

Efficient computation of the iteration of functions

TSUNG-HSI TSAI

Institute of Statistical Science

Academia Sinica

Taipei 115

Taiwan

e-mail: chonghi@stat.sinica.edu.tw

Tel: (886)-2-2783-5611 ext 210

Fax: (886)-2-2783-1523

January 3, 2009

Abstract

Given a function f from $\{0, 1, \dots, N-1\}$ to $\{0, 1, \dots, N-1\}$, we prove that $f^m(x)$, the m th iterate of f at x , can be computed in time $O(\log N)$ for each natural number m and each x by using $O(N)$ information that is generated in a preprocessing procedure. Two types of optimal orbit decompositions of functional graphs are proposed for the preprocess. Both preprocesses require only linear time and linear space. Our decompositions minimize the number of recursions in the computation of $f^m(x)$ and solve some open problems in Tsaban (Discrete Applied Mathematics 155 (2007) 386–393).

Keywords: Fast forward property; Horton-Strahler number; Register functions

1 Introduction

Let $V = \{0, 1, \dots, N-1\}$ and $f : V \rightarrow V$. Use $f^m(x)$ to denote the m th iterate of f at $x \in V$. The problem we are interested in here is how to preprocess a function f so that the evaluation of $f^m(x)$ can be done efficiently for each natural number m and $x \in V$ by using the information generated in the preprocess. The property that the iteration of a function can be computed efficiently is called the *fast forward property*. Such a property is important in many cryptographic applications, in particular in the issue of pseudo-randomness [12, 15, 16].

The approach in the paper requires $O(N)$ space to store the information generated in the preprocess. Practically, this means that N cannot be too large. In contrast, N is very large in the context of pseudo-randomness. Still, the order N space is tolerable for the problem, since most random mappings have no shorter definition than specifying their value for all mappings. However, if space is not a main issue, the problem becomes quite easy, since the iterates are ultimately periodic and they could be tabulated in order N^2 space.

Tsaban [15, 16] investigated the problem for f being a permutation and f being an arbitrary function. The approach is to construct an “orbit decomposition” of a function f and derive a formula for

$f^m(x)$ from the orbit decomposition. The formula consists of one recursion and five mappings and these mappings are implemented as lookup tables. Then the complexity is measured by the number of times the recursion is called in the process of evaluating $f^m(x)$. The occasion of applying the recursion corresponds to the “descent” in the orbit decomposition. Thus, the efficiency of evaluating $f^m(x)$ by this approach relies essentially on the number of descents. Tsaban proposed a greedy orbit decomposition to lower the number of descents for practical use. The average number of descents with respect to the greedy orbit decomposition is $O(\log N)$ by experiments. However, this result is not optimal and the maximal number of descents is $O(\sqrt{N})$ in the worst case.

In this paper we follow the same approach as that of Tsaban, which reduces the problem to finding orbit decompositions that minimize the number of descents. However, different from Tsaban [16], the orbit decompositions that we propose are optimal. Namely, we propose two types of orbit decompositions that minimize the maximal number of descents and the average number of descents, respectively. Moreover, the numbers of descents of both types are $O(\log N)$ for all cases.

The optimal orbit decompositions are closely related to certain functionals on rooted trees. The procedures for constructing the orbit decompositions are the same, but the rules of orbit are different. The orbits of the decompositions of the first type are determined by the extended Horton-Strahler numbers (see [2] and a list of references in Section 3.2 for more information). The orbits of the second type are determined by the number of nodes of subtrees while the orbits in the greedy orbit decomposition [16] are determined essentially by the number of levels (or depth) of subtrees.

2 Preliminary

We recall some basic definitions and results in [15, 16].

Definition. The *orbit* of an element x in $U \subset V$ is the shortest tour $(x, f(x), f^2(x), \dots, f^k(x))$ such that either $f^{k+1}(x) = f^i(x)$ for some $0 \leq i \leq k$ or $f^{k+1}(x) \notin U$.

Definition. The sequence of orbits $C_0, C_1, \dots, C_{\ell-1}$ is an *orbit decomposition* of f if C_0 is an orbit in V and C_i is an orbit in $V - C_0 \cup \dots \cup C_{i-1}$ for $i > 0$.

