}$*,*

   2.a $L(T) = |A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}| + \sum_{r=1}^{t_i} \frac{|A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}|}{|A_{d_{r,i}}|}$,

   2.b *Suppose* $f(d_{r,i}) = w_{s_i,d_{r,i}}$ *for* $r = 1, \ldots, t_i$*. Then* f *corresponds to the path*

$$p = (0)^\frown(p_1 \otimes \cdots \otimes p_{t_i}) \in Path(T)$$

   where $rank_{T_{A_{d_{r,i}}}}(p_r) = s_r$ *and hence*

$$rank_{\mathcal{D}_i}(f) := rank_T(p) = \sum_{r=1}^{t_i} s_r \prod_{l=r+1}^{t_i} |A_{d_{l,i}}| \tag{15}$$

   2.c *Suppose* $f(d_{u,i}) = d_{u,i}$ *and* $f(d_{r,i}) = w_{s_i,d_{r,i}}$ *for* $r \in \{1, \ldots, t_i\} - \{u\}$*. Then* f *corresponds to the path*

$$p = (d_{u,i})^\frown(p_1 \otimes \cdots \otimes p_{u-1} \otimes p_{u+1} \cdots \otimes p_{t_i}) \in Path(T)$$

where $rank_{T_{A_{d_{r,i}}}}(p_r) = s_r$ *and hence*

$$\begin{aligned} rank_{\mathcal{D}_i}(f) &:= rank_T(p) \qquad (16) \\ &= |A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}| + \sum_{r=1}^{u-1}\frac{|A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}|}{|A_{d_{r,i}}|} + \sum_{r\in\{1,\dots,t_i\}-\{u\}} s_r \prod_{l\in\{r+1,\dots,t_i\}-\{u\}} |A_{d_{l,i}}| \end{aligned}$$

Similarly, it easy to see our decoding proceedures for the trees $T^{j,}_{\mathcal{D}_i}$ and $T^{j,*}_{\mathcal{D}_t}$ and Lemma 4.2 that we have the following unranking lemma.

Lemma 4.4 *For* $i = t,\dots,k$, *let* $\mathcal{D}_i = \mathcal{C}_i - \{1,\dots,m,n\} = \{d_{1,i} < \dots < d_{t_i,i}\}$. *For each* r *such that* $m+1 \leq r \leq n-1$, *let* $A_r = \{j : (r,j) \in E\} = \{w_{0,r} < \dots < w_{|A_i|-1,r}\}$.

1. *Let* $t \leq i \leq m$ *and suppose that* $T^j_{\mathcal{D}_i}$ *and/or* $T^{j,*}_{\mathcal{D}_i}$ *are defined via Case 1 so that*

   $$T = T_{A_{d_{1,i}}} \otimes T_{A_{d_{2,i}}} \otimes \cdots \otimes T_{A_{d_{t_i,i}}}.$$

   Suppose T *is either* $T^j_{\mathcal{D}_i}$ *or* $T^{j,*}_{\mathcal{D}_i}$ *and* $p \in Path(T)$ *such that* $rank_T(p) = r_0$. *Then* $p = p_1 \otimes \cdots \otimes p_{t_i} \in Path(T)$ *where* $rank_{T_i}(p_i) = q_i$ *and*

   $$\begin{aligned} r_0 &= q_1\prod_{l=2}^{t_i}|A_{d_{l,i}}| + r_1 \text{ where } 0 \leq r_1 < \prod_{l=2}^{t_i}|A_{d_{l,i}}| \\ r_1 &= q_2\prod_{l=3}^{t_i}|A_{d_{l,i}}| + r_2 \text{ where } 0 \leq r_2 < \prod_{l=3}^{t_i}|A_{d_{l,i}}| \\ &\vdots \\ r_{t_i-2} &= q_{t_i-1}|A_{d_{t_i,i}}| + r_{t_i-1} \text{ where } 0 \leq r_{t_i-1} < |A_{d_{t_i,i}}| \\ r_{t_i-1} &= q_{t_i} \end{aligned}$$

   Moreover, p *corresponds to a function* f *such that for* $r = 1,\dots,t_i$, $f(d_{r,i}) = w_{d_{r,i},q_r}$

2. *Let* $t \leq i \leq m$ *and suppose that* $T^j_{\mathcal{D}_i}$ *and/or* $T^{j,*}_{\mathcal{D}_i}$ *are defined via Case 2. Suppose* T *is either* $T^j_{\mathcal{D}_i}$ *or* $T^{j,*}_{\mathcal{D}_i}$ *and* $p \in Path(T)$ *such that* $rank_T(p) = r_0$. *Then*

   2.a *If* $0 \leq r_0 \leq \prod |A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}| - 1$, *then* $p = (0)^\frown(p_1 \otimes \cdots \otimes p_{t_i}) \in Path(T)$ *where* $rank_{T_i}(p_i) = q_i$ *and*

   $$\begin{aligned} r_0 &= q_1\prod_{l=2}^{t_i}|A_{d_{l,i}}| + r_1 \text{ where } 0 \leq r_1 < \textstyle\prod_{l=2}^{t_i}|A_{d_{l,i}}| \\ r_1 &= q_2\prod_{l=3}^{t_i}|A_{d_{l,i}}| + r_2 \text{ where } 0 \leq r_2 < \textstyle\prod_{l=3}^{t_i}|A_{d_{l,i}}| \\ &\vdots \\ r_{t_i-2} &= q_{t_i-1}|A_{d_{t_i,i}}| + r_{t_i-1} \text{ where } 0 \leq r_{t_i-1} < |A_{d_{t_i,i}}| \\ r_{t_i-1} &= q_{t_i} \end{aligned}$$

   Moreover, p *corresponds to a function* f *such that for* $r = 1,\dots,t_i$, $f(d_{r,i}) = w_{d_{r,i},q_r}$

   2.b *If*

   $$|A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}| + \sum_{r=1}^{u-1}\frac{|A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}|}{|A_{d_{r,i}}|} \leq r_0 < |A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}| + \sum_{r=1}^{u}\frac{|A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}|}{|A_{d_{r,i}}|},$$

then let

$$\overline{r}_0 = r_0 - |A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}| + \sum_{r=1}^{u-1} \frac{|A_{d_{1,i}}|\cdots|A_{d_{t_i,i}}|}{|A_{d_{r,i}}|}$$

and

$$A_{c_{r,i}} = \begin{cases} A_{d_{r,i}} & \textit{if } i < u \\ A_{d_{r+1,i}} & \textit{if } i < u. \end{cases} \tag{17}$$

Then $p = (d_{u,i})^\frown(p_1 \otimes \cdots \otimes p_{t_i-1}) \in Path(T)$ *where* $rank_{T_i}(p_i) = q_i$ *and*

$$\begin{aligned}
\overline{r}_0 &= q_1 \prod_{l=2}^{t_i-1} |A_{c_{l,i}}| + r_1 \ \textit{where } 0 \leq r_1 < \textstyle\prod_{l=2}^{t_i-1} |A_{d_{l,i}}| \\
r_1 &= q_2 \prod_{l=3}^{t_i-1} |A_{c_{l,i}}| + r_2 \ \textit{where } 0 \leq r_2 < \textstyle\prod_{l=3}^{t_i-1} |A_{d_{l,i}}| \\
&\vdots \\
r_{t_i-3} &= q_{t_i-2}|A_{d_{t_i-1,i}}| + r_{t_i-2} \ \textit{where } 0 \leq r_{t_i-1} < |A_{d_{t_i,i}}| \\
r_{t_i-2} &= q_{t_i-1}
\end{aligned}$$

Moreover, p *corresponds to a function* f *such that (i)* $f(d_{u,i}) = d_{u,i}$*, (ii) for* $r = 1, \ldots, u_1$*,* $f(d_{r,i}) = w_{d_{r,i},q_r}$ *and (iii) for* $r = u+1, \ldots, t_i$*,* $f(d_{r,i}) = w_{d_{r,i},q_{r-1}}$

We are now in position to give the final ranking and unranking procedures for $\mathcal{T}^G_{[m]}$. Suppose the final tree $\mathcal{T}$ is of the form

$$\mathcal{T} = \mathcal{T}_1 \oplus \cdots \oplus \mathcal{T}_u \oplus \mathcal{T}^*_{u+1} \oplus \cdots \oplus \mathcal{T}^*_m. \tag{18}$$

It follows that

$$L(\mathcal{T}) = \sum_{j=1}^{m} L(\mathcal{T}_j). \tag{19}$$

Then for each $j \leq u$,

$$\mathcal{T}_j = T^j_{\mathcal{D}_t} \otimes \cdots \otimes T^j_{\mathcal{D}_k} \tag{20}$$

and for each $u < j \leq k$,

$$\mathcal{T}_j = T^{j,*}_{\mathcal{D}_t} \otimes \cdots \otimes T^{j,*}_{\mathcal{D}_k}. \tag{21}$$

