University of Nevada, Reno

# Ordering and Self-Similarity in Non-Binary Trees

A thesis submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Science in Mathematics and the Honors Program

by

Nicholas Cleymaet

Dr. Ilya Zaliapin, Thesis Advisor

May, 2017

**UNIVERSITY OF NEVADA RENO**

**THE HONORS PROGRAM**

We recommend that the thesis
prepared under our supervision by

**Nicholas Cleymaet**

entitled

**Ordering and Self-Similarity in Non-Binary Trees**

be accepted in partial fulfillment of the
requirements for the degree of

**Bachelor of Science in Mathematics**

---

Ilya Zaliapin, Ph. D., Thesis Advisor

---

Tamara Valentine, Ph. D., Director, **Honors Program**

May, 2017

# Abstract

Ordering and self-similarity in binary trees has been well-studied. Self-similarity was first observed in river networks, and has since been shown to have many useful properties and modeling applications, from leaf vein structures to level-set trees of symmetric Markov chains. Systems for assigning orders to the vertices of non-binary trees, though, have not been the subject of such thorough inquiry, despite potential applications in modeling stochastic processes. The Horton-Strahler ordering for binary trees has been central to the study of self-similarity in random trees. We introduce two classes of multiplier-sum orderings which generalize the Horton-Strahler ordering for binary trees to the space of trees with arbitrary branching and demonstrate that each has an associated pruning process, along with proving some basic properties of these orderings and relationships between them. We also present the results of preliminary numerical simulations which demonstrate the potential validity of Horton law in non-binary trees under these orderings.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction and Literature Review

The first observations of self-similar branching structures came from Horton's study of river networks (1945). The ordering of tree vertices comes from stream orders in river networks, which can be represented as binary trees, with edges representing streams and vertices representing the points where two streams join. Horton, introducing an ordering system which assigned higher orders to streams fed by streams of lower orders, observed that the ratio of the number streams of order $i$ (represented in the associated tree by branches of order $i$) to the number of streams of order $i+1$ was approximately constant for all $i$ in the river networks he studied. He also observed that this ratio $R$ was approximately equal to 4. Strahler (1952, 1957) expanded on Horton's results, observing several properties of drainage basins which correlated with the order of their river networks, and proposed a modification to the ordering system used by Horton. The resulting ordering system is the Horton-Strahler ordering, which has been used without further modification in the study of hydrology and in the study of self-similarity in other binary branching structures since.

Shreve (1967) introduced a theoretical model of topologically random channel networks, which are equivalent to critical binary Galton-Watson trees conditioned to have exactly $n$ leaves (Burd, Waymire, & Winn, 2000) and showed that such trees followed a Horton law and that, for large $n$, the ratio $\frac{N_i}{N_{i+1}}$ converges to 4 (here, $N_i$ denotes the number of branches of order $i$). He later expanded upon his results, showing that $T_{ij}$ (the mean number of side branches of order $i$ per branch of order $j$) converges to $2^{j-i-1}$ for large $n$ (Shreve, 1969).

Tokunaga (1978) introduced a broader class of structurally self-similar branching structures, with the constraint $T_{ij} = ac^{j-i-1}$ for some positive $a, c$. Peckham (1995) corroborated the validity of this two-parameter model for river networks, while Zanardo, Zaliapin, and Foufoula-Georgiou (2013) observed that the parameters of the model showed low variance between regions.

It has been shown that, under certain conditions, Horton law follows from self-similarity. McConnell and Gupta (2008) showed that the Tokunaga constraint $T_{ij} = ac^{j-i-1}$ is sufficient for this to occur and that, in this case,

$$R = \frac{2 + a + c + \sqrt{(2 + a + c)^2 - 8c}}{2}$$

Kovchegov and Zaliapin (2016) expanded the conditions under which a Horton law follows from self-similarity, showing that this occurs whenever the limit of the sequence $\left(T_k^{1/k}\right)$ is finite.

Self-similarity and Horton law were first observed in river networks, but further research has shown that other naturally occurring branching structures exhibit these properties as well. Most notably, self-similarity has been demonstrated in diffusion-limited aggregation, leaf vein structures, and phylogenetic trees, and likely applies to other biological systems as well (Newman, Turcotte, & Gabrielov, 1997; Peckham, 1995; Phelps, 2015).

Burd et al. (2000) expanded upon Shreve's results for topologically random channel networks, demonstrating that Galton-Watson trees, in general, are self-similar if and only if they have critical binary offspring distribution. Similarly, Kovchegov and Zaliapin (2012; 2017) demonstrated that level-set trees of symmetric Markov chains and a tree representation of the celebrated Kingman's coalescent process are self-similar.

Self-similarity appears to be an intrinsic property of many natural branching structures and theoretical models; this has been well-established in the space of binary trees. Results related to self-similarity in non-binary trees are more limited, though. Burd et al. (2000) and Zaliapin and Kovchegov (2012) state that their results are generalizable to the non-binary case, but these generalizations rely on the equivalence of self-similarity to invariance under the operation of pruning (which produces a smaller tree by eliminating the leaves of a tree and their parental edges). While the ordering related to the classical operation of pruning is a valid generalization of the Horton-Strahler ordering to the space of trees with arbitrary (not necessarily binary) branching, there is a much broader class of potential generalizations worth considering. Exploring this broader class of generalizations is the focus of this thesis.

# Chapter 2

# Preliminaries

## 2.1 Graphs and Trees

The objects of our focus are rooted tree graphs, so we will begin by establishing relevant definitions for the study of those objects.

**Definition 1.** A *graph* $G = (V, E)$ is a pair of a set of vertices $V$ and a set of edges $E$ which connect pairs of vertices.

A *chain* in a graph is a sequence $(v_1, e_1, v_2, e_2, v_3, \ldots, e_{n-1}, v_n)$ of vertices $v_i \in V$ and edges $e_i \in E$, where $e_i$ connects $v_i$ to $v_{i+1}$ for all $i \in \{1, \ldots, n\}$. Essentially, a chain forms a path from a vertex $v_1$ to a vertex $v_n$ following the edges in the graph.

A graph $G$ is called *connected* if for every $v, w \in V$, there exists a chain $(v, \ldots, w)$ in $G$. A chain $(v_1, e_1 \ldots, e_{n-1}, v_n)$ is called *simple* if $e_i \neq e_j$ for all $i, j$ distinct. A chain $(v_1, \ldots, v_n)$ is called *closed* if $v_1 = v_n$. A closed, simple chain is called a *circuit*.

**Definition 2.** A connected graph with no circuits is called a *tree*. Equivalently, a tree is a graph in which there is exactly one simple chain $(v, \ldots, w)$ for each pair $(v, w) \in V \times V$ with $v \neq w$.

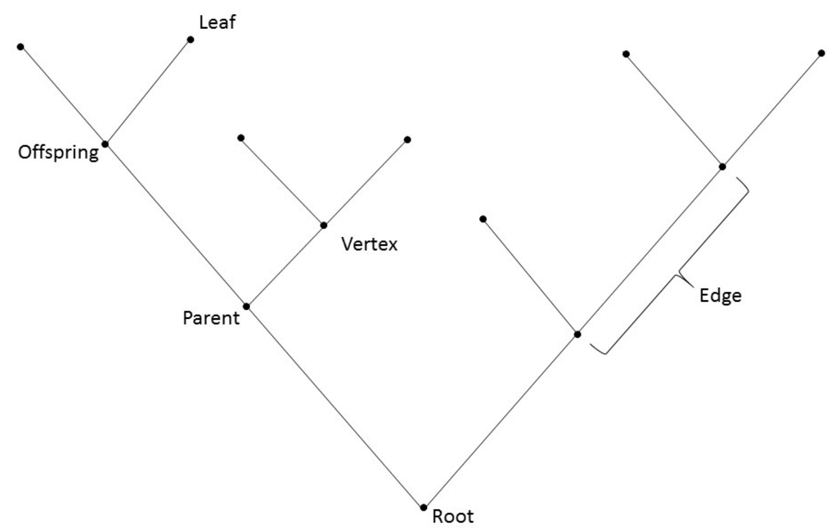


Figure 2.1: A Binary Tree

The *distance* between to vertices $v, w$ in a connected graph is the number of edges in the shortest chain of the form $(v, \ldots, w)$. This chain is necessarily simple.

**Definition 3.** A *rooted tree* is a tree with a vertex $v_0$ is designated as the *root* of the tree.

All trees discussed in this paper are rooted and finite.

