THE MOVE-TO-ROOT RULE FOR SELF-ORGANIZING TREES WITH MARKOV DEPENDENT REQUESTS

by Robert P. Dobrow[1]

National Institute of Standards and Technology

ABSTRACT

The move-to-root (MTR) heuristic is a self-organizing rule which attempts to keep a binary search tree in near-optimal form. It is a tree analogue of the well-studied move-to-front (MTF) scheme. We study a Markov move-to-root (MMTR) model, where the sequence of record requests is a Markov chain, and analyze several characteristics of the tree chain, including the stationary distribution, eigenvalues, and stationary expected search cost.

INTRODUCTION AND SUMMARY

Self-organizing linear search heuristics are adaptive algorithms that seek to dynamically reorder elements in a list in order to increase search efficiency. See [8] for a gentle introduction.

The most well-studied self-organizing rule is the move-to-front (MTF) scheme. A collection of n items is stored in a sequential list. Associated with the ith record is a probability (weight) r_i measuring the long-run frequency of its use. At each unit of time, item i is removed from the list with probability r_i and replaced at the front of the list. This gives a Markov chain on the permutation group S_n.

In this paper we consider an extension of this model to binary search trees. Allen and Munro [1] introduce a natural analogue of MTF for binary search trees—the move-to-root (MTR) rule, described in Section 2. They also give an exact formula for stationary expected search cost (the asymptotic average cost of retrieving a record). Dobrow and Fill [4] show that the MTR

[1] Research supported in part by NSF grant DMS-9311367 while the author was a graduate student at The Johns Hopkins University.

Markov chain can be derived by lumping the MTF chain and derive numerous characteristics of the tree chain, including spectral structure and stationary distribution. Dobrow and Fill [5] treat rates of convergence to stationarity. Other treatments of self-organizing trees include Bitner [2], who considers various search rules, and Sleator and Tarjan [15], who introduce splay trees and develop (non-probabilistic) amortized analysis of search cost. We also note that there is a large literature on random search trees, much of which is referenced in [11].

In most of the probability-based literature on self-organizing data structures it is assumed that records are requested independently of all other requests. This assumption is often unrealistic. Requests may exhibit what in the computer science literature is frequently called "locality of reference." That is, there is often dependence between successive record requests. Knuth [9] cites computational experiments involving compiler symbol tables and notes that typically "successive searches are not independent (small groups of [records] tend to occur in bunches)."

Lam et al. [10] formally set up the Markovian model for self-organizing linear search where the sequence of record requests is a Markov chain. Other work along these lines includes [3], [13], [6], and [14].

In this paper we treat self-organizing binary search trees where the sequence of record requests is a Markov chain. We analyze several characteristics of the tree chain for this *Markov move-to-root* (MMTR) model. We are aware of no work to date which has considered self-organizing trees with dependent structure for the sequence of requests.

This paper is organized as follows: In Section 2 we explain the move-to-root rule, set up the Markovian model for requests, and establish some preliminary results, including the fundamental connection between Markov move-to-front (MMTF) for lists and MMTR for trees. In Section 3 we consider specific models for the request chain and give corresponding formulas for the stationary distribution of MMTR. In Section 4 we treat stationary expected search cost. In Section 5 the eigenvalues and their multiplicities are derived for the MMTR transition matrix.

PRELIMINARIES AND LUMPING

We will review background material briefly. For more details, see [4], [5].

Consider an ordered, indexed set of n records. For ease of notation and exposition we identify the records with their indices and simply consider the records $[n] := \{1, 2, \ldots, n\}$.

A *binary tree* is a finite tree with at most two "children" for each node and in which each child is distinguished as either a left or right child. By defining an empty binary tree as a binary tree with no nodes we can give a useful recursive definition: a binary tree is either empty or is a node with left and right subtrees, each of which is a binary tree.

Consider a binary tree in which the nodes are labeled with elements of some linearly ordered set. A *binary search tree* is a binary tree with the property that for each node x, all the records in the left subtree of x are less than the record stored at x, and all the records in the right subtree of x are greater than the record stored at x.

Let B_n be the set of all labeled binary search trees on n nodes. In what follows we use the term "tree" for binary search tree.

The *move-to-root* (MTR) operation is defined as a series of simple exchanges between nodes. A *simple exchange* (SE) for a requested record j is best understood by examining Figure 1. The MTR operation performs a sequence of simple exchanges until the requested record is moved to the root of the tree.