Given an orbit decomposition of f

$$\underbrace{(b_0, b_1, \dots, b_{s_0-1})}_{C_0}, \underbrace{(b_{s_0}, \dots, b_{s_1-1})}_{C_1}, \dots, \underbrace{(b_{s_{\ell-2}}, \dots, b_{N-1})}_{C_{\ell-1}},$$

where $s_i = |C_0| + \dots + |C_i|$. Define two mappings

$$\sigma(x) = b_x \quad \text{and} \quad \theta(x) = \begin{cases} p_i & \text{if } x = s_i - 1, \\ \pi(x) & \text{otherwise,} \end{cases}$$

where p_i is the sequence such that $f(b_{s_i-1}) = b_{p_i}$ and π is the permutation with the cycle decomposition

$$(0, \dots, s_0 - 1)(s_0, \dots, s_1 - 1) \cdots (s_{\ell-2}, \dots, N - 1).$$

Then

$$f = \sigma \circ \theta \circ \sigma^{-1}.$$

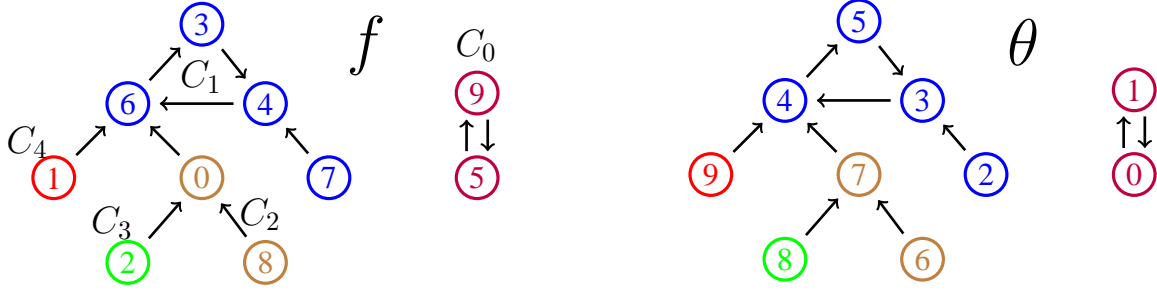


Figure 1: An example of an orbit decomposition of the function f , defined by the arrows of the directed graph on the left hand side, is given as follows: $C_0 = (5, 9)$, $C_1 = (7, 4, 6, 3)$, $C_2 = (8, 0)$, $C_3 = (2)$, $C_4 = (1)$. Then $\{\sigma(i)\}_{i=0}^9 = \{5, 9, 7, 4, 6, 3, 8, 0, 2, 1\}$, $\{p_i\}_{i=0}^4 = \{0, 3, 4, 7, 4\}$, $\pi = (0, 1)(2, 3, 4, 5)(6, 7)(8)(9)$ and θ is the function defined by the arrows of the directed graph on the right hand side. Also, the connections of C_2 to C_1 , C_4 to C_1 and C_3 to C_2 are descents. Thus $D_\theta(x) = 0$ for $x = 0, 1, 2, 3, 4, 5$, $D_\theta(x) = 1$ for $x = 6, 7, 9$ and $D_\theta(x) = 2$ for $x = 8$.

Note that the orbit $C_i = (b_{s_{i-1}}, \dots, b_{s_i-1})$ is connected to a prior orbit $C_j, j < i$, by the mapping $f(b_{s_{i-1}}) = b_{p_i}$ if $p_i < s_{i-1}$. We call this mapping, or the connection between two orbits, a *descent*. An example of an orbit decomposition of a function and the associated terms are given in Figure 1.

Now

$$f^m = \sigma \circ \theta^m \circ \sigma^{-1}.$$

By implementing σ and σ^{-1} as lookup tables, we only need to investigate the complexity of evaluating $\theta^m(x)$. Let $i(x)$ be the function such that $s_{i(x)} \leq x < s_{i(x)+1}$.