By the Lemma 4.3, we can compute from G, the number of leaves of $T^j_{\mathcal{D}_i}$, $L(T^j_{\mathcal{D}_i})$ for $i = m, \ldots, k$. Hence, for $j \leq u$,

$$L(\mathcal{T}_j) = \prod_{i=m}^{k} L(T^j_{\mathcal{D}_i}) \tag{22}$$

can be computed from G. Similarly, for each $u < j \leq k$, the number of leaves of $T^{j,*}_{\mathcal{D}_i}$, $L(T^{j,*}_{\mathcal{D}_i})$ can be computed from G for $i = m, \ldots, k$. Thus for $u < j \leq m$,

$$L(\mathcal{T}_j) = \prod_{i=m}^{k} L(T^j_{\mathcal{D}_i}) \tag{23}$$

can be computed from G. It follows that

$$L(\mathcal{T}) = \sum_{j=1}^{m} L(\mathcal{T}_j). \tag{24}$$

We can compute all the information in (18)-(23) from G. This given, we then have the following ranking and unranking procedures for $\mathcal{T}^G_{[m]}$.

Ranking Forests F in $\mathcal{T}_{[m]}^G$.

Step 1. Find j such that n is the tree of F with root j.

Step 2.

Step 2.A. If $j \leq u$, then compute $f = \Theta_j(F)$ and use Lemma 4.3 to compute $rank_{T^j_{\mathcal{D}_i}}(f)$ for $i = 1, \ldots, t$. Then

$$rank_{\mathcal{T}_{[m]}^G}(F) = \sum_{i=1}^{j_1} L(\mathcal{T}_i) + \sum_{s=m}^{k} rank_{T^j_{\mathcal{D}_i}}(f) \prod_{r=s+1}^{k} L(T^j_{\mathcal{D}_r}). \tag{25}$$

Step 2.B. If $j < u$, then compute $f = \Theta_j^*(F)$ and use Lemma 4.3 to compute $rank_{T^{j,*}_{\mathcal{D}_i}}(f)$ for $i = 1, \ldots, t$. Then

$$rank_{\mathcal{T}_{[m]}^G}(F) = \sum_{i=1}^{j_1} L(\mathcal{T}_i) + \sum_{s=m}^{k} rank_{T^{j,*}_{\mathcal{D}_i}}(f) \prod_{r=s+1}^{k} L(T^{j,*}_{\mathcal{D}_r}). \tag{26}$$

For example, suppose that G is the graph pictured in Figure 5 and $m = 2$. Then it is easy to see from Figure 6 that

$$L(T^{1,*}_{\mathcal{D}_2}) = L(T^{2,*}_{\mathcal{D}_2}) = 15$$

and

$$L(T^{1,*}_{\mathcal{D}_3}) = L(T^{2,*}_{\mathcal{D}_3}) = 4.$$

Thus

$$L(\mathcal{T}_i) = L(T^{i,*}_{\mathcal{D}_2}) \cdot L(T^{i,*}_{\mathcal{D}_3}) = 60$$

for $i = 1, 2$. Now consider the forest $F \in \mathcal{T}_{[2]}^G$ picture in Figure 7.

It is easy to see that 6 is in the tree rooted at 2 so that $j = 2$ in the case. Moreover, it easy to check that $f = \Theta_2^*(F)$ is the graph pictured just below the graph of F in Figure 7. Thus on $\mathcal{D}_2$, $f(3) = 5$ and $f(4) = 1$. Thus f has no fixed points on $\mathcal{D}_2$. By Lemma 4.3, it is easy to see that

$$rank_{T^{2,*}_{\mathcal{D}_2}}(f) = 6.$$

Similarly it is easy to see that since $f(5) = 3$,

$$rank_{T^{2,*}_{\mathcal{D}_3}}(f) = 2.$$

Thus

$$\begin{aligned} rank_{\mathcal{T}_{[m]}^G}(F) &= L(\mathcal{T}_1) + \sum_{s=2}^{3} rank_{T^{j,*}_{\mathcal{D}_i}}(f) \prod_{r=s+1}^{3} L(T^{j,*}_{\mathcal{D}_r}) \\ &= 60 + (6 \cdot 4) + 2 = 86. \end{aligned}$$

The unrank procedure for $\mathcal{T}_{[m]}^G$ is as follows.

Finding the forest $F \in \mathcal{T}_{[m]}^G$ such that $rank_{\mathcal{T}_{[m]}^G}(F) = a$.

Step 1. Find j such that

$$\sum_{i=1}^{j-1} L(\mathcal{T}_i) \leq r \leq \sum_{i=1}^{j} L(\mathcal{T}_i).$$

Thus n will be in the tree rooted at j in F. Set $b_{m-1} = a - \sum_{i=1}^{j-1} L(\mathcal{T}_i)$.

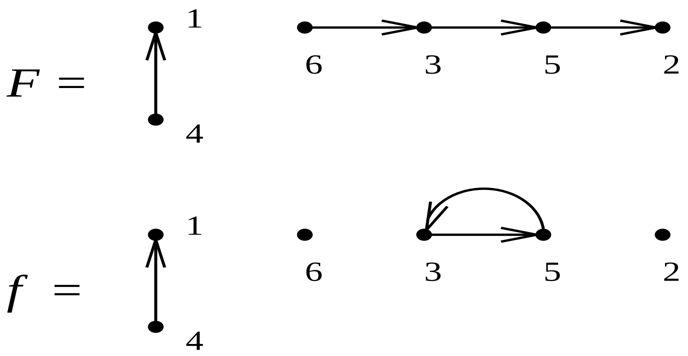


Figure 7: The forest F

Step 2.

Step 2.a. Suppose $j \leq u$. $p = p_m \otimes \cdots \otimes p_k \in T^j_{\mathcal{D}_m} \otimes \cdots \otimes T^j_{\mathcal{D}_k}$ such that

$$rank_{\mathcal{T}_j}(p) = b_{m-1}.$$

.

$$b_{m-1} = q_m \prod_{l=m+1}^{k} L(T^j_{\mathcal{D}_l}) + b_m \text{ where } 0 \leq b_m < \textstyle\prod_{l=m+1}^{k} L(T^j_{\mathcal{D}_l}) \quad (27)$$

$$b_m = q_{m+1} \prod_{l=m+2}^{k} L(T^j_{\mathcal{D}_l}) + b_{m+1} \text{ where } 0 \leq r_2 < \textstyle\prod_{l=m+2}^{k} L(T^j_{\mathcal{D}_l}) \quad (28)$$

$$\vdots \quad (29)$$

$$b_{k-2} = q_{k-1} L(T^j_{\mathcal{D}_k}) + b_{k-1} \text{ where } 0 \leq b_{k-1} < L(T^j_{\mathcal{D}_k}) \quad (30)$$

$$b_{k-1} = b_k \quad (31)$$

Then for $s = m, \ldots, k$, $rank_{T^j_{\mathcal{D}_s}}(p_s) = q_s$.

Step 2.b. Suppose $u < j \leq m$. $p = p_m \otimes \cdots \otimes p_k \in T^{j,*}_{\mathcal{D}_m} \otimes \cdots \otimes T^{j,*}_{\mathcal{D}_k}$ such that

$$rank_{\mathcal{T}_j}(p) = b_{m-1}.$$

.

$$b_{m-1} = q_m \prod_{l=m+1}^{k} L(T^{j,*}_{\mathcal{D}_l}) + b_m \text{ where } 0 \leq b_m < \textstyle\prod_{l=m+1}^{k} L(T^{j,*}_{\mathcal{D}_l}) \quad (32)$$

$$b_m = q_{m+1} \prod_{l=m+2}^{k} L(T^{j,*}_{\mathcal{D}_l}) + b_{m+1} \text{ where } 0 \leq r_2 < \textstyle\prod_{l=m+2}^{k} L(T^{j,*}_{\mathcal{D}_l}) \quad (33)$$

$$\vdots \quad (34)$$

$$b_{k-2} = q_{k-1} L(T^{j,*}_{\mathcal{D}_k}) + b_{k-1} \text{ where } 0 \leq b_{k-1} < L(T^{j,*}_{\mathcal{D}_k}) \quad (35)$$

$$b_{k-1} = b_k \quad (36)$$

Then for $s = m, \ldots, k$, $rank_{T^{j,*}_{\mathcal{D}_s}}(p_s) = q_s$.

Step 3.