Let $\sigma \in S_n$ represent an ordered list of records, with σ_k denoting the record at the kth position of σ. Let $r_1, \ldots, r_n$ be a sequence of probabilities (weights) with the interpretation that record i is requested with long-run frequency r_i.

For $T \in B_n$, let $x \in T$ be a node in T. We will sometimes abuse notation and let $x \in T$ also refer to the record stored at x; the meaning should be clear from the context. If record j is stored at node x, then $r_x := r_j$.

Let R by the $n \times n$ transition matrix for the request chain. Thus $R(i,j)$ is the probability of accessing record j given that the previous request was for record i. In the case of independent requests the rows of R are identical and equal to $(r_1, \ldots, r_n)$. We will denote such a request matrix by R_I and refer to MMTR with such a request matrix as the *i.i.d. case* or as MTR.

For $T \in B_n$, let $\mathrm{rt}(T)$ denote the record stored at the root of T. Let Q be the transition matrix for MMTR. Then for $S, T \in B_n$,

$$Q(S,T) = \begin{cases} R(\mathrm{rt}(S), \mathrm{rt}(T)), & \text{if } T \text{ can be obtained from } S \\ & \quad \text{by a move-to-root operation} \\ 0, & \text{otherwise.} \end{cases}$$

In all that follows we require that the R-chain be ergodic, which implies that its stationary distribution is unique and strictly positive. Note that this does *not* imply that the tree chain is ergodic.

For $T \in B_n$ and $i \neq j$, we say that i is an *ancestor* of j in T, and write $i <_a^T j$, if j is an element of the subtree which has i as its root. A tree is uniquely determined by its ancestry relations.

A key observation for analyzing MTR is Lemma 3.2 in [1], which we reproduce, with a slight extension:

Lemma 0.1 *Suppose record i has been requested at least once in a tree modified according to the MTR rule. If $i < j$, then $i <_a j$ if and only if the most* *recent request for i has occurred since the most recent request (if any) for any of $i+1,\ldots,j$. Similarly, if $i>j$, then $i <_a j$ if and only if the most recent request for i has occurred since the most recent request for any of $j,\ldots,i-1$.*

Consider the operation of inserting records into an initially empty tree. This defines a mapping $t : S_n \to B_n$ where for $\sigma \in S_n$, $t(\sigma)$ is the tree obtained by successively inserting records $\sigma_1,\ldots,\sigma_n$ into an empty tree. Dobrow and Fill [4] show that the MTR chain can be obtained by lumping the MTF chain with respect to the mapping t. It is easy to modify their argument to show that the MMTR chain (for any R) can be obtained by lumping the MMTF chain (for the same R). This gives

Theorem 1 *Let Q denote the transition matrix for MMTR and P the transition matrix for MMTF. Let $\Pi(T)$ denote the set of permutations that are mapped to a given tree T by t. Then for $S,T \in B_n$ and $k \geq 0$,*

$$Q^k(S,T) = \sum_{\sigma\in\Pi(T)} P^k(\pi,\sigma) \quad \textit{for all } \pi \in \Pi(S). \tag{1}$$

Corollary 0.1 *Let Q^∞ denote the stationary distribution for MMTR and P^∞ the stationary distribution for MMTF. Then for $T \in B_n$,*

$$Q^\infty(T) = \sum_{\sigma\in\Pi(T)} P^\infty(\sigma). \tag{2}$$

MODELS AND STATIONARY DISTRIBUTION

Models

Throughout this paper we will consider two general models for the request chain.

For a fixed probability distribution $\mathbf{p} = (p_1,\ldots,p_n)$ on $[n]$, with $p_i > 0$ for all i, and a vector $(c_1,\ldots,c_n)$, where $0 \leq c_i < 1$ for each i, define

$$R_P(i,j) := \begin{cases} c_i, & \text{if } i=j \\ (1-c_i)p_j/(1-p_i), & \text{if } i \neq j. \end{cases}$$

For $i \neq j$, $R_P(i,j)$ is a product of two functions, one depending only on i and the other only on j. We will refer to MMTR with this request matrix as the *product case.*

The case when $c_i = 0$ for all i (and implicitly the more general R_P) was introduced in [14]. For $0 \leq \alpha < 1$, taking $c_i = (1-\alpha)p_i + \alpha$ gives the *mixture case*

$$R_M := (1-\alpha)R_I + \alpha I_n,$$

where I_n is the $n \times n$ identity matrix and R_I is the request matrix for the i.i.d. case introduced in Section 2. Taking $\alpha = 0$ gives the i.i.d. case.