Case 1. If $s_{i(x)} \leq x + m < s_{i(x)+1}$, then

$$\theta^m(x) = x + m. \quad (1)$$

Case 2. If $x + m \geq s_{i(x)+1}$ and $p_{i(x)+1} \geq s_{i(x)}$, then

$$\theta^m(x) = p_{i(x)+1} + ((x + m - s_{i(x)+1}) \bmod (s_{i(x)+1} - p_{i(x)+1})). \quad (2)$$

Case 3. If $x + m \geq s_{i(x)+1}$ and $p_{i(x)+1} < s_{i(x)}$, then $\theta^m(x)$ is computed recursively by

$$\theta^m(x) = \theta^{x+m-s_{i(x)+1}}(p_{i(x)+1}). \quad (3)$$

Recursion (3) corresponds to a descent. By implementing the mappings $i \rightarrow p_i, i \rightarrow s_i$ and $x \rightarrow i(x)$ as lookup tables, we see that the complexity is measured by the number of descents on the tour $(x, \theta(x), \theta^2(x), \dots, \theta^m(x))$. $D_\theta(x)$ denotes the number of descents on the infinite tour $(x, \theta(x), \theta^2(x), \theta^3(x), \dots)$.

Theorem (Tsaban, 2007). The evaluation of $\theta^m(x)$ can be done in time $O(D_\theta(x) + 1)$ for all $x \in V$ by using (1), (2), (3) and the lookup tables mentioned above.

Nevertheless, $D_\theta(x)$ is not always small (i.e. $O(\log N)$). Tsaban proposed a greedy orbit decomposition for a solution.

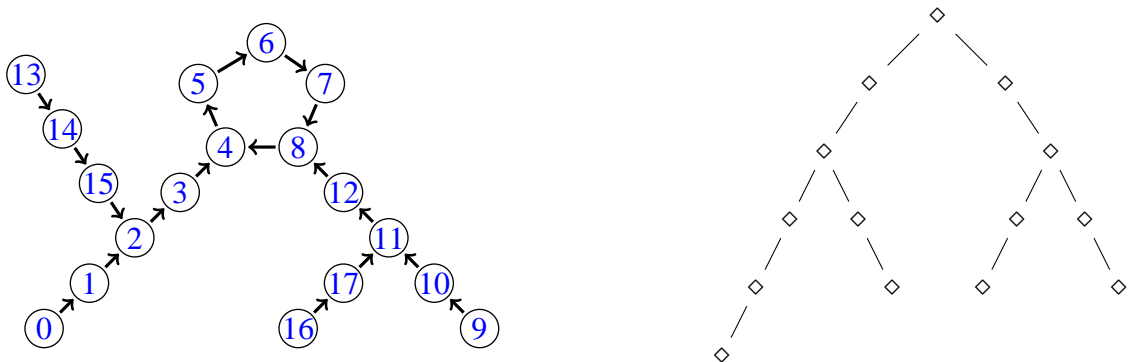


Figure 2: The functional graph on the left hand side can be regarded as the rooted tree on the right hand side by ignoring all labels and shrinking the cycle to a root.

Definition. The *greedy orbit decomposition* of f is $C_0, C_1, \dots, C_{\ell-1}$ constructed as follows. C_0 is a maximal length orbit in V and C_i is a maximal length orbit in $V - C_0 \cup \dots \cup C_{i-1}$ for $i > 0$.

A simulation by Tsaban indicates that the average (for random functions f and random points x) of $D_\theta(x)$ with respect to the greedy orbit decompositions is only about $(\log_2 N)/5$. However, Tsaban also proved that the worst case is about $\sqrt{2N}$.

3 New results

We present two types of orbit decompositions that minimize

$$\max_{x \in V} D_\theta(x) \text{ and } \sum_{x \in V} D_\theta(x), \quad (4)$$

respectively, in this section. The minimization implies optimal algorithms for computing $f^m(x)$ in the sense of worst and average performance.

3.1 Reduction of the problem to the orbit decomposition of a rooted tree

A functional graph (or random mapping) is a disjoint union of components of directed graphs with out-degree equal to one and each component containing exactly one cycle. Functional graphs are widely used and studied in discrete probability (see [4] and the references cited there). We simplify the problem from finding orbit decompositions of a functional graph to finding orbit decompositions of a rooted tree.

We consider the functional graphs without labels since they are unrelated to the values in (4). The components are orbit independent. The first orbit in a component must complete the cycle and terminate on the cycle. Thus the cycle can be regarded as a root and each component can be regarded as a rooted tree (see an example in Figure 2).

In the procedure of constructing an orbit decomposition, for each component an orbit, a path from a node to the root, is chosen and called *chief orbit* and then is eliminated from the component. The remainder is again a disjoint union of rooted trees (see an example in Figure 3). The procedure continues recursively until all remainders are empty.