Step 3.a. Suppose $j \leq u$. Then having found $q_m, \ldots, q_k$ from Step 2, we can use Lemma 4.4 to find p_i such that $rank_{\mathcal{T}^{j}_{\mathcal{D}_i}}(p_i) = q_i$ for $i = m, \ldots, k$. Moreover by Lemma 4.4, each p_i corresponding to a function $f_i : \mathcal{D}_i \to [n]$ such that $f = \bigcup_{i=m}^{k} f_i$ is an element of $\mathcal{F}_n(G, \mathbf{F}, m)$. Let $F = \Theta_j(f)$. Then $rank_{\mathcal{T}^{G}_{[m]}}(F) = a$.

Step 3.b. Suppose $j > u$. Then having found $q_m, \ldots, q_k$ from Step 2, we can use Lemma 4.4 to find p_i such that $rank_{\mathcal{T}^{j,*}_{\mathcal{D}_i}}(p_i) = q_i$ for $i = m, \ldots, k$. Moreover by Lemma 4.4, each p_i corresponding to a function $f_i : \mathcal{D}_i \to [n]$ such that $f = \bigcup_{i=m}^{k} f_i$ is an element of $\mathcal{F}^*_n(G, \mathbf{F}, m)$. Let $F = \Theta^*_j(f)$. Then $rank_{\mathcal{T}^{G}_{[m]}}(F) = a$.

For example, suppose that we want to find the forest F such that $rank_{\mathcal{T}^{G}_{[m]}}(F) = 45$ for the graph pictured in Figure 5. Since $45 < L(\mathcal{T}_1) = 60$, then 6 will be in the tree rooted at 1 in F. Now $\mathcal{T}_1 = T^{1,*}_{\mathcal{D}_2} \otimes T^{1,*}_{\mathcal{D}_3}$ where $L(T^{1,*}_{\mathcal{D}_2}) = 15$ and $L(T^{1,*}_{\mathcal{D}_4}) = 4$. Since $45 = 1 + 11 \cdot 4$, it follows that the path p such that $rank_{\mathcal{T}_1}(p) = 45$ is of the form

$$p = p_2 \otimes p_3$$

where $rank_{T^{1,*}_{\mathcal{D}_2}}(p_2) = 11$ and $rank_{T^{1,*}_{\mathcal{D}_3}}(p_3) = 1$. Then it is easy to see from Figure 6 that the path p_2 corresponds to the function f_2 on $\mathcal{D}_2$ such that $f_2(3) = 3$ and $f_2(4) = 5$. Similarly, p_3 corresponds to the function f_3 on $\mathcal{D}_3$ such that $f_3(5) = 2$. Thus the function $f = f_2 \cup f_3$ such that $rank_{\mathcal{T}}(f) = 45$ is given by:

$$\begin{aligned} f(3) &= 3 \\ f(4) &= 5 \\ f(5) &= 2 \end{aligned}$$

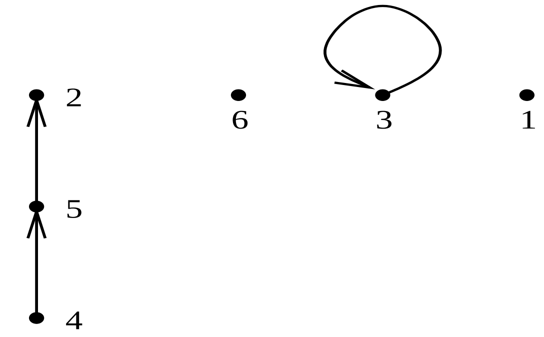
The graph of f is pictured in Figure 8. It then easy to check that $\Theta^*_1(f) = F$ where F is also pictured in Figure 8.

It is now easy to check that once we build the appropriate data structures and trees $T^{j}_{\mathcal{D}_i}$ and $T^{j,*}_{\mathcal{D}_i}$ from G that both the ranking and unranking algorithms for $\mathcal{T}^{G}_{[m]}$ can be carried out in linear time.

5 Examples

Example 5.1. Perhaps the simplest examples of a filtered digraph are the complete graphs K_n. Consider the complete digraph $G = ([n], E)$ where E is the set of all pairs (i, j), $i \neq j$. The graph G corresponds in the obvious manner to K_n, the complete undirected graph on n. We take the composition $\mathbf{F} = (c_1, \ldots, c_n)$ of n where $c_i = 1$ for all i. The filtration $\{\mathcal{C}_i \mid i = 1, \ldots, n\}$ associated with this composition of n is the discrete partition. Note that $G = ([n], E)$ is a filtered digraph with respect to $\mathbf{F}$, $I_B = \{1, 2, \ldots, n-1\}$, and

$\mathbf{f}$ =



$\mathbf{F}$ =

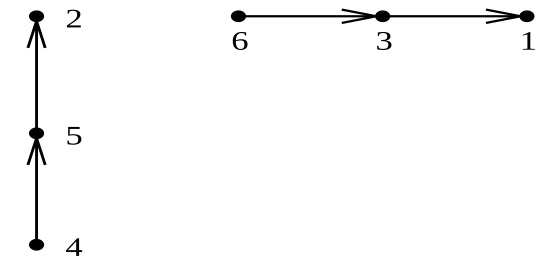


Figure 8: The forest F

$I_S = \{2, 3, \ldots, n\}$. Note that in this case, $\mathcal{F}_n(G, \mathbf{F}, m) = \mathcal{F}_n^*(G, \mathbf{F}, m)$ so that identity (5) of Theorem 2.4 becomes

$$\begin{aligned}\sum_{T \in \mathcal{T}^G_{[m]}} W(T) &= q_n(t_1 + \cdots + t_m) \sum_{f \in \mathcal{F}_n(G,\mathbf{F},m)} W(f) \\ &= q_n(t_1 + \cdots + t_m) \prod_{j=m+1}^{n-1} (q_j(t_1 + \ldots t_j) + p_j(s_{j+1} + \ldots + s_n)). \end{aligned} \tag{37}$$

The set of root-directed spanning forests $T \in \mathcal{T}^G_{[m]}$ with roots in $[m]$ corresponds in a natural way to the undirected spanning forests of K_n with component trees rooted at the vertices $[m]$.

In the special case when $m = 1$, (37) becomes

$$\sum_{T \in \mathcal{T}^G_{[1]}} W(T) = q_n t_1 \prod_{j=2}^{n-1} (q_j(t_1 + \ldots t_j) + p_j(s_{j+1} + \ldots + s_n)). \tag{38}$$

When we set all the variables equal to one, (38) reduces to Cayley's formula for the number of trees on n vertices, $|\mathcal{T}^G_{[1]}| = n^{n-2}$. Similarly, (37) becomes $\mathcal{T}^G_{[m]} = n^{n-1-t}$.

In this case, the problems of ranking and unranking spanning forests in $\mathcal{T}^G_{[m]}$ is very simple. That is, for each $m \leq j \leq n-1$, $\mathcal{D}_| = \{j\}$ and since $j \in I_S \cap I_B$, the trees $T^i_{\mathcal{D}_j} = T^{i,*}_{\mathcal{D}_j}$ are the same for all $i = 1, \ldots, m$ and are pictured in Figure 9. It follows that if $f(j) = k$, then the corresponding path p_k in $T_{\mathcal{D}_j}$ has

$$rank_{T_{\mathcal{D}_j}}(p_k) = \begin{cases} k-1 & \text{if } k < j \\ n-1 & \text{if } k = j \\ k-2 & \text{if } k > j \end{cases} \tag{39}$$

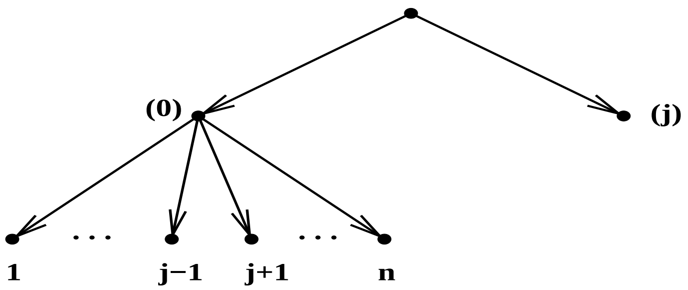


Figure 9: The trees $T^i_{\mathcal{D}_j} = T^{i,*}_{\mathcal{D}_j}$ for K_n.