A generalization of the mixture case which seems to capture locality of reference reasonably well is a mixture of the i.i.d. chain and a birth-and-death chain, that is,

$$R_B := (1-\alpha)R_I + \alpha B, \tag{3}$$

where B is a birth-and-death transition matrix. Unfortunately, MMTF with this request matrix seems hard to analyze cleanly. We shall, however, treat the extreme case when $\alpha = 1$ in (3), that is the *birth-and-death case* $R_B = B$.

For $A \in \{B,\ I,\ M,\ P\}$, let Q_A denote the transition matrix for MMTR with request matrix R_A. Let Q_A^∞ denote the corresponding stationary distribution. Further, let P_A and P_A^∞ denote the transition matrix and stationary distribution for MMTF, respectively, when the request matrix is R_A.

Stationary distribution

(a) *Product case.* Dobrow and Fill [4] give a tree-based description of the stationary distribution for MTR which we record for completeness.

Theorem 2 *For $T \in B_n$,*

$$Q_I^\infty(T) = \prod_{x \in T} \left(\frac{r_x}{\sum_{y \in T_x} r_y} \right), \tag{4}$$

where T_x is the subtree of T with root x.

There is a clear advantage in using Theorem 2, as opposed to the lumping formula in Corollary 0.1, to compute the stationary distribution of MTR. Indeed, (4) can be computed in linear time, while the number of summands in (2) is exponentially large for many T.

More generally, consider MMTF when R_P is the request matrix. For $\sigma \in S_n$, a slight extension of the argument in [14] gives

$$P_P^\infty(\sigma) = r_{\sigma_1} \prod_{i=2}^{n} \left(\frac{p_{\sigma_i}}{\sum_{j=i}^{n} p_{\sigma_j}} \right),$$

where $\mathbf{r} = (r_1, \ldots, r_n)$ is the stationary distribution of R_P. One can verify that R_P is reversible with

$$r_i = \frac{p_i(1-p_i)}{S(1-c_i)},$$

where

$$S := \sum_{i=1}^{n} \frac{p_i(1-p_i)}{1-c_i}. \tag{5}$$

An obvious modification of the proof of Theorem 2 in [4] gives a tree-based description of the corresponding stationary distribution for MMTR.

Theorem 3 *For $T \in B_n$,*

$$Q_P^\infty(T) = r_{\mathrm{rt(T)}} \prod_{\substack{x \in T: \\ x \neq \mathrm{rt}(T)}} \left(\frac{p_x}{\sum_{y \in T_x} p_y} \right).$$

(b) *Birth-and-death case.* For this case, write

$$R_B(i,j) = \begin{cases} q_i, & \text{if } j = i-1 \\ 1 - q_i - p_i, & \text{if } j = i \\ p_i, & \text{if } j = i+1 \\ 0, & \text{otherwise} \end{cases}$$

for $1 \leq i, j \leq n$, with $q_1 = p_n = 0$. Here the MMTR chain is not ergodic on B_n. It follows from results in [6] that $Q_B^\infty(T) > 0$ if and only if there exists $\sigma \in \Pi(T)$ such that $(\sigma_2, \ldots, \sigma_n)$ is an interleaving of the two sequences $(\sigma_1 - 1, \sigma_1 - 2, \ldots, 1)$ and $(\sigma_1 + 1, \sigma_1 + 2, \ldots, n)$. If σ is such an interleaving then the tree $t(\sigma)$ is of the form shown in Figure 2. Thus the support of Q_B^∞ are the n trees of this form. One sees easily that the tree chain is essentially the request chain itself. If T^i is the tree shown in Figure 2 with i at the root then

$$Q_B^\infty(T^i) = r_i.$$

(c) *General R.* For completeness we conclude this section by giving an expression for the stationary distribution of MMTR for general request matrix R, using (2) and Theorem 2 in [6]. Before so doing we establish some notation which will also be used in Section 5 to derive the eigenvalues of MMTR.