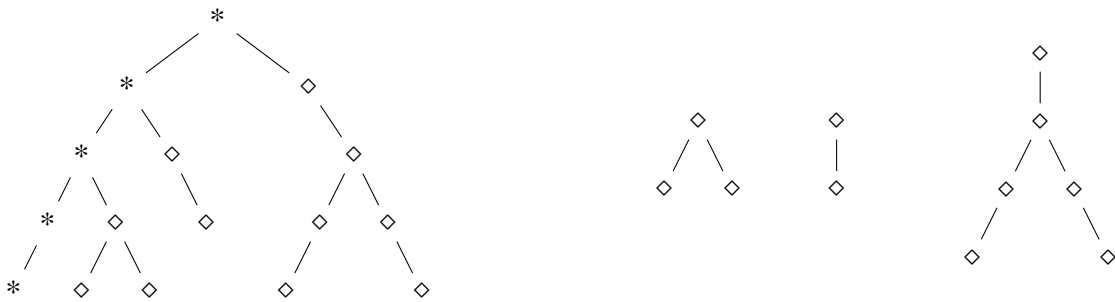


Figure 3: Eliminate the chief orbit (marked by $*$) from the rooted tree on the left hand side and then the remainder is the three disjoint rooted trees on the right hand side.

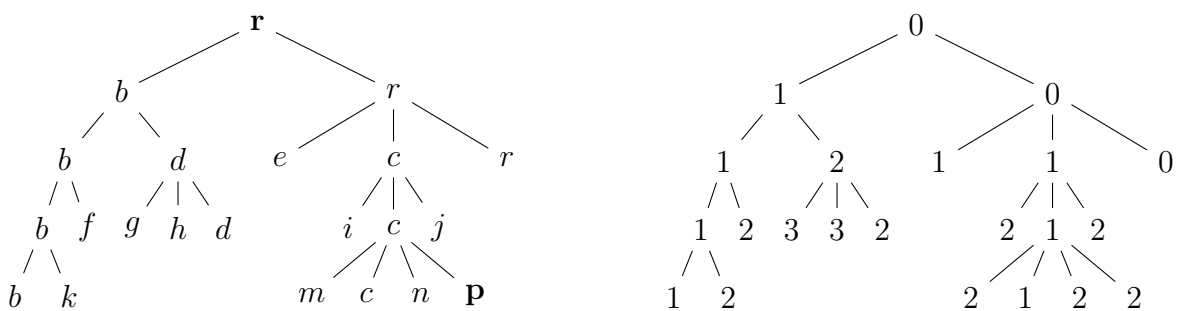


Figure 4: The left graph indicates an orbit decomposition of a rooted tree, where the nodes in the same orbit are labeled with the same letters. The number of descents of a node is the number of orbits passed on the path from the node to the root. For instance, the number of descents of \mathbf{p} is two since the path from \mathbf{p} to the root \mathbf{r} passes two orbits (orbit c and chief orbit r). The right graph shows the number of descents for all nodes.

From the above graphical point of view, the number of descents for a node is the number of orbits passed through by the path from the node to the root (cycle). See a demonstration of counting the number of descents in Figure 4. Also, an example showing a good orbit decomposition and a poor orbit decomposition of a comb tree is given in Figure 5.

Now, the problem is to find two types of orbit decompositions of a rooted tree that minimize the maximal number of descents and the average number of descents, respectively.

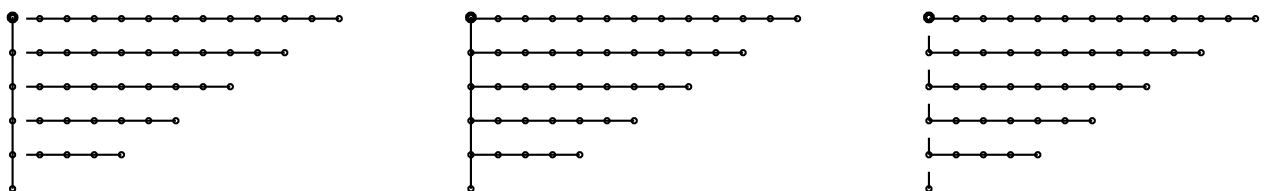


Figure 5: Middle: a tree rooted at the bold point. Left: a good orbit decomposition with a maximal number of descents 1. Right: a poor orbit decomposition with a maximal number of descents as large as 5. Actually, the left orbit decomposition is an orbit decomposition minimizing the maximal number of descents and the right orbit decomposition is the greedy orbit decomposition.