However, there is a key property of our bijections that allows us to rank and unrank much more than just the collection of root directed spanning forests of K_n that are rooted at $1,\ldots,m$. That is, the only edges that change under either the bijection Θ_i or Θ^*_i are descent edges, i.e. edges of the form (i,j) where $i \geq j$. Now suppose that we want to rank and unrank the set $\mathcal{T}^G_{[m],\mathfrak{E}}$ of root directed spanning forests of K_n that contain a prespecified set of ascent edges $\mathfrak{E} = \{(s_1,t_1),\ldots,(s_r,t_r)\}$ where $s_1,\ldots,s_r \in \{m+1,\ldots,n-1\}$. In this case, we can realize this set as the set of root directed spanning forests of a slightly modified graph $G_{\mathfrak{E}} = ([n],E_{\mathfrak{E}})$. That is, suppose we take the composition $\mathbf{F} = (c_1,\ldots,c_n)$ of n where $c_i = 1$ for all i. Again the filtration $\{\mathcal{C}_i \mid i = 1,\ldots,n\}$ associated with this composition of n is the discrete partition. Then the graph $G_{\mathfrak{E}}$ differs from our graph G corresponding to the complete graph only in that, for $i = 1,\ldots,r$, the only directed edges of the form (s_i,k) that are in $E_{\mathfrak{E}}$ are the edges (s_i,t_i), instead of allowing all directed edges (s_i,k) . Thus we let $I_{B_{\mathfrak{E}}} = \{1,2,\ldots,n-1\}$, and $I_{S_{\mathfrak{E}}} = \{2,3,\ldots,n\} - \{s_1,\ldots,s_r\}$. In terms of our ranking and unranking algorithms, the only difference is that the tree $T_{\mathcal{D}_{s_i}}$ becomes the tree pictured in Figure 10 (A).

Similarly suppose that we want to rank and unrank the set $\mathcal{T}^G_{[m],\mathfrak{V}}$ of root direct spanning forests of K_n where we insist that the set of edges out of the vertices $\mathfrak{V} = \{s_1,\ldots,s_r\}$ where $s_1,\ldots,s_r \in \{m+1,\ldots,n-1\}$ are ascent edges. In this case, we can realize this set as the set of root directed spanning forests of a another modified graph $G_{\mathfrak{V}} = ([n],E_{\mathfrak{V}})$. Again we take the composition $\mathbf{F} = (c_1,\ldots,c_n)$ of n where $c_i = 1$ for all i so that the filtration $\{\mathcal{C}_i \mid i = 1,\ldots,n\}$ associated with this composition of n is the discrete partition. Then the graph $G_{\mathfrak{V}}$ differs from our graph G corresponding to the complete graph only in that, for $i = 1,\ldots,r$, the only directed edges of the form (s_i,k) that are in $E_{\mathfrak{E}}$ are the edges (s_i,k) where $k > s_i$ instead of allowing all directed edges (s_i,k) . Thus we let $I_{B_{\mathfrak{V}}} = \{1,2,\ldots,n-1\}$, and $I_{S_{\mathfrak{V}}} = \{2,3,\ldots,n\} - \{s_1,\ldots,s_r\}$. In terms of our ranking and unranking algorithms, the only difference is that the tree $T_{\mathcal{D}_{s_i}}$ becomes the tree pictured in Figure 10 (B).

For example, suppose that G is the graph that corresponds to the complete graph on 10 vertices and we have $m = 1$ so that we are ranking all Cayley trees on n vertices. There are 10^8 such trees. If we set $\mathfrak{E} = \{(4,10),(6,8)\}$, then ranking and unranking with respect to $G_{\mathfrak{E}}$ corresponds to ranking and unranking

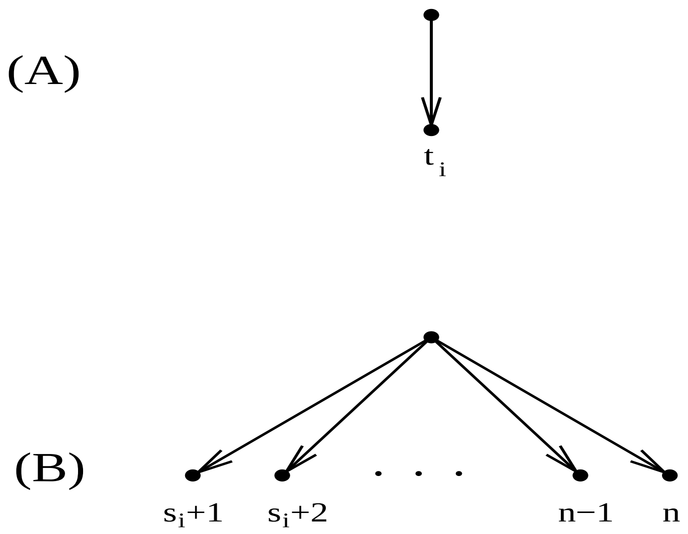


Figure 10: The trees $T^i_{\mathcal{D}_j} = T^{i,*}_{\mathcal{D}_j}$ for $G_{\mathfrak{C}}$ and $G_{\mathfrak{V}}$.

with respect to all Cayley trees that contain the edges $(4,10)$ and $(6,8)$. If we let $\mathfrak{V} = \{4,10\}$, then ranking and unranking with respect to $G_{\mathfrak{V}}$ corresponds to ranking and unranking with respect to all Cayley trees for which the directed edges out vertices 4 and 10 are ascent edges. Now consider the tree T pictured in Figure 11 (A) and its corresponding $f = \Theta_1^*(T)$ which is pictured in Figure 11 (B).

i	$f(i)$	$rank_{T_{\mathcal{D}_i}}(p_i)$ for G	contribution	$rank_{T_{\mathcal{D}_i}}(p_i)$ for $G_{\mathfrak{C}}$	contribution	$rank_{T_{\mathcal{D}_i}}(p_i)$ for $G_{\mathfrak{V}}$	contribution
2	2	9	9×10^7	9	9×10^5	9	$9 \times 24 \times 10^5$
3	8	6	6×10^6	6	6×10^4	6	$6 \times 24 \times 10^4$
4	10	8	8×10^5	0	0×10^4	5	$5 \times 4 \times 10^4$
5	5	9	9×10^4	9	9×10^3	9	$9 \times 4 \times 10^3$
6	8	6	6×10^3	0	0×10^3	1	1×10^3
7	2	1	1×10^2	1	1×10^2	1	1×10^2
8	3	2	2×10	2	2×10	2	2×10
9	8	7	7	7	7	7	7

Table 1

Table 1 then gives the rank of the path p_i in the tree $T_{\mathcal{D}_i}$ corresponding to the function value $f(i)$ and its contribution to the final rank in the calculations of $rank_{\mathcal{T}^G_{[1]}}(T)$, $rank_{\mathcal{T}^G_{[1],\mathfrak{C}}}(T)$, and $rank_{\mathcal{T}^G_{[1],\mathfrak{V}}}(T)$ respectively, according to our general ranking procedure. It follows that

$$\begin{aligned} rank_{\mathcal{T}^G_{[1]}}(T) &= 96,896,127 \\ rank_{\mathcal{T}^G_{[1],\mathfrak{C}}}(T) &= 969,127 \\ rank_{\mathcal{T}^G_{[1],\mathfrak{V}}}(T) &= 23,277,127 \end{aligned}$$

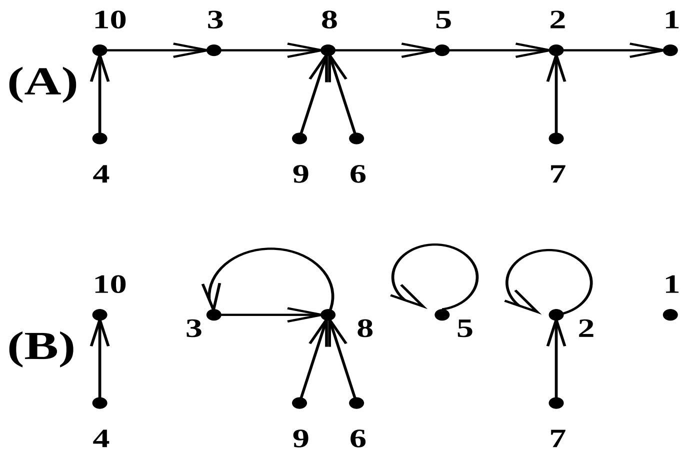


Figure 11: The tree T and its corresponding function.