**Definition 4.** An *offspring* of a vertex $k$ is a vertex which is adjacent to $k$ (connected to $k$ by an edge) and further from the root of the tree than $k$, where distance to the root $v_0$ for a vertex $v \in V$ is measured by the length of the chain $(v_0, \ldots, v)$. If a vertex $l$ is the offspring of a vertex $k$, then $l$ is called the *parent* of $k$. A vertex with no offspring is called a *leaf*.

In a rooted tree, there is a natural sense of direction. Since the study of self-similarity in binary trees began studying tree representations of rivers, these directions are often called *upstream* and *downstream*. Since these terms are less clear in reference to combinatorial trees, I will instead use *leafward* and *rootward*.

**Definition 5.** Given a tree $T$ and a vertex $k \in T$, the *subtree rooted at* $k$ is the tree made up of $k$ and all edges and vertices leafward from $k$. The *mass* of a vertex $k$ is the number of leaves in the subtree rooted at $k$.

**Definition 6.** A tree $T$ is called *binary* if each non-leaf vertex $k \in T$ has exactly two offspring. A *perfect binary tree* is a binary tree in which each leaf has the same distance from the root. See figure 2.1 for an example of a binary tree.

A *random tree* is a tree chosen from some space of trees based on a probability distribution. A *binary random tree* is a random tree chosen from the space of binary trees.

Let $\mathbb{T}$ denote the space of (rooted) trees, and let $\mathbb{T}_B$ denote the space of binary (rooted) trees.

### 2.1.1 Galton-Watson Trees

A *Galton-Watson tree* is a random tree generated by the following process:

1. Begin with a single vertex. This is the first generation.

2. Choose a probability distribution $P$ on $\mathbb{N} \cup \{0\}$.

3. To generate the $(k+1)$th generation: For each vertex $i$ in the $k$th generation, create $X_i$ offspring for $i$, where $X_i \sim P$.

A *critical Galton-Watson tree* is a Galton-Watson tree with $E(X) = 1$, where $X$ is a random variable representing the number of offspring for a given vertex. In particular, the *critical binary Galton-Watson tree* has $P(0 \text{ offspring}) = P(2 \text{ offspring}) = \frac{1}{2}$.

## 2.2 Ordering and Self-Similarity in Binary Trees

### 2.2.1 Ordering in Perfect Binary Trees

Given a perfect binary tree $T$, we can assign orders to the vertices of $T$ by the following process:

1. Assign each leaf order 1
2. Assign each non-leaf vertex an order one greater than the order of its offspring.

Let $M(k)$ denote the mass of a vertex $k$ of order $i$. Under this ordering:

$$M(k) = 2^{i-1}$$

This process only works if the orders of the offspring of $k$ are equal for all $k \in T$; this does not hold for binary trees in general.

### 2.2.2 Horton-Strahler Ordering

Since the ordering scheme proposed in the previous section does not apply to binary trees in general and trees encountered in nature are often not perfect, there was a need for a meaningful generalization. The ordering scheme proposed by Horton (1945) and refined by Strahler (1952, 1957), defined as follows, fills this need.

**Definition 7.** The *Horton-Strahler Ordering* assigns orders to the vertices in a tree $T \in \mathbb{T}_B$ by the following process:

1. Assign order 1 to all leaves.
2. Given non-leaf vertex $k \in T$, let $i$ and $j$ be the orders of the offspring of $k$. Assign order $\max(i, j)$ to $k$ if $i \neq j$, and assign order $\max(i, j) + 1$ to $k$ if $i = j$.

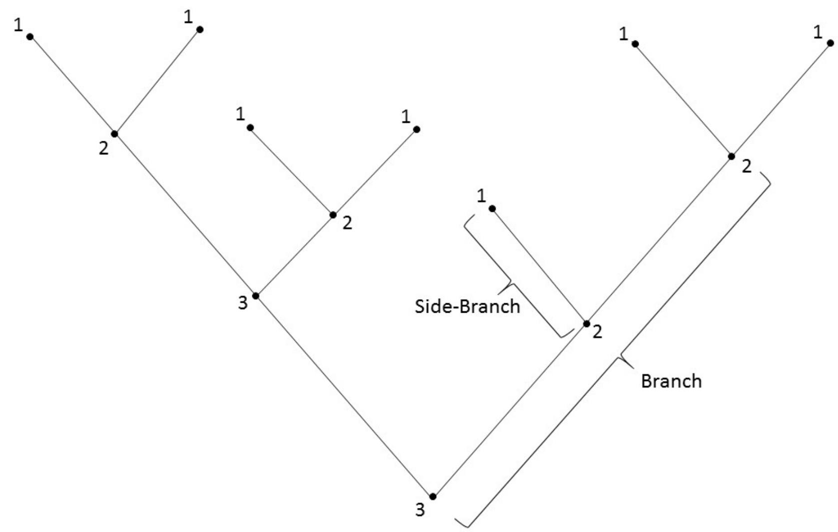


Figure 2.2: A binary tree with the Horton-Strahler orders of its vertices

See figure 2.2 for an example of the Horton-Strahler ordering applied to a binary tree.

A *branch* in a tree is a connected set $B$ of vertices of the same order such that there is no larger connected set $V \supset B$ of vertices of the same order, together with the parental edges of vertices in $B$. In binary trees under the Horton-Strahler Ordering, each branch will necessarily be a chain, but this is not true in general.

**Definition 8.** A *side-branch* is a branch whose rootmost vertex is an offspring of a vertex in another branch, but which does not contribute to increasing the order of that branch. in a binary tree, this is equivalent to beaing an offspring of a non-leafmost vertex in another branch.

Observe that the length of a branch in a binary tree is equal to the total number of side-branches for that branch plus one.

### 2.2.3 Pruning

*Pruning* is a function $\mathcal{R} : \mathbb{T} \to \mathbb{T}$ which, given a tree $T \in \mathbb{T}$, produces $\mathcal{R}(T)$ through the following process (see figure 2.3 for a demonstration):

1. If this is the first iteration of pruning, prime $T$ by performing series reduction. Note that this will have no effect in the binary case; it will, however, become necessary when $T$ can have vertices with only one offspring.

2. Eliminate all leaves.

3. Perform series reduction, eliminating all vertices which are offspring of vertices which have exactly 1 offspring, repeating until no such vertices exist.

If a vertex is eliminated in series reduction, it is part of the same branch as its parent.

**Remark 1.** Given a tree $T \in \mathbb{T}_B$ and a vertex $k \in T$, let $T^k$ denote the subtree rooted at $k$ and let $\omega_k$ denote the Horton-Strahler order of $k$. Then the number of iterations of pruning process necessary to eliminate $T^k$ is equal to the Horton-Strahler order of $k$.

$$\omega_k = \min\{j \mid \mathcal{R}^j(T^k) = \emptyset\}$$

### 2.2.4 Self-Similarity

Given a binary random tree $T$, let $\Omega$ denote the order of $T$ (the order of its root), let $N_i$ denote the number of branches of order $i$. For any $m$ be a branch of order $j$, let $\tau_{ij}^{(m)}$ denote the number of side branches of order $i$ for $k$ and let the *Tokunaga coefficients* $T_{ij}$ for $T$ denote the mean of $\{\tau_{ij}^{(m)}\}$ for $i, j$ fixed (we assume that $\{\tau_{ij}^{(m)}\}$ are drawn from the same distribution for fixed $i, j$).

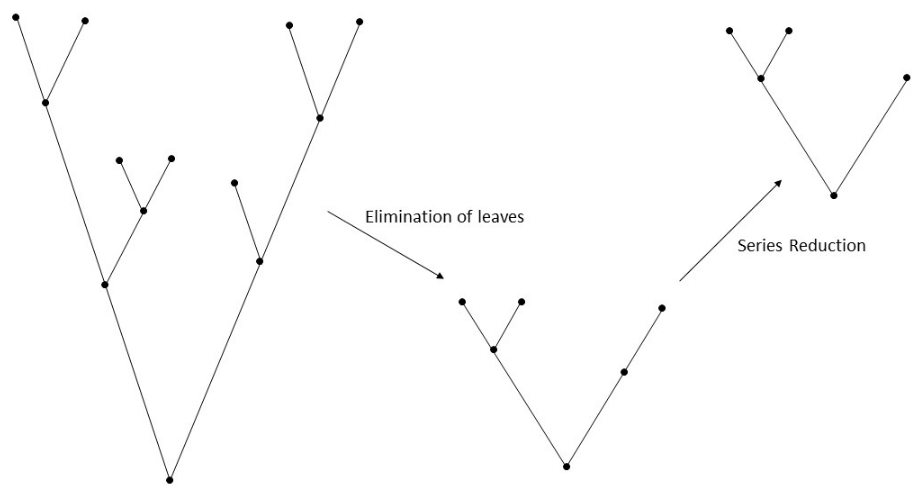


Figure 2.3: An example of pruning applied to a binary tree

1. $T$ is said to follow a *Horton Law* if the ratio $\lim_{\Omega\to\infty} \frac{E(N_i)}{E(N_{i+1})} = R$ for all $i$ and for some $R > 0$.