For a vector σ of length at least m, let $\sigma_{\to m}$ denote the m-element vector consisting of the first m elements of σ. Notation such as $y_{t_{\to m}}$ is shorthand for $(y_{t_1}, \ldots, y_{t_m})$. We write $\sigma[m]$ to denote the unordered set $\{\sigma_1, \ldots, \sigma_m\}$. Let $\widetilde{R}$ be the transition matrix of the time-reversed request chain; that is, $\widetilde{R}(x,y) := r_y R(y,x)/r_x$. Let $\widetilde{X} = (\widetilde{X}_k)_{k \geq 0}$ be a Markov chain with transition matrix $\widetilde{R}$. We use the notation $P_x(\cdot)$ for conditional probability given that $\widetilde{X}_0 = x$ and $P_{\mathbf{r}}(\cdot)$ for probability with respect to the $\widetilde{X}$-chain started in its stationary distribution $\mathbf{r}$. Let $\widetilde{C}_m$ be the first time at which $\widetilde{X}$ has visited m distinct states. Formally,

$$\widetilde{C}_m := \inf\{k \geq 0 : |\widetilde{X}[k]| = m\}.$$

Theorem 4 *For $T \in B_n$,*

$$Q^\infty(T) = \sum_{\sigma \in \Pi(T)} P_{\mathbf{r}}[\widetilde{X}_{\widetilde{C}_{\to n}} = \sigma]. \tag{6}$$

This has a straightforward probabilistic interpretation: For $T \in B_n$, $Q^\infty(T)$ is the probability of choosing n records by "Markov coupon-collecting" so that the tree T is formed by inserting these records into an empty tree. However, there does not appear to be a direct, tree-based formula for $Q^\infty(T)$ that would simplify computation.

STATIONARY EXPECTED SEARCH COST

Our main result in this section is Theorem 5, which gives an expression for the stationary expected search cost (ESC) for MMTR with any request matrix R. For a given subset A of the state space, let H_A (respectively, H_A^+) be the first nonnegative (resp., positive) hitting time of A for the time-reversal $\widetilde{X}$ of the request chain.

Theorem 5

$$\mathrm{ESC} = 1 + \sum_{i,j\,:\,i<j} r_i P_i[H_j < H^+_{\{i,i+1,\ldots,j-1\}}] + \sum_{i,j\,:\,i>j} r_i P_i[H_j < H^+_{\{j+1,\ldots,i-1,i\}}]. \tag{7}$$

Proof The proof is similar to the approach taken by Lam et al. [10], who derive a formula for stationary expected search cost for MMTF. For $T \in B_n$ and $i \in [n]$, let $\mathrm{depth}_T(i)$ denote the depth of record i in T, where the depth of the root record is 1. Let $1(A)$ be the indicator of the event A. Let (T^*, i^*) be a random pair where T^* is a random tree distributed according to Q^∞ and i^* is the current (random) record requested, with

$$P[i^* = i \,|\, T^* = T] = R(\mathrm{rt}(T), i).$$

Stationary expected search cost is then given by

$$\mathrm{ESC} := E[\mathrm{depth}_{T^*}(i^*)] = \sum_{i=1}^n r_i E[\mathrm{depth}_{T^*}(i^*) \,|\, i^* = i].$$

But $\mathrm{depth}_T(i) = 1 + \sum_{j \neq i} 1(j <_a^T i)$ and thus

$$E[\mathrm{depth}_{T^*}(i^*) \,|\, i^* = i] = 1 + \sum_{j\,:\,j\neq i} P[j <_a^{T^*} i \,|\, i^* = i].$$

From Lemma 0.1,

$$\begin{aligned} &E[\mathrm{depth}_{T^*}(i^*) \,|\, i^* = i] \\ &\quad = \; 1 + \sum_{j\,:\,j\neq i} P[j <_a^{T^*} i \,|\, i^* = i] \\ &\quad = \; 1 + \sum_{j\,:\,i<j} P_i[H_j < H^+_{\{i,i+1,\ldots,j-1\}}] + \sum_{j\,:\,i>j} P_i[H_j < H^+_{\{j+1,\ldots,i-1,i\}}], \end{aligned}$$

and the result follows. ■

The following gives simple expressions for stationary expected search cost for our two models.

Corollary 0.2 *(a) In the product case,*

$$\mathrm{ESC}_P = 1 + \frac{2}{S} \sum_{i,j\,:\,i<j} \frac{p_i p_j}{p_i + \cdots + p_j}, \tag{8}$$

where $S = \sum_{i=1}^n \frac{p_i(1-p_i)}{1-c_i}$ *is as defined at (5).*

(b) For the birth-and-death case,

$$\mathrm{ESC}_B = 2 - \sum_{i=1}^n r_i(1 - p_i - q_i). \tag{9}$$

Proof (a) For fixed $i < j$, let

$$f(k) := P_k[H_j < H^+_{\{i,\ldots,j-1\}}].$$

Clearly, for $k \notin \{i, \ldots, j\}$

$$f(k) = \frac{p_j}{p_i + \cdots + p_j},$$

and $f(j) = 1$. For $k \in \{i, \ldots, j-1\}$, we thus have

$$\begin{aligned} f(k) &= R(k,j) + \sum_{l \notin \{i,\ldots,j\}} R(k,l) f(l) \\ &= \frac{(1-c_k)p_j}{1-p_k} + \sum_{l \notin \{i,\ldots,j\}} \frac{(1-c_k)p_l p_j}{(1-p_k)(p_i + \cdots + p_j)} \\ &= \frac{(1-c_k)p_j}{(1-p_k)(p_i + \cdots + p_j)}. \end{aligned}$$