On the other hand suppose that we want to find the trees of rank 550,054 with respect to these three classes of spanning trees. That is, suppose that we want to find the trees T_1, T_2 and T_3 such that

$$\begin{aligned} rank_{\mathcal{T}^G_{[1]}}(T_1) &= 550,054 \\ rank_{\mathcal{T}^G_{[1],\mathfrak{C}}}(T_2) &= 550,054 \\ rank_{\mathcal{T}^G_{[1],\mathfrak{Y}}}(T_3) &= 550,054 \end{aligned}$$

In each case, we must find the paths $p^j = (p^j_2, \ldots, p^j_9)$ for $j = 1, 2, 3$ through the corresponding trees that have 550, 054 leaves to its left. In Table 2, we have exhibited to the corresponding decomposition of 550,054 according to our unranking algorithm, the corresponding to ranks of the paths p^j_s in the trees $T_{\mathcal{D}_s}$, and value of the correspond functions f_{T_j}. Then in Figure 12, we have pictured the trees T_1, T_2 and T_3 that arise by applying the Θ^*_1 bijection to f_{T_1}, f_{T_2}, and f_{T_3} respectively.

i	550,054 for G	$rank_{T_{\mathcal{D}_i}}(p^1_i)$ for G	$f_{T_1}(i)$	550,054 for $G_{\mathfrak{C}}$	$rank_{T_{\mathcal{D}_i}}(p^2_i)$ for $G_{\mathfrak{C}}$	$f_{T_2}(i)$	550,054 for $G_{\mathfrak{Y}}$	$rank_{T_{\mathcal{D}_i}}(p^3_i)$ for $G_{\mathfrak{Y}}$	$f_{T_3}(i)$
2	0×10^7	0	1	5×10^5	5	7	$0 \times 24 \times 10^5$	0	1
3	0×10^6	0	1	5×10^4	5	7	$2 \times 24 \times 10^4$	2	4
4	5×10^5	5	7	0×10^4	0	10	$1 \times 4 \times 10^4$	1	6
5	5×10^4	5	7	0×10^3	0	1	$7 \times 4 \times 10^3$	7	9
6	0×10^3	0	1	0×10^3	0	8	2×10^3	2	9
7	0×10^2	0	1	0×10^2	0	1	0×10^2	0	1
8	5×10	5	6	5×10	5	6	5×10	5	6
9	4×1	4	5	4×1	4	5	4×1	4	5

Table 2

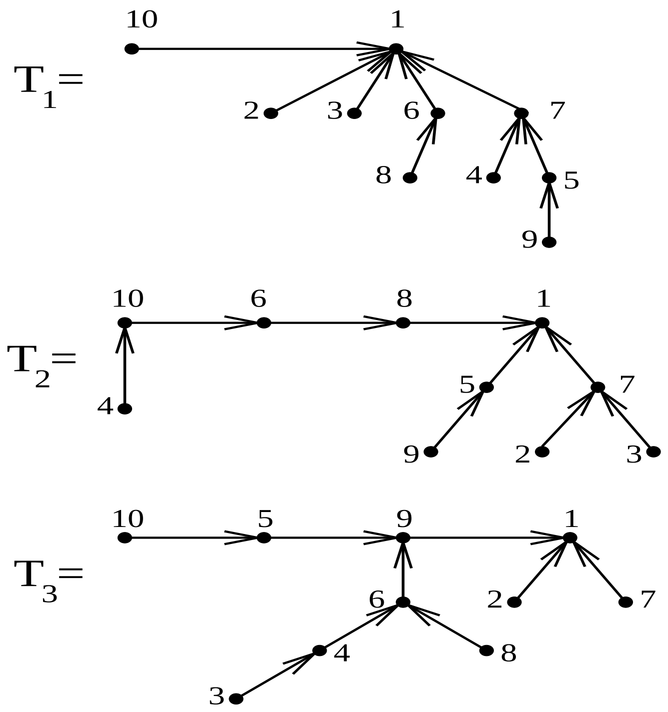


Figure 12: The trees T_1, T_2, and T_3.

Example 5.2. We next consider the case where $G = K_{n_1,\ldots,n_k}$ is the n-vertex multipartite graph with k parts of size $n_1, \ldots, n_k$. Thus, $n = n_1 + \cdots + n_k$. In this case, Onodera [7] showed that the number of spanning trees of $K_{n_1,\ldots,n_k}$ is $n^{k-2} \prod_{i=1}^{k} (n - k_i)^{k_i - 1}$.

We start with the case $k = 2$. In the context of this paper, the composition is $\mathbf{F} = (n_1, n_2)$. The filtration is $\{\mathcal{C}_i \mid i = 1, 2\}$ where $\mathcal{C}_1 = \{1, 2, \ldots, n_1\}$ and $\mathcal{C}_2 = \{n_1 + 1, n_1 + 2, \ldots, n\}$. $\mathcal{C}_1$ is a base and $\mathcal{C}_2$ is a summit. We assume that $1 \le m \le n_1$ so that equation (5) becomes

$$\sum_{T \in \mathcal{T}^G_{[m]}} W(T) = q_n(t_1 + \cdots + t_m) \sum_{f \in \mathcal{F}_n(G,\mathbf{F},m)} W(f)$$

$$= \quad q_n(t_1 + \cdots + t_m) \prod_{j=m+1}^{n_1} p_j(s_{n_1+1} + \ldots + s_n) \prod_{j=1}^{n_2-1} q_{n_1+j}(t_1 + \ldots + t_{n_1}). \tag{40}$$

The set of root-directed spanning forests $T \in \mathcal{T}^G_{[m]}$, with roots $[m]$, corresponds in a natural way to the undirected spanning forests of K_{n_1,n_2}, with component trees rooted at the vertices $[m]$. As in the previous example, by specializing the weights by setting $p_i = q_i = s_i = t_i = 1$, we get the following formula for the number of root directed spanning forests of K_{n_1,n_2} rooted at $1, \ldots, m$.

$$|T \in \mathcal{T}^G_{[m]}| = m n_1^{n_2-1} n_2^{n_1-m}. \tag{41}$$

Next we consider the more difficult case of $k > 2$. In this case we will need the full power of Theorem 2.4 in that we will need to consider functions whose associated digraphs have loops. In particular, $K_{n_1,\ldots,n_k}$ is the n-vertex multipartite graph with $k > 2$ parts of size $n_1, \ldots, n_k$. Thus, $n = n_1 + \cdots + n_k$ and the composition is $\mathbf{F} = (n_1, \ldots n_k)$. Again, we denote the partial sum $n_1 + \cdots + n_j$ by N_j and let $N_0 = 0$. The filtration in this case is $\{\mathcal{C}_i \mid i = 1, \ldots k\}$ where

$$\mathcal{C}_1 = \{1, 2, \ldots, N_1\}, \mathcal{C}_2 = \{N_1 + 1, N_1 + 2, \ldots, N_2\}, \ldots, \mathcal{C}_k = \{N_{k-1} + 1, N_{k-1} + 2, \ldots, N_k\}.$$

$\mathcal{C}_1, \ldots, \mathcal{C}_{k-1}$ are bases and $\mathcal{C}_2, \ldots, \mathcal{C}_k$ are summits. Take $1 \le m \le N_{k-1}$ and assume that $m \in \mathcal{C}_t$. Identity (5) of Theorem 2.4 then becomes

$$\left[q_n(t_1 + \cdots + t_{N_{t-1}})\tilde{B}_t \prod_{j=t+1}^{k} \tilde{A}_j\right] + \left[q_n(t_{N_{t-1}+1} \cdots + t_m)\tilde{D}_t \prod_{j=t+1}^{k} \tilde{A}_j\right] = \sum_{T \in \mathcal{T}^G_{[m]}} W(T). \tag{42}$$

Here for $t < j < k$,

$$\tilde{A}_j = \prod_{i=N_{j-1}+1}^{N_j} \left(p_i(s_{N_j+1} + \cdots + s_n) + q_i(t_1 + \cdots + t_{N_{j-1}})\right)$$

$$+ \sum_{r=N_{j-1}+1}^{N_j} q_r t_r \prod_{\substack{i=N_{j-1}+1 \\ i \neq r}}^{N_j} \left(p_i(s_{N_j+1} + \cdots + s_n) + q_i(t_1 + \cdots + t_{N_{j-1}})\right).$$

The summation in the above expression takes into account the possible loops in the digraphs corresponding to the functions $f \in \mathcal{F}_n(G, \mathbf{F}, m)$. Similarly,

$$\tilde{B}_t = \prod_{i=m+1}^{N_t} \left(p_i(s_{N_t+1} + \cdots + s_n) + q_i(t_1 + \cdots + t_{N_{t-1}})\right)$$

$$+ \sum_{r=m+1}^{N_t} q_r t_r \prod_{\substack{i=m+1 \\ i \neq r}}^{N_t} \left(p_i(s_{N_t+1} + \cdots + s_n) + q_i(t_1 + \cdots + t_{N_{t-1}})\right)$$

and

$$\tilde{D}_t = \prod_{i=m+1}^{N_t} \left(p_i(s_{N_t+1} + \cdots + s_n) + q_i(t_1 + \cdots + t_{N_{t-1}})\right).$$