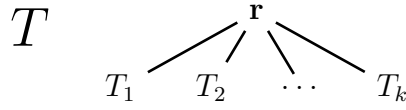


Figure 6: T is a tree rooted at \mathbf{r} with subtrees T_1, T_2, \dots, T_k .

3.2 An orbit decomposition minimizing the maximal number of descents

A bottom-up method is applied to construct optimal orbit decompositions. Let T be a tree rooted at \mathbf{r} with subtrees $T_1, \dots, T_k, k \geq 1$, as the graph shows in Figure 6. Assume that each subtree T_i has been given an orbit decomposition. Then there are k possible extensions of orbit decomposition to T . That is, we can prolong the chief orbit of any subtree T_i by including \mathbf{r} in the chief orbit and then it forms an extended orbit decomposition on T . The following lemma shows the increase of the number of descents for the extended orbit decomposition.

Lemma 1. Let $D_{old}(x)$ be the number of descents of x with respect to the existing orbit decompositions on subtrees T_1, \dots, T_k . If we extend the orbit decompositions to T by prolonging the chief orbit of T_a to \mathbf{r} , then the number of descents with respect to the extended orbit decomposition on T is $D_{new}(\mathbf{r}) = 0$ and

$$D_{new}(x) = \begin{cases} D_{old}(x) & \text{if } x \in T_a, \\ D_{old}(x) + 1 & \text{otherwise.} \end{cases} \quad (5)$$

Proof. It is obvious that (5) holds since $D_{new}(x)$ is the number of orbits passed on the path from x to \mathbf{r} . ■

Suppose that each subtree T_i is given an orbit decomposition that minimizes the maximal number of descents. What is the best choice of subtree T_a in Lemma 1 to minimize the maximal number of descents in T ? It is easy to guess that the answer is provided by the subtree that has a node with the largest $D_{old}(x)$. Moreover, if there is only one subtree that has a node with the largest $D_{old}(x)$ then the maximum of $D_{new}(x)$ is equal to the maximum of $D_{old}(x)$ by Lemma 1. Otherwise, if there is more than one subtree with the same largest $D_{old}(x)$, then the maximum of $D_{new}(x)$ is the maximum of $D_{old}(x)$ plus 1 by Lemma 1.

The function characterized by the above rule has been studied in other areas. It was originally used to classify river systems [7, 14] and later also appeared in computer science as register functions [6, 8]. The definition was on binary trees. Now, we extend it to rooted trees with branching factor ≥ 1 .

The extended Horton-Strahler (H-S) number. It is defined on nodes inductively by

$$S(\mathbf{r}) = \begin{cases} 0 & \text{if } \mathbf{r} \text{ has no child,} \\ M(\mathbf{r}) & \text{if there is only one child } \mathbf{u}_i \text{ with } S(\mathbf{u}_i) = M(\mathbf{r}), \\ M(\mathbf{r}) + 1 & \text{if there are at least two children } \mathbf{u}_i \text{ and } \mathbf{u}_j \text{ with } S(\mathbf{u}_i) = S(\mathbf{u}_j) = M(\mathbf{r}), \end{cases}$$

where $M(\mathbf{r}) = \max\{S(\mathbf{u}_1), \dots, S(\mathbf{u}_k)\}$ and $\mathbf{u}_1, \dots, \mathbf{u}_k$ are children of \mathbf{r} .

Average-case analysis of the H-S number on random binary trees was studied in [2, 5, 6, 8, 9, 10, 11, 13]. A different extension to m -ary trees was proposed and analyzed in [1, 3]. However, the extended H-S numbers for random functional graphs (random mappings) have not been investigated yet.

For each component, a “best” chief orbit should be a path from the root to a leaf passing the nodes with the largest extended H-S number locally (if there is more than one node with the same largest extended H-S number, then choose any one of them). Examples of a rooted tree associated with the extended H-S numbers and the number of descents for the nodes are given in Figure 7.