2. $T$ is called *self-similar* if $E(T_{ij}) = T_{j-i} = T_k$ for all $i < j$. That is, in a self-similar random tree, $E(T_{ij})$ depends only on the difference $j - i$.

3. $T$ is called *Tokunaga self-similar* if it is self-similar with $T_k = ac^k$ for some $a, c > 0$.

4. $T$ is called *stochastically self-similar* if it is distributionally invariant under pruning.

As noted in the introduction, these properties appear to be followed in many physical systems and stochastic processes that can be modeled by binary trees (Kovchegov & Zaliapin, 2017; Newman et al., 1997; Peckham, 1995; Phelps, 2015; Zaliapin & Kovchegov, 2012).

Let $M(k)$ denote the mass of a vertex $k$ of order $i$. In a self-similar tree with

Horton Ratio $R$:

$$E[M(k)] = R^{i-1}$$

The following four theorems are significant results related to self-similarity. The first three show how different form of self-similarity in binary trees are related to each other, while the fourth gives the conditions under which Galton-Watson trees are self-similar.

**Theorem 1.** *(Burd et al., 2000) Let $T$ be a Galton-Watson tree. Then $T$ is stochastically self-similar if and only if $T$ is self-similar.*

**Theorem 2.** *(McConnell & Gupta, 2008) Let $T$ be a random tree and suppose $T$ is Tokunaga self-similar with $T_k = ac^k$. Then $T$ follows a Horton law and the Horton ratio $R$ is given by*

$$R = \frac{2 + a + c + \sqrt{(2 + a + c)^2 - 8c}}{2}$$

**Theorem 3.** *(Kovchegov & Zaliapin, 2016) Let $T$ be a random tree and suppose $T$ is self-similar with $\lim T_k^{1/k} < \infty$. Then $T$ follows a Horton law.*

**Theorem 4.** *(Burd et al., 2000) Let $T$ be a Galton-Watson tree. Then $T$ is Tokunaga self-similar if and only if it has critical binary offspring distribution ($P$(0 offspring) $=$ $P$(2 offspring) $= \frac{1}{2}$). In this case, $T_k = 2^k$ and $R = 4$. If $T$ has critical offspring distribution, then $\lim_{n \to \infty} \mathcal{R}^n(T)$ has critical binary offspring distribution.*

# Chapter 3

# Ordering in Trees with Arbitrary Branching

## 3.1 Orderings

We will now propose generalizations of the Horton-Strahler ordering to the space of trees with arbitrary branching, beginning with defining orderings in the most general sense before moving on to defining more specifically the orderings which will be the focus of our study.

**Definition 9.** An *ordering* on $\mathbb{T}$ is a function which assigns to each vertex $k$ in a tree $T \in \mathbb{T}$ an order $i \in \mathbb{N}$. A *proper ordering* on $\mathbb{T}$ is an ordering which satisfies the following properties:

1. All leaves in $T$ are assigned order 1.

2. No vertex is assigned an order less than the maximum order of its offspring.

If an ordering is proper, it follows that the orders of all of its vertices will be positive integers.

Let $n_{i,k}$ denote the number of offspring of order $i$ for a given vertex $k$ and let $I_k$ denote the set of orders of offspring of a vertex $k$.

**Definition 10.** Let $n_{i,k}$ denote the number of offspring of order $i$ for a vertex $k$. Assign a multiplier $m_{i,j} \in \mathbb{R}$ to each ordered pair $(i, j) \in \mathbb{N}^2$. A *multiplier-sum ordering* assigns each vertex $k \in T$ an order $\omega_k$ as follows:

$$\omega_k = \begin{cases} \max\{j \mid \sum_i m_{i,j} n_{i,k} \geq 1\} & \text{if there exists any such } j \\ 1 & \text{otherwise} \end{cases}$$

**Proposition 1.** *A multiplier-sum ordering is proper if and only if* $m_{i,i} \geq 1$ *for all* $i$.

*Proof.* Let $T \in \mathbb{T}$ and let $k \in T$. Suppose the vertices of $T$ are assigned order according to a multiplier-sum ordering such that $m_{i,i} \geq 1$ for all $i$. If $k$ is a leaf, then $\sum_i m_{i,j} n_{i,k} = 0$ for all $j$, and so $k$ will be assigned order 1. If $k$ is not a leaf, then let $I = \max\{i \in I_k\}$. Then $k$ has at least one offspring of order $I$, so $\sum_i m_{I,I} n_{I,k} \geq 1$, and therefore the order of $k$ is at least $I$. Thus, $m_{i,i} \geq 1$ for all $i$ is sufficient for a multiplier-sum ordering to be proper.

Now consider a tree which has some vertex $k$ which has exactly one offspring of order $i$ (such trees will exist for any $i$ chosen). If $m_{i,i} < 1$, then vertex $k$ would be assigned order less than $i$, and therefore the ordering would not be proper, and so $m_{i,i} \geq 1$ for all $i$ is a necessary condition for a multiplier-sum order to be proper. Thus, a multiplier-sum ordering is proper if and only if $m_{i,i} \geq 1$ for all $i$. □

**Definition 11.** The *Type I ordering* is a multiplier-sum ordering with $m_{i,i+1} = \frac{1}{s}$ and $m_{i,i} = 1$ for all $i$ and for some $s \in \mathbb{N}$, and $m_{i,j} = 0$ otherwise.

In this ordering, the order of the parent depends only on maximum order of its offspring and the number of offspring with that maximum order—a parent will have order one greater than the maximum order of its offspring if it has at least $s$ offspring