In particular,

$$f(i) = \frac{(1-c_i)p_j}{(1-p_i)(p_i + \cdots + p_j)}.$$

Thus by Theorem 5,

$$\begin{aligned} \mathrm{ESC}_P &= 1 + \sum_{i,j\,:\,i<j} r_i \frac{(1-c_i)p_j}{(1-p_i)(p_i + \cdots + p_j)} + \sum_{i,j\,:\,i>j} r_i \frac{(1-c_i)p_j}{(1-p_i)(p_j + \cdots + p_i)} \\ &= 1 + \frac{2}{S} \sum_{i,j\,:\,i<j} \frac{p_i p_j}{p_i + \cdots + p_j}. \end{aligned}$$

(b) For the birth-and-death case, for $i < j$

$$P_i[H_j < H^+_{\{i,\ldots,j-1\}}] = \begin{cases} R_B(i,i+1) = p_i, & \text{if } j = i+1 \\ 0, & \text{otherwise,} \end{cases}$$

while for $i > j$,

$$P_i[H_j < H^+_{\{j+1,\ldots,i\}}] = \begin{cases} R_B(i,i-1) = q_i, & \text{if } j = i-1 \\ 0, & \text{otherwise.} \end{cases}$$

Thus

$$\mathrm{ESC}_B = 1 + \sum_{i=1}^{n-1} r_i p_i + \sum_{i=2}^{n} r_i q_i$$

giving (9).

■

Remarks:

1. Note that

$$\mathrm{ESC}_P - 1 = \frac{1}{S}(\mathrm{ESC}_I - 1),$$

where

$$\frac{1}{S} = \frac{\sum_{i=1}^n r_i(1-c_i)}{\sum_{i=1}^n p_i(1-p_i)}. \tag{10}$$

The righthand side of (10) is the ratio of chances (in stationarity) of obtaining differing successive requests, with the numerator calculated under MMTR and the denominator under MTR.

2. A natural question to ask is how MMTR fares with respect to stationary expected search cost in relation to the independent model. In the product case,

$$\mathrm{ESC}_P \left\{ \begin{array}{c} < \\ = \\ > \end{array} \right\} \mathrm{ESC}_I \text{ according as } S \left\{ \begin{array}{c} > \\ = \\ < \end{array} \right\} 1.$$

In the mixture case,

$$S = \sum_{i=1}^n \frac{p_i(1-p_i)}{(1-\alpha)(1-p_i)} = \frac{1}{1-\alpha} \geq 1.$$

Thus

$$\mathrm{ESC}_M \leq \mathrm{ESC}_I,$$

with equality if and only if $\alpha = 0$. For the case $c_i = 0$, $S = \sum_{i=1}^n p_i(1-p_i) < 1$, so $\mathrm{ESC}_P > \mathrm{ESC}_I$. The analogues of these results for self-organizing lists have been noted by various authors.

EIGENVALUES

Phatarfod [12] derived the eigenvalues and their multiplicities for MTF. Suppose for simplicity throughout this section that sums of distinct collections of weights r_i are distinct. Then the eigenvalues of MTF are all the partial sums of the weights, excluding those n cases where the summation is over $n-1$ weights. The multiplicity of each eigenvalue $\lambda_A = \sum_{j \in A} r_j$ corresponding to a sum of $|A| = m$ weights is the number of permutations in S_n fixing exactly those points in A, namely, the number of derangements (permutations with no fixed points) in S_{n-m}.

There is an interesting parallel between the spectral structure of MTR and MTF. Dobrow and Fill [4] define the notions of unit gap and fixed point of a tree and show (i) that the eigenvalues for MTR are the partial sums of weights excluding sets which have unit gaps, and (ii) that the multiplicity of the eigenvalue λ_A is the number of trees in B_n fixing exactly those points in A.

For $A \subseteq [n]$, write $a_1 < a_2 < \cdots < a_m$ for the elements of A. Define $a_0 := 0$ and $a_{m+1} := n+1$. Let

$$g_i(A) := a_{i+1} - a_i - 1, \quad i = 0, \ldots, m,$$

denote the number of integers in the interval (a_i, a_{i+1}). Then $g_i(A)$ is called the *i-th gap* of A.