An algorithm to construct an optimal orbit decomposition: Step 1. Find the functional graph. Step 2. Compute the extended H-S numbers. Step 3. Create a to-do list of components. Step 4. Choose a best chief orbit from the first component in the to-do list. Step 5. Eliminate the chosen orbit and add new components into the to-do list. Stop when the to-do list is empty. The complexity of the algorithm is $O(N)$.

We have the following theorem.

Theorem 2. The least maximal number of descents is equal to the extended H-S number on the root of the tree.

Proof. The theorem can be proved easily by induction on N . It is trivial that the theorem holds for $N = 1$. For $N \geq 2$, assume that the theorem holds for all rooted trees with $N - 1$ nodes. Let T be a tree rooted at \mathbf{r} with N nodes and subtrees $T_1, \dots, T_k, k \geq 1$, as the graph shows in Figure 6. Let \mathbf{u}_i be the root of T_i . Then the least maximal number of descents for T_i is equal to $S(\mathbf{u}_i)$ by the assumption. By the definition of the extended H-S number and Lemma 1, the least maximal number of descents for T is equal to $S(\mathbf{r})$. ■

Remark. Let T be a rooted tree with N nodes and $S(T)$ be the extended H-S number of the root of T . Then we have

$$S(T) \leq \log_2(N + 1) - 1$$

and the equality holds if, and only if, T is a complete binary tree; or, equivalently, if $2^d - 1 \leq N < 2^{d+1} - 1$ then $S(T) \leq d - 1$, and if $N = 2^d - 1$ then $S(T) = d - 1$ if, and only if, T is a complete binary tree.

Proof of the remark. We prove the second proposition by induction on N . It is trivial that it holds for $N = 1$. Assume it holds for $N - 1$. Let T_1, \dots, T_k be the subtrees of T , as the graph shows in Figure 6. Without loss of generality, assume that $|T_1| \geq |T_2| \geq \dots \geq |T_k|$. There are two cases:

Case 1. $N = 2^d - 1$. If T is a complete binary tree then clearly $S(T) = d - 1$. Suppose T is not a complete binary tree. We will show $S(T) < d - 1$. Note that for all i , $|T_i| < 2^d - 1$ and thus $S(T_i) \leq d - 2$. Thus, it is enough to show that there is at most one i such that $S(T_i) = d - 2$. If $k = 2$ and $|T_1| = |T_2| = 2^{d-1} - 1$ then at least one of T_1 and T_2 is not a complete binary tree, and thus $\min\{S(T_1), S(T_2)\} < d - 2$. Otherwise, for all $i > 1$, we have $|T_i| < 2^{d-1} - 1$ and then $S(T_i) < d - 2$.
Case 2. $2^d - 1 < N < 2^{d+1} - 1$. Note that $|T_1| < N < 2^{d+1} - 1$, thus $S(T_1) \leq d - 1$. For all $i > 1$, we have $|T_i| < 2^d - 1$ and then $S(T_i) < d - 1$. Thus $S(T) \leq d - 1$. ■

3.3 An orbit decomposition minimizing the average number of descents

We follow the bottom-up method in the previous subsection. From Lemma 1 we have the following lemma.

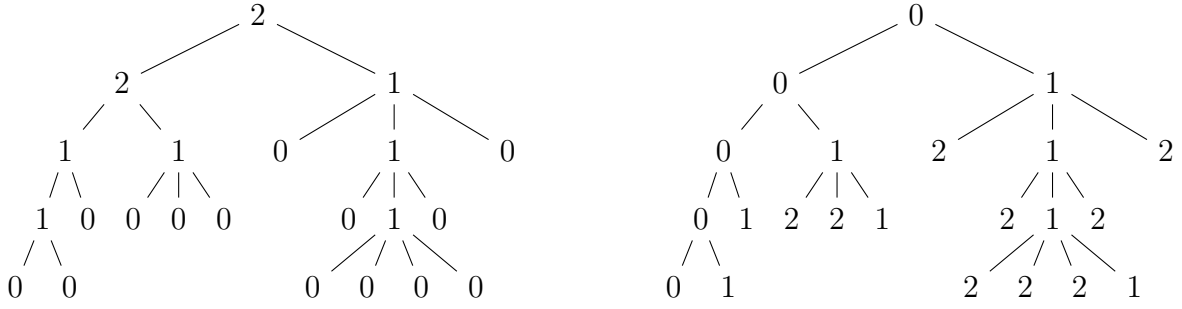


Figure 7: Left: the extended H-S numbers. Right: the number of descents for the nodes with respect to an orbit decomposition determined by the extended H-S numbers.