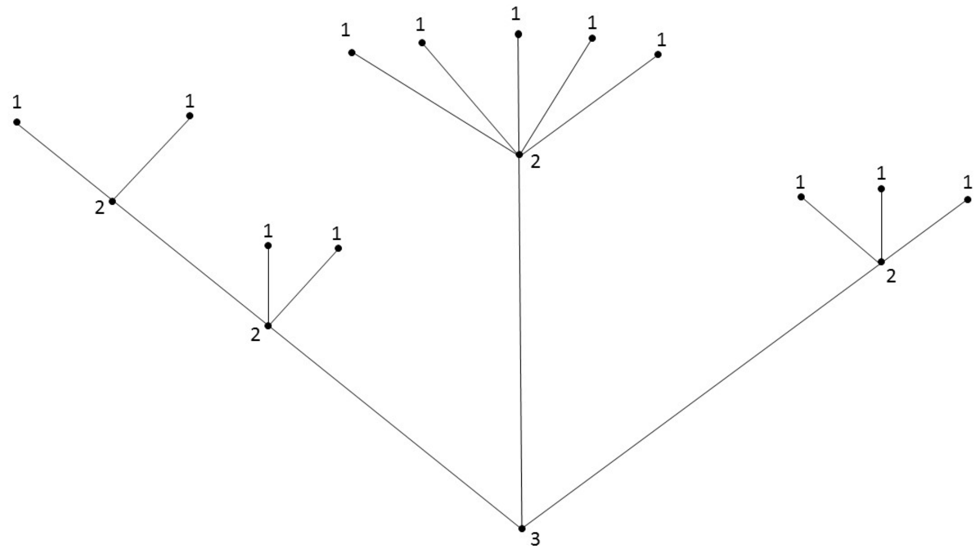


Figure 3.1: A tree with the type I ($s = 2$) orders of its vertices

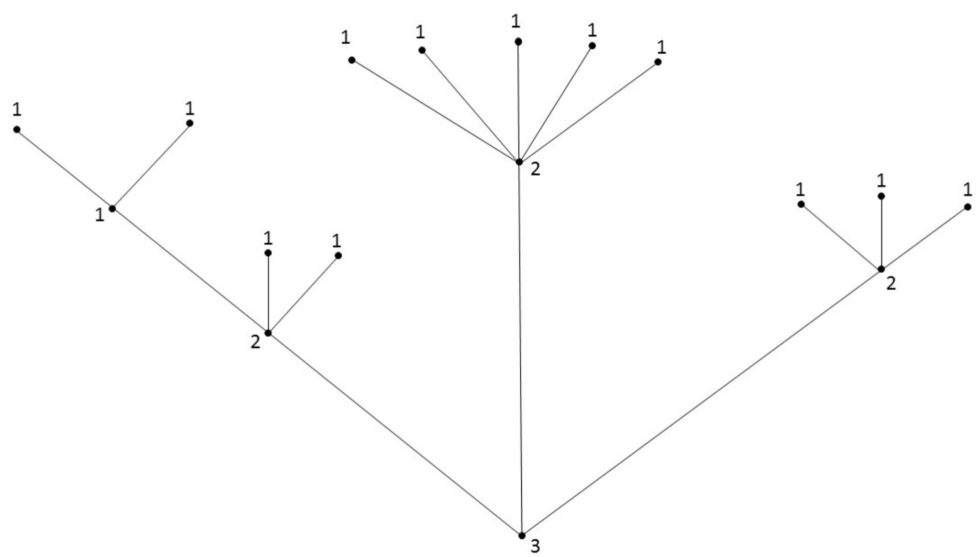


Figure 3.2: A tree with the type I ($s = 3$) orders of its vertices

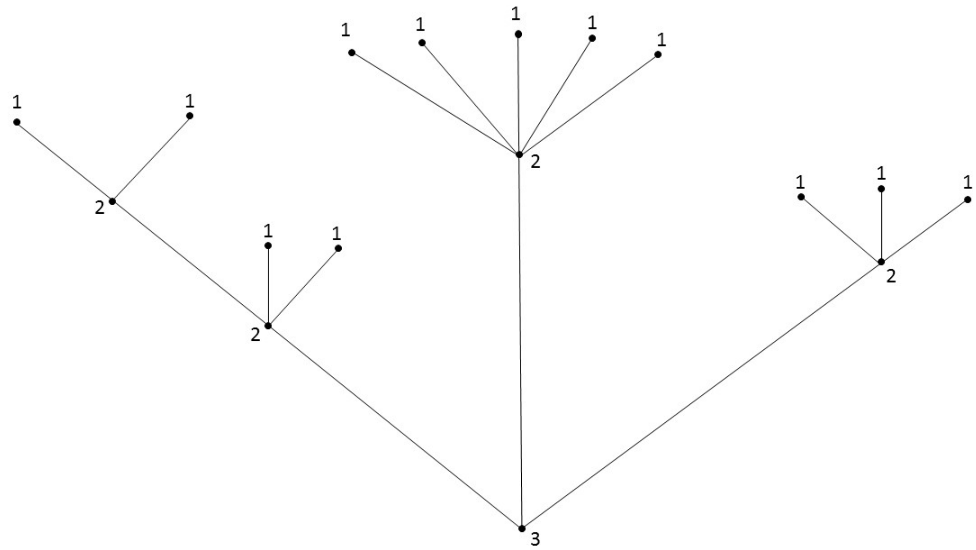


Figure 3.3: A tree with the type II ($s = 2$) orders of its vertices

with that maximum order, and will have an order equal to the maximum order of its offspring otherwise.

**Definition 12.** The *Type II ordering* is a Multiplier-Sum Ordering with $m_{i,j} = \frac{1}{s^{j-i}}$ for all $j \geq i$ and for some $s \geq 2$.

Under the type II ordering, the count of offspring of any order contributes to the order of a parent vertex, rather than only those of maximum order, as was the case under the type I ordering. In particular, the effect of $s$ offspring of order $i$ is equivalent to the effect of one offspring of order $i + 1$ for all $i \in \mathbb{N}$.

**Remark 2.** The type I and type II orderings both reproduce the Horton-Strahler ordering when applied to binary trees.

See figures 3.1-3.3 for examples of the type I and type II orderings applied to non-binary trees.

**Proposition 2.** *Given a tree $T$ and a vertex $k \in T$, let $\omega_1$ denote the type I order of $k$ and let $\omega_2$ denote the type II order of $k$, both with parameter $s$. Then $\omega_2 \geq \omega_1$.*

*Proof.* Since all multipliers which are nonzero in the type I ordering have the same value in the type II ordering, no vertex with the same offspring counts can be assigned lower order. As a consequence, any vertices which have different offsring counts under the type II ordering will have offspring of greater order, and no offspring of lower order, and so will be assigned order at least as great as they would be assigned under the type I ordering. □

**Proposition 3.** *Given a tree $T$ and a vertex $k \in T$, let $s, t \in \mathbb{N} \setminus \{1\}$ with $s < t$. Let $\omega_s$ denote the order of $k$ with parameter $s$ and let $\omega_t$ denote the order of $k$ with parameter $t$, both type I or both type II. Then $\omega_s \geq \omega_t$.*

*Proof.* $m_{i,j}^{(s)} \geq m_{i,j}^{(t)}$ for all $i, j$, so no vertex with the same offspring counts can be assigned lower order under parameter $s$ than under parameter $t$. As above, this implies that no vertex will be assigned lower order. □

**Proposition 4.** *Let $n_k$ denote the number of offspring of a vertex $k$. If $n_k \leq M$ for some $M \in \mathbb{N}$ for all $k$, then the type I and type II orderings are equivalent for all $s > \frac{M+1}{2}$. However, the ordering is trivial (assigns order 1 to all vertices) for $s > M$.*

*Proof.* Let $T \in \mathbb{T}$, let $n_k$ denote the number of offspring of a vertex $k$, and let $M \in \mathbb{N}$. Suppose $\sum n_k \leq M$ for all $k \in T$.

First note that, for any order $i$, a vertex must have at least $s$ offspring in order to have order $j > i$ (under either ordering). Thus if $s > M$, no vertices will be assigned order greater than the maximum order of their offspring, and so each vertex will have order 1.

Now note that, in order for the type II ordering to assign greater order to a vertex than it was assigned under the type I ordering with the same parameter, it must have at least the equivalent of $s$ offspring of maximum order, but fewer than $s$ offspring which are exactly maximum order. It must therefore have at least $2s-1$ offspring ($s-1$ of maximum order, and $s$ of order 1 less than maximum order—other configurations would require more offspring). Thus, the type I and type II orderings are equivalent for all $s > \frac{M+1}{2}$. □

## 3.2 Binary Decompositions

Given a non-binary tree $T$, we can construct an associated binary tree. The associated tree is not unique, but here we consider two particular methods of construction.

**Definition 13.** Given a tree $T$ for which no vertices have only one offspring, a *binary decomposition* of $T$ is a binary tree produced by adding intermediate vertices and edges which are offspring of vertices in $T$ and have offspring which, in $T$, are offspring of their parent. If $T$ has vertices with one offspring, a binary decomposition can be performed on $T'$, which can be produced by performing series reduction on $T$. In this case, define the binary decomposition of $T$ to be equivalent to the binary decomposition of $T'$.

**Definition 14.** To obtain the *type I binary decomposition* of a tree $T$, apply the following process to $T$

1. Assign order 1 to all leaves in $T$.
2. If all vertices have been assigned orders, stop. The current tree is the type I binary decomposition of $T$.
3. Find a vertex $K$ for which all offspring of $K$ have been assigned an order, but $K$ itself has not.

4. If $K$ has exactly 2 offspring, assign $K$ an order according to the Horton-Strahler ordering, then return to 2.

5. If $K$ has more than two offspring, find vertices $k_1$ and $k_2$ which are offspring of $K$ such that the sum of the orders of $k_1$ and $k_2$ is maximized (the order of $k_1$ will be equal to the maximum order of the offspring of $K$, and the order of $k_2$ will be equal to the maximum order of the offspring of $K$ other than $k_1$).

6. Add an intermediate vertex with offspring $k_1$ and $k_2$ and parent $K$. And assign it an order according to the Horton-Strahler ordering, then return to 4.

Note that this process of binary decomposition also assigns each vertex in the binary decomposition of $T$ an order according to the Horton-Strahler ordering.