We say that a tree $T \in B_n$ *fixes* a record j if there exists $\pi \in \Pi(T)$ such that $\pi_j = j$ and π maps $\{1, \ldots, j-1\}$ to itself and $\{j+1, \ldots, n\}$ to itself.

Phatarfod and Dyte [13] determined the eigenvalues and their multiplicities for MMTF. For a matrix M and set A, let $M_{(A)}$ denote the principal submatrix of M determined by the rows and columns identified by A. The set of eigenvalues of MMTF is the set of all eigenvalues of all the principal submatrices $R_{(A)}$ with $|A| \neq 0, n-1$ of the request chain R. For $A \subseteq [n]$, the eigenvalue $\lambda_i(A)$ corresponding to the ith eigenvalue of $R_{(A)}$ has multiplicity in P (the MMTF list transition matrix) equal to $D_{n-|A|}$, the number of derangements (permutations with no fixed points) of $n - |A|$ objects.

In light of these results the next theorem is not surprising.

Theorem 6 *The set of eigenvalues for the MMTR transition matrix Q is the set of all eigenvalues of all the principal submatrices $R_{(A)}$ of the request matrix R, where $A \neq \emptyset$ has no gaps of size 1. For such $A \subseteq [n]$, the eigenvalue $\lambda_i(A)$ corresponding to the ith eigenvalue of $R_{(A)}$ has multiplicity in Q equal to $\alpha_n(A)$, the number of trees which fix exactly those points in A.*

For $A \subseteq [n]$, the numbers $\alpha_n(A)$ can be computed directly. As shown in [4],

$$\alpha_n(A) = \prod_{i=0}^{|A|} \alpha_{g_i(A)},$$

where

$$\alpha_j = \frac{1}{2}\left[\left(-\frac{1}{2}\right)^j + \sum_{i=0}^{j}\left(-\frac{1}{2}\right)^i \beta_{j-i}\right], \quad j \geq 0,$$

with $\beta_k = \binom{2k}{k}/(k+1)$. An easy method to compute α_j is to use the initial values $\alpha_0 = 1$ and $\alpha_1 = 0$ together with the recurrence relation

$$2(j+1)\alpha_j = (7j-5)\alpha_{j-1} + 2(2j-1)\alpha_{j-2}, \quad j \geq 2.$$

For $T \in B_n$, recall that $\Pi(T)$ is the set of permutations $\sigma \in S_n$ such that $t(\sigma) = T$. For $m \in [n]$, let $\Pi_m(T)$ be the set of projections of the elements of $\Pi(T)$ onto their first m coordinates. For a vector v, let $[v]$ denote the unordered set formed by the elements of v.

Proof The proof follows the proof of part (a) of Theorem 4 in [6]. After computing the trace of Q^k, the result will follow by elementary linear algebra. For $\pi \in S_n$, let

$$L(\pi) = \max\{i \in [n-1] : \pi_i > \pi_{i+1}\}.$$

By Theorem 1 and Theorem 1 in [6], for any $\pi \in \Pi(T)$

$$\begin{aligned}
Q^k(T,T) &= \sum_{\sigma\in\Pi(T)} P^k(\pi,\sigma) \\
&= \sum_{\sigma\in\Pi(T)} \frac{1}{r_{\pi_1}} \sum_{m=L(\pi^{-1}\sigma)}^{n} P_{\mathbf{r}}[\widetilde{X}_{\widetilde{C}_{\to m}} = \sigma_{\to m},\ \widetilde{C}_m < k \leq \widetilde{C}_{m+1},\ \widetilde{X}_k = \pi_1] \\
&= \sum_{\sigma\in\Pi(T)} \frac{1}{r_{\sigma_1}} \sum_{m=L(\pi^{-1}\sigma)}^{n} P_{\mathbf{r}}[\widetilde{X}_{\widetilde{C}_{\to m}} = \sigma_{\to m},\ \widetilde{C}_m < k \leq \widetilde{C}_{m+1},\ \widetilde{X}_k = \sigma_1] \\
&= \sum_{\sigma\in\Pi(T)} \sum_{m=L(\pi^{-1}\sigma)}^{n} P_{\sigma_1}[\widetilde{X}_{\widetilde{C}_{\to m}} = \sigma_{\to m},\ \widetilde{C}_m < k \leq \widetilde{C}_{m+1},\ \widetilde{X}_k = \sigma_1],
\end{aligned}$$