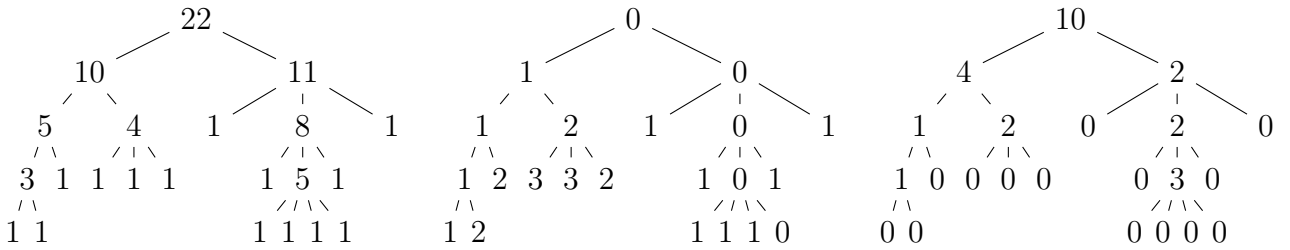


Figure 8: Left: the number of offspring for nodes (including the node itself). Middle: the number of descents for the nodes with respect to an orbit decomposition determined by the number of offspring. Right: the values of $\Lambda(\mathbf{r})$. Also, the sum of the numbers on the center tree is equal to the sum of the numbers on the right tree.

Lemma 3. Let m_i be the total number of descents for nodes in T_i with respect to the existing orbit decompositions on subtrees T_1, \dots, T_k . If we extend the orbit decompositions to T by prolonging the chief orbit of T_a to \mathbf{r} , then the total number of descents of nodes in T with respect to the extended orbit decomposition is equal to

$$\sum_{i=1}^k (m_i + |T_i|) - |T_a|. \quad (6)$$

To minimize (6), the best choice of subtrees should be the one with the largest number of nodes. Thus, a best chief orbit should be a path from the root to a leaf passing the nodes with the largest number of offspring locally (if there is more than one node with the same largest number of offspring then choose any of them). The construction of the orbit decomposition is similar to the algorithm in the previous subsection, except for the rule of choosing the orbit.

Define

$$\Lambda(\mathbf{r}) = \sum_{i=1}^k |T_i| - \max\{|T_1|, \dots, |T_k|\},$$

as the smallest increment in (6). Examples of a rooted tree associated with the number of offspring of nodes, the number of descents for the nodes with respect to an orbit decomposition determined by the number of offspring and the values of $\Lambda(\mathbf{r})$ are given in Figure 8.

It is easy to prove that the least total number of descents is equal to $\sum_{\mathbf{r}} \Lambda(\mathbf{r})$ by induction on N . Thus we have the following theorem.

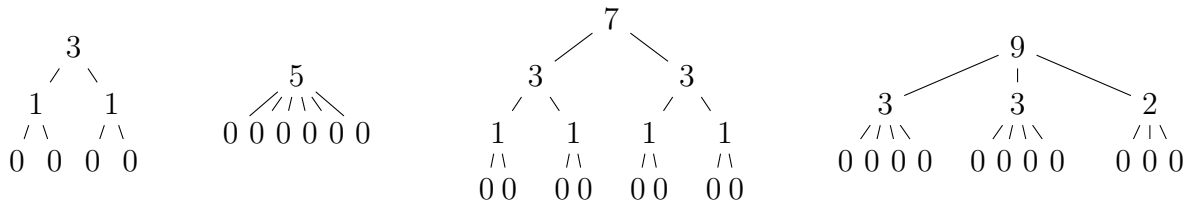


Figure 9: The numbers are the values of $\Lambda(\mathbf{r})$. The value of $\sum_{\mathbf{r}} \Lambda(\mathbf{r})$ is 5 and the number of nodes is 7 for the left pair of trees. For the right pair of trees, the value of $\sum_{\mathbf{r}} \Lambda(\mathbf{r})$ is 17 and the number of nodes is 15.