**Remark 3.** The type I binary decomposition is a minimum-order binary decomposition.

**Theorem 5.** *The type I binary decomposition of $T$ assigns the type I ($s = 2$) orders to the vertices of $T$.*

*Proof.* Let $T$ be a tree and let $k$ be a vertex in $T$ and let $I$ be the maximum order of the offspring of $k$. Recall that, under the type I ordering with $s = 2$, $k$ is assigned order 1 if $k$ is a leaf, order $I + 1$ if $k$ has at least two offspring of order $I$, and order $I$ if $k$ has exactly one offspring of order $I$, so it suffices to show that the type I binary decomposition of $T$ assigns order according to these same parameters.

Suppose a vertex has order 1 under the type I ordering with $s = 2$. Then it has exactly one offspring of order 1 and no offspring of other orders or is a leaf, and therefore is either a leaf in $T'$ or (and thus assigned order 1) is eliminated in series reduction, and is therefore not in the binary decomposition of $T$.

Now suppose that for some $n \in \mathbb{N}$, the type I binary decomposition of $T$ assigns order $i$ to a vertex $k$ if and only if the order of $k$ in $T$ under the type I ordering with

$s = 2$ is $i$ for all $i \leq n$. Let $K$ be a vertex of order $n+1$ in $T'$ under the type I ordering with $s = 2$. Then either $K$ has exactly one offspring of order $n+1$ (with maximum order of its offspring $n+1$) or $K$ has at least two offspring of order $n$ (with maximum order of its offspring $n$).

Case 1 ($K$ has exactly one offspring of order $n+1$, and no offspring of greater order): If $K$ has exactly two offspring, then $K$ is assigned order $n+1$ under the Horton-Strahler Ordering. If $K$ has more than two offspring, then an intermediate vertex will created which will have one offspring of order $n+1$ and one offspring of lesser order, and so will be assigned order $n+1$, after which $K$ will again fall under Case 1. The process will repeat until $K$ has exactly two offspring and is assigned order $n+1$ as above.

Case 2 ($K$ has at least two offspring of order $n$ and no offspring of greater order): If $K$ has exactly two offspring, then $K$ is assigned order $n+1$ under the Horton-Strahler ordering. If $K$ has more than two offspring, then an intermediate vertex will created which will have two offspring of order $n$, and so will be assigned order $n+1$, after which $K$ will fall under Case 1 (see above).

Hence, a vertex $k$ in the binary decomposition of $T$ has order $n+1$ in the type I binary decomposition of $T$ if and only if it has order $n+1$ in $T$ under the type I ordering with $s = 2$.

Thus, a vertex $k$ which is in both $T$ and the binary decomposition of $T$ has order $i$ if and only if it has order $i$ in $T$ under the type I ordering with $s = 2$. □

**Definition 15.** To obtain the *type II binary decomposition* of a tree $T$, apply the following process to $T$

1. Assign order 1 to all leaves in $T$.

2. If all vertices have been assigned orders, stop. The current tree is the type II binary decomposition of $T$.

3. Find a vertex $K$ for which all offspring of $K$ have been assigned an order, but $K$ itself has not.

4. If $K$ has exactly 2 offspring, assign $K$ an order according to the Horton-Strahler ordering, then return to 2.

5. If $K$ has more than two offspring, find vertices $k_1$ and $k_2$ which are offspring of $K$ such that the sum of the orders of $k_1$ and $k_2$ is minimized (the order of $k_1$ will be equal to the minimum order of the offspring of $K$, and the order of $k_2$ will be equal to the minimum order of the offspring of $K$ other than $k_1$).

6. Add an intermediate vertex with offspring $k_1$ and $k_2$ and parent $K$. And assign it an order according to the Horton-Strahler ordering, then return to 4.

**Remark 4.** The type II Binary Decomposition is a maximum-order binary decomposition.

**Theorem 6.** *The type II binary decomposition of $T$ assigns the type II ($s = 2$) orders to the vertices of $T$.*

*Proof.* Let $T$ be a tree and let $k$ be a vertex in $T$. Recall that, under the type II ordering with $s = 2$, $k$ is assigned order 1 if $k$ is a leaf, and is assigned order $I$ otherwise, where $I$ is the maximum $j$ such that $\sum_i \frac{1}{2^{j-i}} n_{i,k} \geq 1$

Suppose a vertex has order 1 under the type II ordering with $s = 2$. Then it has exactly one offspring of order 1 and no offspring of other orders or is a leaf, and therefore is either a leaf in $T'$ or (and thus assigned order 1) is eliminated in series reduction, and is therefore not in the binary decomposition of $T$.

Observe that the type II binary decomposition replaces $n_1$ vertices of order 1 with $\lfloor \frac{n_1}{2} \rfloor$ vertices of order 2, then $n_2$ vertices of order 2 with $\lfloor \frac{n_2}{2} \rfloor$ vertices of order 3, and so on, replacing $n_i$ vertices of order $i$ with $\lfloor \frac{n_i}{2} \rfloor$ vertices of order $i + 1$ in ascending order. Thus, $n_i$ vertices of order $i$ have an impact on the order of their parent vertex the order of their parent vertex equivalent to $\lfloor \frac{n_i}{2^k} \rfloor$ vertices of order $i+k$, exactly as in the type II ordering with $s = 2$. Thus, the type II binary decomposition gives the same orders to the vertices of $T$ as the type II ordering with $s = 2$. □

## 3.3 Pruning

**Definition 16.** *Pruning with leaf-restricted series reduction* is a modified pruning process which, given a tree $T$, produces $\mathcal{R}'(T)$ by the following process::

1. Perform leaf-restricted series reduction, eliminating all leaves which are offspring of vertices which have exactly 1 offspring, repeating until no such leaves exist.
2. Eliminate all leaves.

**Lemma 1.** *Given a tree $T$ and a vertex $k \in T$, the number of iterations of pruning necessary to eliminate the subtree rooted at $k$ is equal to the number of iterations of pruning with leaf-restricted series reduction necessary to eliminate $k$.*

*Proof.* This follows from the fact that pruning with leaf-restricted series reduction still performs series reduction, but does so only on the iteration which will eliminate a branch. □

**Theorem 7.** *Given a tree $T$ and a vertex $k \in T$, the number of iterations of pruning necessary to eliminate the subtree rooted at $k$ gives the Type I order of $k$ with $s = 2$.*

*Proof.* Suppose that a vertex is of order 1. Then either it is a leaf or it has only one offspring, which is of order 1. In either case, it will be eliminated in the first iteration of pruning with leaf-restricted series reduction, either through series reduction or through elimination of leaves.

Suppose that a vertex has order 2. Then it has either at least two offspring, all of order 1, or one offspring of order 2 and no offspring of higher order. In the first case, the vertex's children will be eliminated through the elimination of leaves in the first iteration, and the vertex itself will be left untouched, as that step ends the iteration and the vertex the vertex is not a leaf at that point. In the second case, the vertex is part of a branch whose terminal vertex falls into the first case, and since no parent can be eliminated before its offspring in this pruning process, it will not be eliminated in the first iteration of pruning. Now suppose that for some $n \geq 2$, all vertices of order $n$ remain after one iteration of pruning and suppose that a vertex has order $n+1$. Then it has either at least two offspring of order $n$ and no offspring of higher order, or at least one offspring of order $n+1$ and no offspring of higher order. In the first case, it has at least one offspring which is not eliminated in the first iteration, and in the second case, it is contained in a branch whose leafmost vertex falls into the first case. In either case, the vertex will not be eliminated in the first iteration of pruning, again because no parent vertex is eliminated before its offspring in the pruning process.

Thus all offspring of order 1 are eliminated in the first iteration of pruning, and no offspring of order greater than or equal to 1 are eliminated in that first iteration.