since $\pi, \sigma \in \Pi(T)$ implies that $\pi_1 = \mathrm{rt}(T) = \sigma_1$. Note that we are using the convention that for $m = 0$,

$$P_{\sigma_1}[\widetilde{X}_{\widetilde{C}_{\to m}} = \sigma_{\to m},\ \widetilde{C}_m < k \leq \widetilde{C}_{m+1},\ \widetilde{X}_k = \sigma_1] = \begin{cases} 1, & \text{if } k = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Continuing,

$$\begin{aligned}
&Q^k(T,T) \\
&\quad = \sum_{m=0}^{n} \sum_{\sigma_{\to m}\in\Pi_m(T)} P_{\sigma_1}[\widetilde{X}_{\widetilde{C}_{\to m}} = \sigma_{\to m},\ \widetilde{C}_m < k \leq \widetilde{C}_{m+1},\ \widetilde{X}_k = \sigma_1] \\
&\qquad\qquad \times |\{\sigma' \in \Pi(T) : \sigma'_{\to m} = \sigma_{\to m},\ L(\pi^{-1}\sigma') \leq m\}|.
\end{aligned}$$

But it is not hard to see that

$$|\{\sigma' \in \Pi(T) : \sigma'_{\to m} = \sigma_{\to m},\ L(\pi^{-1}\sigma') \leq m\}| = 1$$

for any $\pi \in \Pi(T)$ and $\sigma_{\to m} \in \Pi_m(T)$.

Thus, for $T \in B_n$,

$$Q^k(T,T) = \sum_{m=0}^{n} \sum_{\sigma_{\to m} \in \Pi_m(T)} P_{\sigma_1}[\widetilde{X}_{\widetilde{C}_{\to m}} = \sigma_{\to m},\ \widetilde{C}_m < k \le \widetilde{C}_{m+1},\ \widetilde{X}_k = \sigma_1]$$

and so, for $k \ge 0$,

$$\mathrm{tr}(Q^k) = \sum_{T \in B_n} \sum_{m=0}^{n} \sum_{\sigma_{\to m} \in \Pi_m(T)} P_{\sigma_1}[\widetilde{X}_{\widetilde{C}_{\to m}} = \sigma_{\to m},\ \widetilde{C}_m < k \le \widetilde{C}_{m+1},\ \widetilde{X}_k = \sigma_1].$$

For $k \ge 1$,

$$\mathrm{tr}(Q^k) = \sum_{m=1}^{n} \sum_{\sigma_{\to m}} \tau_n(\sigma_{\to m}) P_{\sigma_1}[\widetilde{X}_{\widetilde{C}_{\to m}} = \sigma_{\to m},\ \widetilde{C}_m < k \le \widetilde{C}_{m+1},\ \widetilde{X}_k = \sigma_1],$$

where, as argued in the proof of Theorem 5 in Dobrow and Fill (1994a),

$$\tau_n(\sigma_{\to m}) := |\{T \in B_n : \sigma_{\to m} \in \Pi_m(T)\}|$$

depends only on the unordered set $\sigma[m]$ and is in fact the number of trees that fix at least the points in $\sigma[m]$. Continuing,

$$\begin{aligned} \mathrm{tr}(Q^k) &= \sum_{m=1}^{n} \sum_{A \in \binom{[n]}{m}} \sum_{x \in A} \tau_n(A) P_x[[\widetilde{X}_{\widetilde{C}_{\to m}}] = A,\ \widetilde{C}_m < k \le \widetilde{C}_{m+1},\ \widetilde{X}_k = x] \\ &= \sum_{m=1}^{n} \sum_{A \in \binom{[n]}{m}} \sum_{x \in A} \tau_n(A) P_x[[\widetilde{X}_{\to k}] = A,\ \widetilde{X}_k = x], \end{aligned}$$

where the sum $\sum_{A \in \binom{[n]}{m}}$ is over all m-element subsets A of $[n]$. By inclusion-exclusion, for $x \in A$ we have

$$\begin{aligned} P_x[[\widetilde{X}_{\to k}] = A,\ \widetilde{X}_k = x] &= \sum_{B \subseteq A} (-1)^{|A|-|B|} P_x[[\widetilde{X}_{\to k}] \subseteq B,\ \widetilde{X}_k = x] \\ &= \sum_{B \subseteq A} (-1)^{|A|-|B|} 1(x \in B) \widetilde{R}^k_{(B)}(x,x). \end{aligned}$$