Note that if $t = 1$, $\tilde{D}_t = \prod_{i=m+1}^{N_t} p_i(s_{N_t+1} + \cdots + s_n)$. Also, if $m = N_t$, then we take $\tilde{B}_t = \tilde{D}_t = 1$. Finally, for $j = k$, we have

$$\tilde{A}_k = \prod_{i=N_{k-1}+1}^{n-1} q_i(t_1 + \cdots t_{N_{k-1}}).$$

If we set $q_i = t_i = p_i = s_i = 1$, then equation (42) gives the cardinality $|\mathcal{T}^G_{[m]}|$. In this case, $A_j = (n - n_j)^{n_j - 1}$, $t < j \le k$. $B_t = (n - n_t)^{N_t - m - 1}(n + N_{t-1} - m)$ and $D_t = (n - n_t)^{N_t - m}$. We obtain,

$$n^{k-t-1}\left(N_{t-1}(n - n_t)^{N_t - m - 1}(n + N_{t-1} - m) + (m - N_{t-1})(n - n_t)^{N_t - m}\right) \prod_{j=t+1}^{k} (n - n_j)^{n_j - 1} \tag{43}$$

for $|\mathcal{T}^G_{[m]}|$. Note that if $t = 1$, $N_{t-1} = N_0 = 0$ and thus we obtain

$$n^{k-2}\left(m(n - n_t)^{n_1 - m}\right) \prod_{j=2}^{k} (n - n_j)^{n_j - 1} = |T \in \mathcal{T}^G_{[m]}|. \tag{44}$$

If $m = 1$ this identity becomes the formula of Onodera [7]:

$$n^{k-2} \prod_{j=1}^{k} (n - n_j)^{n_j - 1} = |\mathcal{T}^G_1|. \tag{45}$$

Next we shall explicitly translate our unrank procedure for $|\mathcal{T}^G_{[m]}|$ in the case where $m < n_1$ so that all the roots of any forest are in $\mathcal{C}_1$. In this case it is easy to check that $\mathcal{D}_1 = \{t + 1, \ldots, n_1\}$, $\mathcal{D}_h =$

$\{N_{h-1}+1,\ldots,N_{h-1}+n_h\}$ for $h=2,\ldots,k-1$ and $\mathcal{D}_k=\{N_{k-1}+1,\ldots,N_{k-1}+n_k-1\}$. Moreover it is easy to check from our definitions of $T^{j,*}_{\mathcal{D}_j}$ that for $j=1,\ldots,t$,

$$\begin{aligned} L_1 &= L(T^{j,*}_{\mathcal{D}_1}) = (n-n_1)^{n_1-t} && (46)\\ L_h &= L(T^{j,*}_{\mathcal{D}_h}) = n(n-n_h)^{n_h-1} \text{ for } h=2,\ldots,k-1 && (47)\\ L_k &= L(T^{j,*}_{\mathcal{D}_k}) = (n-n_k)^{n_k-1} && (48)\end{aligned}$$

One can easily see from Figure 4 that the number of leaves of $\mathcal{T}$ is given by

$$L(\mathcal{T}) = m\cdot L_1\cdots L_k. \tag{49}$$

Unranking Procedure for $\mathcal{T}^G_{[m]}$ where $G=K_{n_1,\ldots,n_k}$ and $m<n_1$.

Goal: Find the forest $F\in\mathcal{T}^G_{[m]}$ such that $rank_{\mathcal{T}^G_{[m]}}(F)=a_0$.

Step 1. Find $r\le m$ such that

$$(r-1)L_1\cdots L_k \le a_0 < rL_1\cdots L_k.$$

This means that F will be of the form $\Theta^*_r(f)$ for some function $f\in\mathcal{F}^*_n(G,\mathbf{F},m)$ where

$$rank_{\mathcal{T}_r}(f) = b_0 = a_0-(r-1)L_1\cdots L_k. \tag{50}$$

Step 2.
Goal: Find $p_1\otimes\cdots\otimes p_k\in T^{r,*}_{\mathcal{D}_1}\otimes\cdots\otimes T^{r,*}_{\mathcal{D}_k}$ such that $rank_{\mathcal{T}_r}(p_1\otimes\cdots\otimes p_k)=b_0$.

Find $c_1,\ldots,c_k$ such that

$$\begin{aligned} b_0 &= c_1\cdot L_2\cdots L_k + b_1 \text{ where } 0\le b_1 < L_2\cdots L_k\\ b_1 &= c_2\cdot L_3\cdots L_k + b_2 \text{ where } 0\le b_2 < L_3\cdots L_k\\ &\vdots\\ b_{k-2} &= c_{k-1}\cdot L_k + b_{k-1} \text{ where } 0\le b_k < L_k\\ b_{k-1} &= c_k.\end{aligned}$$

Then $rank_{T_{\mathcal{D}_j}}(p_j)=c_j$.

Step 3.
Goal: Find f_j equals f restricted to $\mathcal{D}_j$ such that $rank_{T^{r,*}_{\mathcal{D}_j}}(f_j)=rank_{T^{r,*}_{\mathcal{D}_j}}(f_j)=c_j$ for $j=1,\ldots,k$.

Step 3: Case 1.
Set $d_{0,1}=c_1$ and compute $e_{s,1}$ for $s=1,\ldots n_1-m$ where

$$\begin{aligned} d_{0,1} &= e_{1,1}\cdot(n-n_1)^{n_1-m-1}+d_{1,1} \text{ where } 0\le d_{1,1}<(n-n_1)^{n_1-m-1}\\ d_{1,1} &= e_{2,1}\cdot(n-n_1)^{n_1-m-2}+d_{2,1} \text{ where } 0\le d_{2,1}<(n-n_1)^{n_1-m-2}\\ &\vdots\\ d_{n_1-m-2,1} &= e_{n_1-m-1,1}\cdot(n-n_1)+d_{n_1-m-1,1} \text{ where } 0\le d_{n_1-m-1,1}<(n-n_1)\\ d_{n_1-m-1,1} &= e_{n_1-m,1}.\end{aligned}$$

Then for $s=1,\ldots,n_1-m$,

$$f(m+s) = N_1+1+e_{s,1}. \tag{51}$$

Step 3: Case 2. For $j=2,\ldots,k-1$, do the following.

Subcase 2.1. $0 \le c_j < (n-n_j)^{n_j}$.
In this case, f will have no fixed points on $\mathcal{C}_j$. Set $d_{0,j} = c_j$ and compute $e_{s,j}$ for $s = 1, \dots n_j$ where

$$\begin{aligned}
d_{0,j} &= e_{1,j} \cdot (n-n_j)^{n_j-1} + d_{1,j} \text{ where } 0 \le d_{1,j} < (n-n_j)^{n_j-1} \\
d_{1,j} &= e_{2,j} \cdot (n-n_j)^{n_j-2} + d_{2,j} \text{ where } 0 \le d_{2,j} < (n-n_j)^{n_j-2} \\
&\vdots \\
d_{n_j-2,j} &= e_{n_j-1,j} \cdot (n-n_j) + d_{n_j-1,j} \text{ where } 0 \le d_{n_j-1,j} < (n-n_j) \\
d_{n_j-1,j} &= e_{n_j,j}.
\end{aligned}$$

Then for $s = 1, \dots, n_j$,

$$f(N_{j-1}+s) = \begin{cases} 1 + e_{s,j} & \text{if } e_{s,j} < N_{j-1} \\ N_j + 1 + e_{s,j} & \text{if } e_{s,j} \ge N_{j-1}. \end{cases} \tag{52}$$

Subcase 2.2. $(n-n_j)^{n_j} + (p-1)(n-n_j)^{n_j-1} \le c_j < (n-n_j)^{n_j} + p(n-n_j)^{n_j-1}$.
In this case, $N_{j-1}+p$ will be the only fixed point of f on $\mathcal{C}_j$. Set

$$d_{0,j} = c_j - [(n-n_j)^{n_j} + (p-1)(n-n_j)^{n_j-1}]$$

and compute $e_{s,j}$ for $s = 1, \dots n_j - 1$ where

$$\begin{aligned}
d_{0,j} &= e_{1,j} \cdot (n-n_j)^{n_j-2} + d_{1,j} \text{ where } 0 \le d_{1,j} < (n-n_j)^{n_j-2} \\
d_{1,j} &= e_{2,j} \cdot (n-n_j)^{n_j-3} + d_{2,j} \text{ where } 0 \le d_{2,j} < (n-n_j)^{n_j-3} \\
&\vdots \\
d_{n_j-3,j} &= e_{n_j-2,j} \cdot (n-n_j) + d_{n_j-2,j} \text{ where } 0 \le d_{n_j-2,j} < (n-n_j) \\
d_{n_j-2,j} &= e_{n_j-1,j}.
\end{aligned}$$