Now suppose that $n$ iterations of the pruning process eliminated all vertices of order $n$ and below, and no vertices of higher order. Then, after $n$ iterations of pruning with leaf-restricted series reduction, vertices of order $n+1$ will Have only offspring of order $n+1$. Since each vertex of order $n+1$ can have no more than one offspring of order $n+1$, each vertex of order $n+1$ will have either one offspring, also of order $n+1$, or will have no offspring. In either case, these vertices will be eliminated in the next iteration of pruning, either through series reduction or through elimination of leaves. Suppose that a vertex has order $n+2$. Then after $n$ iterations of pruning it will have either at least two offspring, all of order $n+1$, or at least one offspring of order $n + 2$ and no offspring of higher order. In the first case, the vertex's children will be eliminated through the elimination of leaves in the $n$th iteration (not through series reduction, since their parent has at least two offspring), and the vertex itself will be left untouched, as that step ends the iteration and the vertex the vertex is not a leaf at that point. In the second case, the vertex is part of a branch whose leafmost vertex falls into the first case, and since no parent can be eliminated before its offspring in this pruning process, it will not be eliminated in the first iteration of pruning with leaf-restricted series reduction. Now suppose that for some $N \geq n + 2$, all vertices of order $N$ remain after $n$ iterations of pruning and suppose that a vertex has order $N + 1$. Then it has either at least two offspring of order $N$ and no offspring of higher order, or at least one offspring of order $N + 1$ and no offspring of higher order. In the first case, it has at least one offspring which is not eliminated in the first iteration, and in the second case, it is contained in a branch whose terminal vertex falls into the first case. In either case, the vertex will not be eliminated in the next iteration of pruning with leaf-restricted series reduction, again because no parent vertex is eliminated before its offspring in the pruning process.

Thus, exactly $n$ iterations of pruning with leaf-restricted series reduction are necessary to eliminate a vertex of order $n$ for all $n \in \mathbb{N}$, and therefore the number of iterations of pruning necessary to eliminate the subtree rooted at a vertex $k$ gives the Type I order of $k$ with $s = 2$. □

**Definition 17.** *Pruning with leaf-restricted $q$-branch reduction* is a pruning process which, given a tree $T$, produces $\mathcal{P}'_q(T)$ by the following process:

1. Perform leaf-restricted branch reduction, eliminating all leaves which are offspring of vertices which have fewer than $q$ offspring, repeating until no such leaves exist.

2. Eliminate all leaves.

**Remark 5.** The number of iterations of runing with leaf-restricted $(s-1)$-branch reduction necessary to eliminate a vertex $k$ gives the type I order of $k$ (with parameter $s$).

Ideally, the same result would hold without restricting branch reduction to leaves, as it does with parameter $s = 2$, but it does not, since offspring counts can increase in branch reduction where they cannot in series reduction.

**Definition 18.** Define *p-pruning* ($p \in \mathbb{N}$) to be an operation which, given a tree $T$, produces $\mathcal{R}_p(T)$ by the following process:

1. If this is the first iteration of pruning, prime $T$ by performing series reduction.

2. For each vertex $k$ with offspring which are leaves, let $n_{1,k}$ denote the number of offspring of $k$ which are leaves. Eliminate those offspring and replace them with $\lfloor \frac{n_{1,k}}{p} \rfloor$ offspring.

3. Perform series reduction, eliminating all vertices which are offspring of vertices which have exactly one offspring, repeating until no such vertices remain.

It is readily apparent that $\mathcal{R}_p(T) = \mathcal{R}(T)$ whenever $T$ has bounded offspring number with $n_k \leq p$ for all $k$.

**Theorem 8.** *Given a tree $T$ and a vertex $k \in T$, the number of iterations of $\mathcal{R}_2$ necessary to eliminate the subtree rooted at $k$ gives the type II order of $k$ with parameter $s = 2$.*

*Proof.* Let $T$ be a tree and let $k$ be a vertex in $T$.

Suppose $k$ has order 1 under the type II ordering with $s = 2$. Then the subtree rooted at $k$ is a simple chain, and so the subtree rooted at $k$ will be eliminated in one iteration of $\mathcal{R}_2$.

Now suppose that the theorem holds for vertices of order $n$ or less, for some $n \in \mathbb{N}$ and suppose $k$ has order $m > n$ but has no offspring of order greater than $n$. Then after $n$ iterations of $\mathcal{R}_2$, the subtree rooted at $k$ will be made up of one non-leaf vertex with $n_{n+1} = \lfloor \sum_{i=1}^{n} \frac{n_{i,k}}{2^{n+1-i}} \rfloor$ offspring. If this number is nonzero, these vertices will be replaced by $n_{n+2} = \lfloor \frac{n_{n+1}}{2} \rfloor$ vertices in the next iteration of $\mathcal{R}_2$, which will be replaced by $n_{n+3} = \lfloor \frac{n_{n+2}}{2} \rfloor$ vertices in the following iteration, and so on, until $k$ has no offspring. This will eliminate $k$ in the $m$th iterations of $\mathcal{R}_2$. Now consider any other vertex $v$ which is rootward from $k$, but in the same branch. After $n$ iterations of $\mathcal{R}_2$, the subtree rooted at $v$ will be identical to the subtree rooted at $k$ after $n$ iterations of $\mathcal{R}_2$, and so the subtree rooted at $v$ will be eliminates after $m$ iterations of $\mathcal{R}_2$.

Thus, the number of iterations of $\mathcal{R}_2$ necessary to eliminate the subtree rooted at $k$ gives the type II order of $k$ with parameter $s = 2$. □

**Remark 6.** The number of iterations of $\mathcal{R}_s$ necessary to eliminate the subtree rooted at $k$ does not, in general, give the type II order of $k$ with parameter $s$.

**Definition 19.** Define *p-pruning with q-branch reduction* ($p, q \in \mathbb{N} \cup \{0\}$, $q < p$) to be an operation which, given a tree $T$, produces $\mathcal{R}_{p,q}(T)$ by the following process:

1. If this is the first iteration of pruning, prime $T$ by performing branch reduction.

2. For each vertex $k$ with offspring which are leaves, let $n_{1,k}$ denote the number of offspring of $k$ which are leaves. Eliminate those offspring and replace them with $\lfloor \frac{n_{1,k}}{p} \rfloor$ offspring.

3. Perform branch reduction, eliminating all vertices which are offspring of vertices which have at most $q$ offspring, repeating until no such vertices remain. Begin with rootmost such vertices.

While $\mathcal{R}_{p,q}$ generalizes $\mathcal{R}_p$, and in some cases the number of iterations of $\mathcal{R}_{s,s-1}$ necessary to eliminate a vertex $k$ can give the type II ordering, this, again, does not hold in general, as offspring counts can increase during branch reduction where they cannot in series reduction. However, this issue can be solved by using leaf-restricted branch reduction, as in the type I ordering.

**Definition 20.** Define *p-pruning with leaf-restricted q-branch reduction* ($p, q \in \mathbb{N} \cup \{0\}$, $q < p$) to be an operation which, given a tree $T$, produces $\mathcal{R}'_{p,q}(T)$ by the following process:

1. Perform branch reduction, eliminating all leaves which are offspring of vertices which have at most $q$ offspring, repeating until no such vertices remain.

2. For each vertex $k$ with offspring which are leaves, let $n_{1,k}$ denote the number of offspring of $k$ which are leaves. Eliminate those offspring and replace them with $\lfloor \frac{n_{1,k}}{p} \rfloor$ offspring.

**Remark 7.** Given a tree $T$ and a vertex $k \in T$, the number of iterations of $\mathcal{R}'_{s,s-1}$ necessary to eliminate the subtree rooted at $k$ gives the type II order of $k$ with parameter $s$.

# Chapter 4

# Self-Similarity in Non-Binary Trees

## 4.1 Defining Self-Similarity

The formulations of a Horton law and stochastic self-similarity require no modification from their formulation in the space of binary trees to work effectively in non-binary trees.

**Definition 21.** Define the mean total neighboring branch counts $S_{ij}$ to be the average counts of branches of order $i$ which are offspring of a vertex in a branch of order $j$

While a binary tree's branching structure is completely determined up to branch sequence by side-branch counts, the same is not true in the non-binary case. Under the type I ordering with $s = 2$, a similar result holds, but the length of branches is no longer determined by side-branch counts. Formulation of self-similarity in terms of mean side-branch counts $T_{ij}$ should remain valid in this case, though, as this should not effect properties like mass or Horton law. Side-branch counts are not sufficient for $s > 2$ or for the type II ordering, though.

**Definition 22.** A random tree should be called *self-similar* if $E(S_{ij}) = S_{j-i} = S_k$ for all $i < j$. That is, in a self-similar random tree, $E(S_{ij})$ depends only on the difference $j - i$.