Theorem 4. The least average number of descents is equal to $\frac{1}{N} \sum_{\mathbf{r}} \Lambda(\mathbf{r})$.

For complete binary trees,

$$\frac{1}{N} \sum_{\mathbf{r}} \Lambda(\mathbf{r}) = \frac{\sum_{k=1}^d (2^{k-1} - 1) 2^{d-k}}{2^d - 1} = \frac{d 2^{d-1} - 2^d + 1}{2^d - 1} = \frac{d}{2} - 1 + \frac{d}{2^d - 1},$$

where d is the number of levels. Intuitively, one would guess that complete binary trees are the worst cases. However, we found some trees that are not complete binary trees, but they have the same values of $\sum_{\mathbf{r}} \Lambda(\mathbf{r})$ and the same number of nodes as complete binary trees. Examples are given in Figure 9. It is not easy to justify whether complete binary trees are always worst cases or not. The average (with respect to all functional graphs of the same number of nodes) of the value $\sum_{\mathbf{r}} \Lambda(\mathbf{r})$ is also unknown.

Remarks. We can give another interpretation of our work in terms of transportation systems. Consider the problem of designing a system of bus transportation without overlapping routes. It is analogous to an orbit decomposition of a graph. By regarding orbits as bus lines, descents as transfers and the root of a tree as the center of the system, our work can then be translated into designing two systems that minimize the maximal and average numbers of transfers from each stop to the center, respectively. The example in Figure 5 shows an efficient system (good orbit decomposition) and an inefficient system (poor orbit decomposition). However, it is not obvious if our method can be extended to general graphs. How to find an orbit decomposition that minimizes the number of “transfers” for general graphs is an interesting question.

References

- [1] D. Auber, M. Delest, J.P. Domenger, P. Duchon and J.M. Fédou, New Strahler numbers for rooted plane trees. In *Mathematics and Computer Science III*. Birkhäuser, Basel, 2004, pp. 203–215.
- [2] L. Devroye and P. Kruszewski, A note on the Horton-Strahler number for random trees. *Information Processing Letters* 56 (1995) 95–99.
- [3] M. Drmota and H. Prodinger, The register function for t -ary trees. *ACM Transactions on Algorithms* 2 (2006) 318–334.
- [4] P. Flajolet and A.M. Odlyzko, Random mapping statistics. *Lecture Notes in Computer Science* (1990) 329–354.

- [5] P. Flajolet and H. Prodinger, Register allocation for unary-binary trees. *SIAM J. Comput.* 15 (1986) 629–640.
- [6] P. Flajolet, J.C. Raoult and J. Vuillemin, The number of registers required for evaluating arithmetic expressions, *Theoretical Computer Science* 9 (1979) 99–125.
- [7] R.E. Horton, Erosional development of streams and their drainage basins; hydro-physical approach to quantitative morphology, *Bull. Geological Soc. America* 56 (1945) 275–370.
- [8] R. Kemp, The average number of registers needed to evaluate a binary tree optimally, *Acta Inform.* 11 (1979) 363–372.
- [9] A. Meir and J.W. Moon, Stream lengths in random channel networks, *Congressus Numerantium* 33 (1980) 25–33.
- [10] A. Meir, J.W. Moon and J.R. Pounder, On the order of random channel networks, *SIAM J. Algebraic Discrete Methods* 1 (1980) 25–33.
- [11] J.W. Moon, On Horton’s law for random channel networks, *Annals of Discrete Mathematics* 8 (1980) 117–121.
- [12] M. Naor and O. Reingold, Constructing pseudo-random permutations with a prescribed structure, *J. Cryptology* 15 (2002) 97–102.
- [13] H. Prodinger, Some recent results on the register function of a binary tree. *Annals of Discrete Mathematics* 33 (1987) 241–260.
- [14] A.N. Strahler, Hypsometric (area-altitude) analysis of erosional topology, *Bull. Geological Soc. America* 63 (1952) 1117–1142.
- [15] B. Tsaban, Permutation graphs, and sampling the cycle structure of a permutation, *J. Algorithms* 47 (2003) 104–121.
- [16] B. Tsaban, Decompositions of graphs of functions and fast iterations of lookup tables. *Discrete Applied Mathematics* 155 (2007) 386–393.