Therefore,

$$\begin{aligned} \mathrm{tr}(Q^k) &= \sum_{m=1}^{n} \sum_{A \in \binom{[n]}{m}} \tau_n(A) \sum_{B \subseteq A} (-1)^{|A|-|B|} \sum_{x \in B} \widetilde{R}^k_{(B)}(x,x) \\ &= \sum_{m=1}^{n} \sum_{A \in \binom{[n]}{m}} \tau_n(A) \sum_{B \subseteq A} (-1)^{|A|-|B|} \mathrm{tr}(\widetilde{R}^k_{(B)}) \\ &= \sum_{\emptyset \ne B \subseteq [n]} \mathrm{tr}(\widetilde{R}^k_{(B)}) \sum_{A \supseteq B} (-1)^{|A|-|B|} \tau_n(A) \\ &= \sum_{\emptyset \ne B \subseteq [n]} \mathrm{tr}(\widetilde{R}^k_{(B)}) \alpha_n(B), \qquad (11) \end{aligned}$$

where the last equality follows from Möbius inversion.

For $k = 0$,

$$\begin{aligned}
\mathrm{tr}(Q^0) &= \beta_n = \sum_{j=1}^{n} |\{T \in B_n : \mathrm{rt}(T) = j\}| \\
&= \sum_{A:|A|=1} \tau_n(A) = \sum_{A\neq\emptyset} \tau_n(A) \sum_{\emptyset\neq B\subseteq A} (-1)^{|A|-|B|}|B| \\
&= \sum_{\emptyset\neq B\subseteq[n]} |B| \sum_{A\supseteq B} (-1)^{|A|-|B|}\tau_n(A) = \sum_{\emptyset\neq B\subseteq[n]} \mathrm{tr}(\widetilde{R}^0_{(B)})\alpha_n(B)
\end{aligned}$$

Since $\mathrm{tr}(R^k_{(B)}) = \mathrm{tr}(\widetilde{R}^k_{(B)})$ the result follows. ■

ACKNOWLEDGEMENTS

This work is part of the author's Ph.D. dissertation. I would like to thank my Ph.D. advisor Prof. Jim Fill for his assistance. I am also grateful to Prof. Eliane R. Rodrigues for making available a preprint of her paper.

References

REFERENCES

[1] Allen, B. and Munro, I. (1978). Self-organizing binary search trees. *J. ACM* **25** 526–535.

[2] Bitner, J. R. (1979). Heuristics that dynamically organize data structures. *SIAM J. Comp.* **8** 82–110.

[3] Chassaing, P. (1993). Optimality of move-to-front for self-organizing data structures with locality of reference. *Ann. Appl. Prob.* **3** 1219–1240.

[4] Dobrow, R. P. and Fill, J. A. (1994a). On the Markov chain for the move-to-root rule for binary search trees. *Ann. Appl. Prob.*, to appear.

[5] Dobrow, R. P. and Fill, J. A. (1994b). Rates of convergence for a self-organizing scheme for binary search trees. *Ann. Appl. Prob.,* to appear.

[6] Dobrow, R. P. and Fill, J. A. (1994c). The move-to-front rule for self-organizing lists with Markov dependent requests. Technical Report #532, Department of Mathematical Sciences, The Johns Hopkins University.

[7] Fill, J. A. (1993). An exact formula for the move-to-front rule for self-organizing lists. Technical Report #529, Department of Mathematical Sciences, The Johns Hopkins University.

[8] Hendricks, W. J. (1989). *Self-organizing Markov Chains.* MITRE Corp., McLean, Va.

[9] Knuth, D. (1973). *The Art of Computer Programming.* Vol. III. Addison-Wesley, Reading, Mass.

[10] Lam, K., Leung, M.-Y., and Siu, M.-K. (1984). Self-organizing files with dependent accesses. *J. Appl. Prob.* **21** 343–359.

[11] Mahmoud, H. M. (1992). *Evolution of Random Search Trees.* John Wiley & Sons, Inc., New York.

[12] Phatarfod, R. M. (1991). On the matrix occurring in a linear search problem. *J. Appl. Prob.* **28** 336–346.

[13] Phatarfod, R. M. and Dyte, D. (1993). The linear search problem with Markov dependent requests. Preprint.

[14] Rodrigues, E. R. (1994). The performance of the move-to-front scheme under some particular forms of Markov requests. Preprint.

[15] Sleator, D. D. and Tarjan, R. E. (1985). Self-adjusting binary search trees. *J. ACM* **32** 652–686.