**Proposition 5.** *This formulation of self-similarity holds in a binary tree if and only if self-similarity as defined through side branching holds.*

*Proof.* This follows from the fact that $S_{i,i+1} = T_{i,i+1} + 2$ and $S_{ij} = T_{ij}$ for all $i < j \leq \Omega$ with $j \neq i + 1$ in a binary tree of order $\Omega$. □

**Conjecture.** *Horton law follows from self-similarity in non-binary trees.*

**Proposition 6.** *Let $N_i^1$ and $N_i^2$ denote the number if branches of order $i$ under the type I and type II orderings with parameter $s = 2$, respectively, in a tree $T$. Let $j \in \mathbb{N} \setminus \{1\}$ Then:*

1. $N_1^2 = N_1^1$*. This also holds for* $s > 2$.
2. *If* $N_i^2 = N_i^1$ *for all* $i < j$*, then* $N_j^2 \leq N_j^1$.
3. *If* $N_j^2 < N_j^1$*, then* $N_i^2 > N_i^1$ *for some* $i > j$.

*Proof.* Let $j \in \mathbb{N} \setminus \{1\}$.

1. Suppose a vertex has order 1 under the type I ordering. Then it has at most $s - 1$ offspring, all of order 1, and so will also have order 1 under the type II ordering. Similarly, each vertex which is of order 1 under the type II ordering also has order 1 under the type 1 ordering. Thus, $N_1^2 = N_1^1$.

2. Suppose $N_i^2 = N_i^1$ for all $i < j$. Then each branch which is of order $j$ under the type I ordering is either of order $j$ under the type II ordering or of order greater than $j$ under the type II ordering, and each branch of order $j$ under the type II ordering is also of order $j$ under the type I ordering. Thus, $N_j^2 \leq N_j^1$.

3. Suppose $N_j^2 < N_j^1$. Then some branch which is of order $j$ under the type I ordering is of order $J > j$ under the type II ordering. This implies that there are more branches of order greater than $j$ under the type II ordering than under the type I ordering. Thus, $N_i^2 > N_i^1$ for some $i > j$.

□

2 and 3 do not hold for $s > 2$ because branches are not necessarily simple chains under those orderings.

## 4.2 Numerical Simulations

Using Mathworks Matlab software, non-binary critical Galton-Watson trees with different offspring distributions were generated and branch counts under different orderings were plotted on a logarithmic scale to check for the validity of Horton law under these orderings, which would correlate with approximate loglinearity. These preliminary simulations indicated the potential for Horton law to hold under these orderings. In each analysis, the Horton ratio $R$ was estimated using least-squares regression according to the following model:

$$\log(N_k) = a - bk + \epsilon$$

$$R = 10^{-b}$$

The offspring distributions used and Horton ratios estimated are shown in table 4.1. Figures 4.1-4.3 show plots of branch counts on a logarithmic scale.

Note that these simulations produced the expected result that the highest orders, and therefore the lowest Horton ratios, were produced under the type II ordering with the minimal value of $s$ ($s = 2$).

| Tree | Offspring Distribution | Vertices | Horton Ratio $R$ | |
|---|---|---|---|---|
| | | | Type I | |
| | | | $s = 2$ | $s = 3$ |
| Tree 1 | $p = (.6,\ 0,\ .25,\ .1,\ .05)$ | 448,949 | $R \approx 4.1162$ | |
| Tree 2 | $p = \left(\frac{2}{3},\ 0,\ 0,\ \frac{1}{3}\right)$ | 2,195,532 | $R \approx 4.0945$ | $R \approx 38.2932$ |
| Tree 3 | Geometric$\left(\frac{1}{2}\right)$ | 962,402 | $R \approx 4.2161$ | $R \approx 78.3891$ |
| **Tree** | **Offspring Distribution** | **Vertices** | **Horton Ratio $R$** | |
| | | | Type II | |
| | | | $s = 2$ | $s = 3$ |
| Tree 1 | $p = (.6,\ 0,\ .25,\ .1,\ .05)$ | 448,949 | $R \approx 3.5859$ | |
| Tree 2 | $p = \left(\frac{2}{3},\ 0,\ 0,\ \frac{1}{3}\right)$ | 2,195,532 | | |
| Tree 3 | Geometric$\left(\frac{1}{2}\right)$ | 962,402 | $R \approx 3.4053$ | $R \approx 25.8219$ |

Table 4.1: Tables of trees generated and estimated Horton ratios

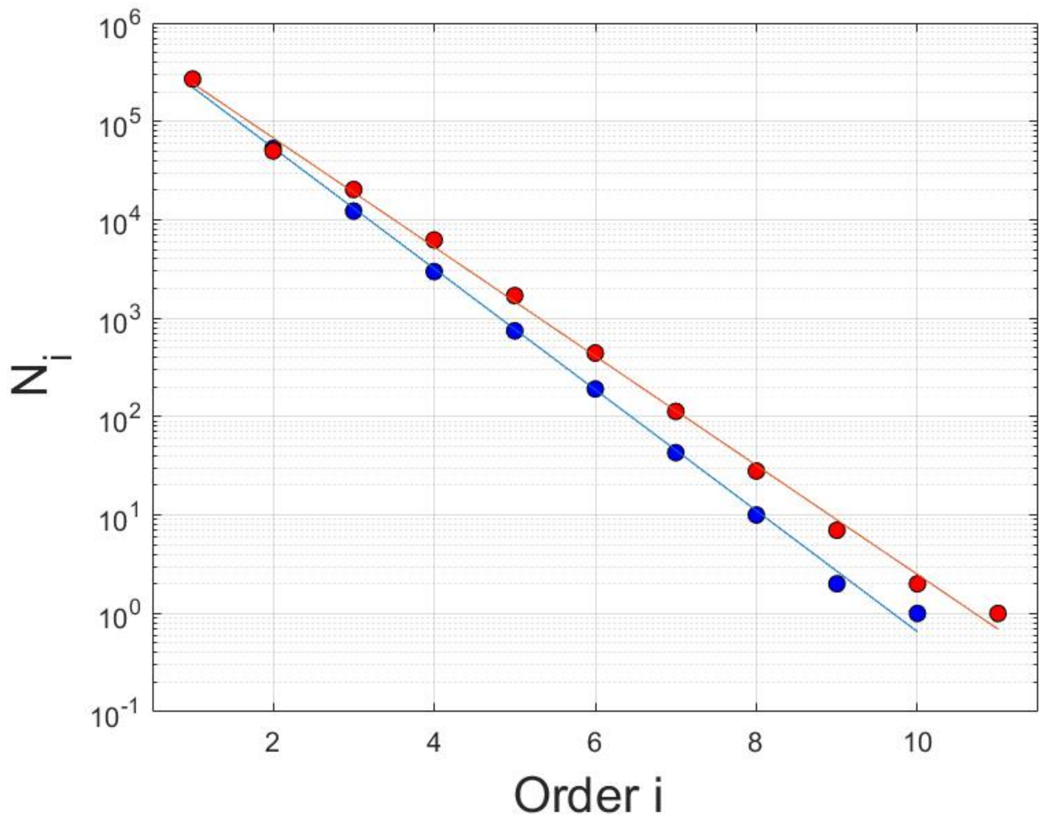


Figure 4.1: Analysis of tree 1, a Galton-Watson tree with offspring distribution $p = (.6,\ 0,\ .25,\ .1,\ .05)$. Blue is type I branch counts with $s = 2$ and red is type II branch counts with $s = 2$. Observe that the tree is of higher order under the type II ordering, but has a lower value for $N_2$.