In this case, $f(N_{j-1}+p) = N_{j-1}+p$. For $s = 1, \dots, p-1$,

$$f(N_{j-1}+s) = \begin{cases} 1 + e_{s,j} & \text{if } e_{s,j} < N_{j-1} \\ N_j + 1 + e_{s,j} & \text{if } e_{s,j} \ge N_{j-1}. \end{cases} \tag{53}$$

and for $s = 1, \dots, n_j - p$,

$$f(N_{j-1}+p+s) = \begin{cases} 1 + e_{p+s-1,j} & \text{if } e_{p+s-1,j} < N_{j-1} \\ N_j + 1 + e_{p+s-1,j} & \text{if } e_{p+s-1,j} \ge N_{j-1}. \end{cases} \tag{54}$$

Step 3: Case 3.
Set $d_{0,k} = c_k$ and compute $e_{s,k}$ for $s = 1, \dots n_k - 1$ where

$$\begin{aligned}
d_{0,k} &= e_{1,k} \cdot (n-n_k)^{n_k-2} + d_{1,k} \text{ where } 0 \le d_{1,k} < (n-n_k)^{n_k-2} \\
d_{1,k} &= e_{2,k} \cdot (n-n_k)^{n_k-3} + d_{2,k} \text{ where } 0 \le d_{2,k} < (n-n_k)^{n_k-3} \\
&\vdots \\
d_{n_k-3,k} &= e_{n_k-2,k} \cdot (n-n_k) + d_{n_k-2,k} \text{ where } 0 \le d_{n_k-2,k} < (n-n_k) \\
d_{n_1-2,k} &= e_{n_k-1,k}.
\end{aligned}$$

Then for $s = 1, \dots, n_k - 1$,

$$f(N_{k-1}+s) = 1 + e_{s,k}. \tag{55}$$

Step 4. Having constructed $f \in \mathcal{F}_n^*(G, \mathbf{F}, m)$ in Step 3, then $F = \Theta_r^*(f)$.

The ranking procedure for $F \in \mathcal{T}^G_{[m]}$ is to find $1 \leq r \leq m$ such that vertex n is in the tree with root j. Then find the function $f \in \mathcal{F}^*_n(G, \mathbf{F}, m)$ such that $(\Theta^*_r)^{-1}(F) = f$. We can then find the rank of F by essentially reversing the steps of our unranking procedure.

Example 5.3. In this example, we consider a directed graph $G = C_{n_1,\ldots,n_k}$ which is not the digraph corresponding to an undirected graph (as were examples (3.1) and (3.2)). This graph is a multipartite cyclic digraph defined by taking the composition of n to be $\mathbf{F} = (n_1, \ldots, n_k)$. Thus, the filtration is the partition $\mathcal{C}_i = \{N_{i-1}+1, \ldots, N_i\}$, with $N_0 = 0$ and $N_i = n_1 + \cdots + n_i$, $i = 1, \ldots, k$. There is one base, $\mathcal{C}_1$, and one summit, $\mathcal{C}_k$. All directed edges that connect a vertex in $\mathcal{C}_i$ to a vertex in $\mathcal{C}_{i+1}$ are present for $i = 1, \ldots, k-1$. As required by our framework (Theorem 2.4), all directed edges that connect a summit vertex in $\mathcal{C}_k$ to a base vertex in $\mathcal{C}_1$ are also present. Thus in this case, the t as in the statement of Theorem 2.4 is just 1. Thus all roots $\{1, \ldots, m\}$ of forests in $\mathcal{T}^G_{[m]}$ will belong to $\mathcal{C}_1$ since all such roots belong to a base and $\mathcal{C}_1$ is the only base.

As with equations (8) and (22), we start with identity (5) of Theorem 2.4:

$$\sum_{T \in \mathcal{T}^G_{[m]}} W(T) = \tag{56}$$
$$q_n(t_1 + \cdots + t_{N_{t-1}}) \sum_{f \in \mathcal{F}_n(G,\mathbf{F},m)} W(f) \;+\; q_n(t_{N_{t-1}+1} + \cdots + t_m) \sum_{f \in \mathcal{F}^*_n(G,\mathbf{F},m)} W(f).$$

In this case, $N_{t-1} = 0$ so we obtain the simplification

$$q_n(t_1 + \cdots + t_m) \sum_{f \in \mathcal{F}^*_n(G,\mathbf{F},m)} W(f) = \sum_{T \in \mathcal{T}^G_{[m]}} W(T). \tag{57}$$

Translating the sum over $f \in \mathcal{F}^*_n(G, \mathbf{F}, m)$ directly into generating functions we obtain

$$q_n(t_1 + \cdots + t_m) \tilde{D}_1 \prod_{j=2}^{k} \tilde{A}_j = \sum_{T \in \mathcal{T}^G_{[m]}} W(T). \tag{58}$$

Here for $t < j < k$,

$$\tilde{A}_j = \prod_{i=N_{j-1}+1}^{N_j} p_i(s_{N_j+1} + \cdots + s_{N_{j+1}}).$$

The summation in the above expression does not take into account fixed points in functions $f \in \mathcal{F}_n(G, \mathbf{F}, m)$. No such fixed points can occur since any such point must belong to both a base and a summit and there are no blocks in the filtration that are both a base and a summit. In addition, we have

$$\tilde{D}_1 = \prod_{i=m+1}^{N_1} p_i(s_{N_1+1} + \cdots + s_{N_2}) \qquad \text{and} \qquad \tilde{A}_k = \prod_{i=N_{k-1}+1}^{n-1} q_i(t_1 + \cdots t_{N_1}).$$

If we set $q_i = t_i = p+i = s_i = 1$ for all i, then combining the products in the above expressions, we obtain

$$|\mathcal{T}^G_{[m]}| = m \cdot (n_2)^{n_1 - m} \cdot (n_1)^{n_k - 1} \prod_{j=2}^{k-1} (n_{j+1})^{n_j}. \tag{59}$$

It is easy to give explicit formulas for the ranking and unranking of forests in $\mathcal{T}^G_{[m]}$ in this case since there are no complications due to fixed points. As an example, we give the unranking procedure. In this case it is easy to check that $\mathcal{D}_1 = \{t+1, \ldots, n_1\}$, $\mathcal{D}_h = \{N_{h-1}+1, \ldots, N_{h-1}+n_h\}$ for $h = 2, \ldots, k-1$ and

$\mathcal{D}_k = \{N_{k-1}+1,\ldots,N_{k-1}+n_k-1\}$. Moreover it is easy to check from our definitions of $T^{j,*}_{\mathcal{D}_j}$ that for $j=1,\ldots,t$,

$$\begin{aligned} L_1 &= L(T^{j,*}_{\mathcal{D}_1}) = (n_2)^{n_1-t} && (60)\\ L_h &= L(T^{j,*}_{\mathcal{D}_h}) = (n_{h+1})^{n_h} \text{ for } h=2,\ldots,k-1 && (61)\\ L_k &= L(T^{j,*}_{\mathcal{D}_k}) = (n_1)^{n_k-1} && (62)\end{aligned}$$

One can easily see from Figure 4 that the number of leaves of $\mathcal{T}$ is given by

$$L(\mathcal{T}) = m\cdot L_1\cdots L_k. \tag{63}$$

Unranking Procedure for $\mathcal{T}^G_{[m]}$ where $G = C_{n_1,\ldots,n_k}$ and $m < n_1$.

Goal: Find the forest $F \in \mathcal{T}^G_{[m]}$ such that $rank_{\mathcal{T}^G_{[m]}}(F) = a_0$.

Step 1. Find $r \le m$ such that
$$(r-1)L_1\cdots L_k \le a_0 < rL_1\cdots L_k.$$
This means that F will be of the form $\Theta^*_r(f)$ for some function $f \in \mathcal{F}^*_n(G,\mathbf{F},m)$ where

$$rank_{\mathcal{T}_r}(f) = b_0 = a_0 - (r-1)L_1\cdots L_k. \tag{64}$$

Step 2.
Goal: Find $p_1\otimes\cdots\otimes p_k \in T^{r,*}_{\mathcal{D}_1}\otimes\cdots\otimes T^{r,*}_{\mathcal{D}_k}$ such that $rank_{\mathcal{T}_r}(p_1\otimes\cdots\otimes p_k) = b_0$.

Find $c_1,\ldots,c_k$ such that

$$\begin{aligned} b_0 &= c_1\cdot L_2\cdots L_k + b_1 \text{ where } 0\le b_1 < L_2\cdots L_k\\ b_1 &= c_2\cdot L_3\cdots L_k + b_2 \text{ where } 0\le b_2 < L_3\cdots L_k\\ &\vdots\\ b_{k-2} &= c_{k-1}\cdot L_k + b_{k-1} \text{ where } 0\le b_k < L_k\\ b_{k-1} &= c_k.\end{aligned}$$

Then $rank_{T_{\mathcal{D}_j}}(p_j) = c_j$.

Step 3.
Goal: Find f_j equals f restricted to $\mathcal{D}_j$ such that $rank_{T^{r,*}_{\mathcal{D}_j}}(f_j) = rank_{T^{r,*}_{\mathcal{D}_j}}(f_j) = c_j$ for $j=1,\ldots,k$.

Step 3: Case 1.
Set $d_{0,1}=c_1$ and compute $e_{s,1}$ for $s=1,\ldots n_1-m$ where

$$\begin{aligned} d_{0,1} &= e_{1,1}\cdot n_2^{n_1-m-1} + d_{1,1} \text{ where } 0\le d_{1,1} < n_2^{n_1-m-1}\\ d_{1,1} &= e_{2,1}\cdot n_2^{n_1-m-2} + d_{2,1} \text{ where } 0\le d_{2,1} < n_2^{n_1-m-2}\\ &\vdots\\ d_{n_1-m-2,1} &= e_{n_1-m-1,1}\cdot n_2 + d_{n_1-m-1,1} \text{ where } 0\le d_{n_1-m-1,1} < n_2\\ d_{n_1-m-1,1} &= e_{n_1-m,1}.\end{aligned}$$

Then for $s=1,\ldots,n_1-m$,
$$f(m+s) = n_1+1+e_{s,1}. \tag{65}$$

Step 3: Case 2. For $j = 2, \ldots, k-1$, do the following.
Set $d_{0,j} = c_j$ and compute $e_{s,j}$ for $s = 1, \ldots n_j$ where

$$\begin{aligned}
d_{0,j} &= e_{1,j} \cdot n_{j+1}^{n_j - 1} + d_{1,j} \text{ where } 0 \le d_{1,j} < n_{j+1}^{n_j-1} \\
d_{1,j} &= e_{2,j} \cdot n_{j+1}^{n_j - 2} + d_{2,j} \text{ where } 0 \le d_{2,j} < n_{j+1}^{n_j-2} \\
&\vdots \\
d_{n_j-2,j} &= e_{n_j-1,j} \cdot n_{j+1} + d_{n_j-1,j} \text{ where } 0 \le d_{n_j-1,j} < n_{j+1} \\
d_{n_j-1,j} &= e_{n_j,j}.
\end{aligned}$$

Then for $s = 1, \ldots, n_j$,

$$f(N_{j-1} + s) = N_j + 1 + e_{s,j}. \tag{66}$$

Step 3: Case 3.
Set $d_{0,k} = c_k$ and compute $e_{s,k}$ for $s = 1, \ldots n_k - 1$ where

$$\begin{aligned}
d_{0,k} &= e_{1,k} \cdot n_1^{n_k - 2} + d_{1,k} \text{ where } 0 \le d_{1,k} < n_1^{n_k-2} \\
d_{1,k} &= e_{2,k} \cdot n_1^{n_k - 3} + d_{2,k} \text{ where } 0 \le d_{2,k} < n_1^{n_k-3} \\
&\vdots \\
d_{n_k-3,k} &= e_{n_k-2,k} \cdot n_1 + d_{n_k-2,k} \text{ where } 0 \le d_{n_k-2,k} < n_1 \\
d_{n_1-2,k} &= e_{n_k-1,k}.
\end{aligned}$$

Then for $s = 1, \ldots, n_k - 1$,

$$f(N_{k-1} + s) = 1 + e_{s,k}. \tag{67}$$

Step 4. Having constructed $f \in \mathcal{F}_n^*(G, \mathbf{F}, m)$ in Step 3, then $F = \Theta_r^*(f)$.

Finally we end this section with a brief discussion of how one can generate additional examples of filtered digraphs by modifying and combining examples 5.2 and 5.3 in various ways.

First, we should note that Example 5.3 is the simplest case of a more general construction. That is, suppose that we start with any acylic directed graph $G = ([k], E)$ such that vertex 1 is the only source in G, vertex k is the only sink in G, and all vertices lie on a path from the source to the sink. Then we can create a filtered digraph as follows. Start with a composition $\mathbf{F} = (c_1, c_2, \ldots, c_k)$ of n and let $\mathcal{C}_1, \ldots, \mathcal{C}_k$ be its corresponding set partition. Next we replace each vertex i by an empty graph $\mathcal{C}_i$. Then whenever there is an edge from $(i, j) \in E$, we create a directed edge from each vertex in $\mathcal{C}_i$ to each vertex in $\mathcal{C}_j$. We let $\mathcal{C}_1$ be the only base and let $\mathcal{C}_k$ be the only summit. Finally, we must add a directed edge from each summit vertex to each base vertex. Clearly, the graphs $C_{n_1,\ldots,n_k}$ arise from this construction by starting with a simple graph

$$1 \to 2 \to \cdots \to k-1 \to k.$$

We can also combine Examples 3.2 and 3.3 in various ways. For example, suppose that we start with blocks of sizes $n_1, \ldots, n_k$ as in Example 3.2. Now instead of taking the empty graph in each of these blocks as we do to create $K_{n_1,\ldots,n_k}$, imagine that we put the structure of a multipartite cyclic digraph in each block. That is, suppose that in each block $\mathcal{C}_i = \{N_{i-1}+1, \ldots, N_i\}$, we put the structure of some $C_{p_1,\ldots,p_l}$ where $p_1 + \cdots + p_l = n_i$. Then there are many possible ways to connect the blocks $\mathcal{C}_1, \ldots, \mathcal{C}_k$ to form a filtered digraph. Clearly we have to connect each vertex in a summit of a $C_{p_1,\ldots,p_l}$ that makes up a block $\mathcal{C}_i$ to all vertices in bases in a block $\mathcal{C}_s$ with $s < i$. However, we have many choices of how to connect the element of $\mathcal{C}_i$ to elements of $\mathcal{C}_t$ with $i < t$ and still conform to Definition 2.2. For example, we could have a directed edge from each vertex of $\mathcal{C}_i$ to every vertex of $\mathcal{C}_t$. We could connect every vertex of the base of $\mathcal{C}_i$ to every vertex $\mathcal{C}_t$ and make no other connections. We could connect every vertex of the summit of $\mathcal{C}_i$ to every vertex $\mathcal{C}_t$ and make no other connections. We could connect every vertex of the base of $\mathcal{C}_i$ to every vertex in the base of $\mathcal{C}_t$ and make no other connections, etc.. Of course, we can iterate this type of procedure. For example, replace the empty graph on $\mathcal{C}_i$ by one of the filtered digraphs that we just described.

References

[1] C.W. Borchardt, *Ueber eine der Interpolation entsprechende Darstellung der Eliminations-Resultante*, J. reine angew. Math. **57** (1860), pp. 111-121.

[2] C.J. Colborn, R.P.J. Day, and J.D. Nel, *Unranking and Ranking Spanning Trees of a Graph*, J. of Algorithms, **10**, (1989), pp. 271-286.

[3] Ömer Eğecioğlu and Jeffrey B. Remmel, *Bijections for Cayley Trees, Spanning Trees, and their q-Analogues*, Journal of Combinatorial Theory, Series A, Vol. 42. No. 1 (1986), pp. 15-30.

[4] Ömer Eğecioğlu and Jeffrey B. Remmel, *A bijection for spanning trees of complete multipartite graphs*, Congress Numerautum **100** (1994), pp. 225-243.

[5] Ömer Eğecioğlu and Jeffrey B. Remmel, *Ranking and Unranking Spanning Trees of Complete Multipartite Graphs*, Preprint.

[6] Ö. Eğecioğlu and L.P. Shen, *A Bijective Proof for the Number of Labeled q-trees*, Ars Combinatoria **25B** (1988), pp. 3-30.

[7] R. Onodera, *Number of trees in the complete N-partite graph*, RAAG Res. Notes **3**, No. 192 (1973), pp. i +6.

[8] H. Prufer, *Never Bewies eines Satzes über Permutationen*, Arch. Math. Phys. Sci. **27** (1918), pp. 742-744.

[9] A. Nijenhaus adn H.S. Wilf, *Combinatorial Algorithms*, 2nd. ed., Academic Press, New York, (1978).

[10] E.M. Reingold, J. Neivergelt, and N. Deo, *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, N.J., (1977)

[11] J.B. Remmel and S.G. Williamson, *Spanning Trees and Function Classes*, Electronic Journal of Combinatorics, (2002), **R34**.

[12] S.G. Williamson, *Combinatorics for Computer Science*, Dover Publications, Inc., Meneola, New York (2002).