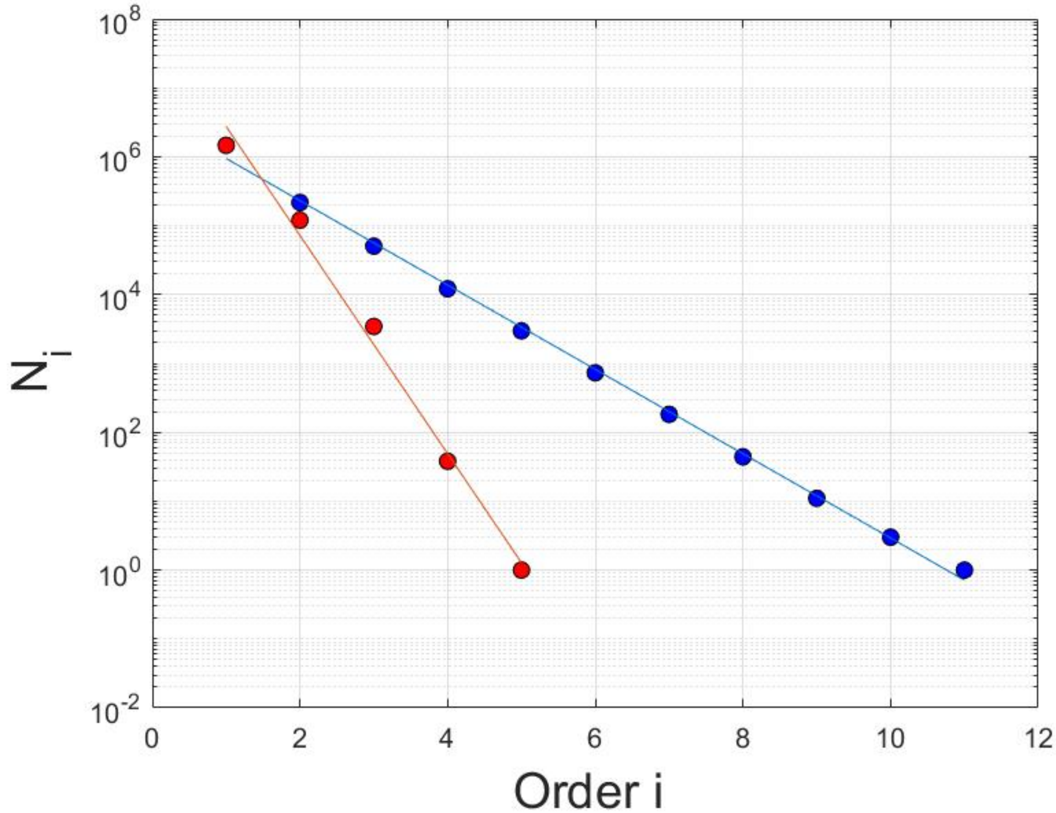


Figure 4.2: Analysis of tree 2, a Galton-Watson tree with offspring distribution $p = \left(\frac{2}{3},\, 0,\, 0,\, \frac{1}{3}\right)$. Blue is type I branch counts with $s = 2$ and red is type I branch counts with $s = 3$. Observe that the tree has much lower order, and therefore much higher Horton ratio $R$, with parameter $s = 3$.

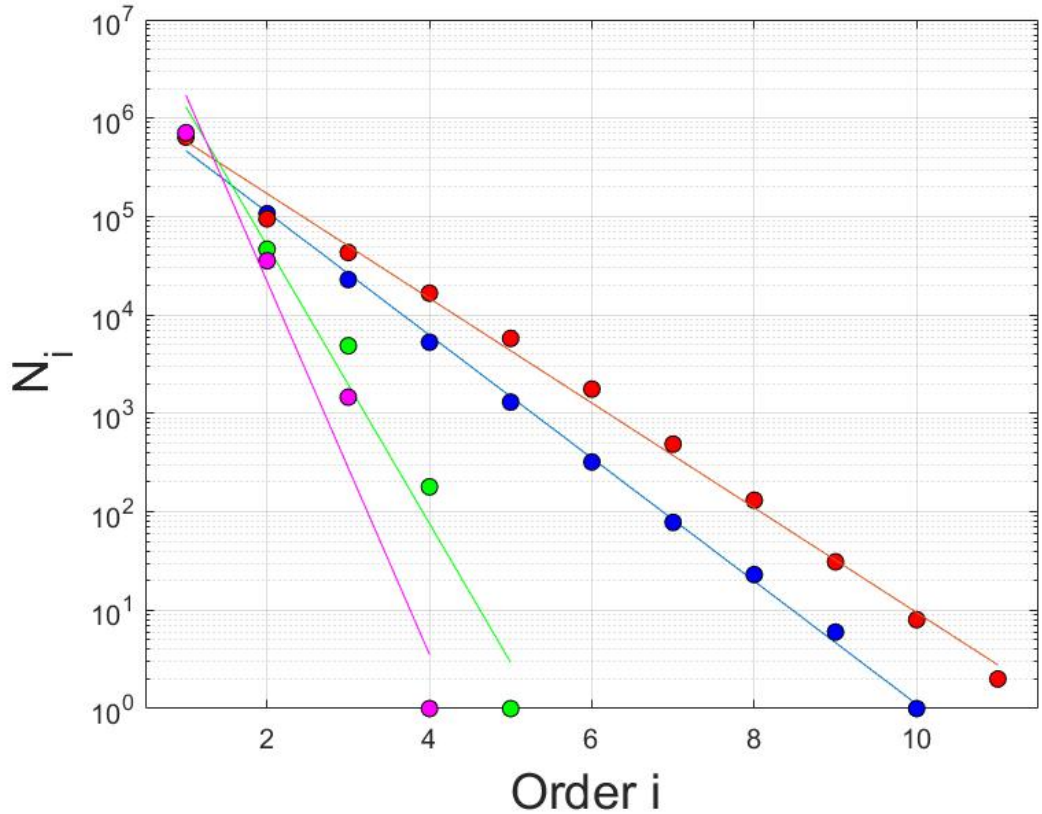


Figure 4.3: Analysis of tree 3, a Galton-Watson tree with geometric critical offspring distribution. Blue is type I branch counts with $s = 2$, red is type II branch counts with $s = 2$, magenta is type I branch counts with $s = 3$, and green is type II branch counts with $s = 3$

# Chapter 5

# Discussion of Results

Preliminary numerical simulations have shown the potential for Horton law under the type I and type II orderings, which we have shown to be direct generalizations of the Horton-Strahler ordering to the space of non-binary trees. Both orderings, especially in the case where $s = 2$, share useful properties of the Horton-Strahler ordering, such as having related pruning processes. In the $s = 2$ case, they can also be directly related to the Horton-Strahler ordering on binary trees through binary decomposition.

The high values of $R$ estimated in numerical simulations under both orderings for $s = 3$ warrant some discussion. Such high values of $R$ indicate that there will be a great amount of variability even in self-similar trees in the exact mass of trees given their order under these orderings. This likely contributed significantly to the drastically different values estimated for $R$ under the type I ordering with $s = 3$, where there was much less variability here under orderings with $s = 2$. The other factor which likely had a significant impact was the relatively low order of the trees. The software had difficulty dealing with larger trees (with millions or more vertices) due to memory and processing constraints, so higher order trees could not be analyzed here. These results indicate that orderings with $s > 2$ have limited application, and should be used only where there is a significant reason that such an ordering is more applicable than an ordering with $s = 2$.

Though to a much lesser degree, the type II orderings tended to produce a lower Horton ratio than type I orderings. This indicates that more information may be given by the order of type II self-similar trees than is given by the order of type I self-similar trees, which is to be expected, as the construction of the ordering uses more information. This indicates that if a tree is self-similar under both orderings, use of the type II ordering may be preferable.

This research leads to a few open questions:

1. Does Horton law necessarily follow from self-similarity in general?

2. How does self-similarity under each ordering proposed here relate to self-similarity under other proposed orderings?

3. What types of random trees are self-similar under the newly proposed orderings? Likely candidates include non-binary Galton-Watson trees and level-set trees for stochastic processes with discrete state spaces.

# References

Burd, G. A., Waymire, E. C., & Winn, R. D. (2000). A Self-Similar Invariance of Critical Binary Galton-Watson Trees. *Bernoulli*, *6*(1), 1-21. doi: 10.2307/3318630

Horton, R. E. (1945). Erosional Development of Streams And Their Drainage Basins; Hydrophysical Approach To Quantitative Morphology. *Geological Society of America Bulletin*, *56*(3), 275-370.

Kovchegov, Y., & Zaliapin, I. (2016). Horton Law in Self-Similar Trees. *Fractals*, *24*(2), 1650017. doi: 10.1142/S0218348X16500171

Kovchegov, Y., & Zaliapin, I. (2017). Horton Self-Similarity of Kingmans Coalescent Tree. *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, *to appear*.

McConnell, M., & Gupta, V. K. (2008). A Proof of the Horton Law of Stream Numbers for the Tokunaga Model of River Networks. *Fractals*, *16*(3), 227-233. doi: 10.1142/S0218348X08003958

Newman, W. I., Turcotte, D. L., & Gabrielov, A. M. (1997). Fractal Trees with Side Branching. *Fractals*, *5*(4), 603-614. doi: 10.1142/S0218348X97000486

Peckham, S. D. (1995). New Results for Self-Similar Trees with Applications to River Networks. *Water Resources Research*, *31*(4), 1023-1029. doi: 10.1029/94WR03155

Phelps, M. (2015). *Analysis of Self-Similarity in Phylogenetic Trees.* Honors

Undergraduate Thesis, University of Nevada Reno.

Shreve, R. L. (1967). Infinite Topologically Random Channel Networks. *The Journal of Geology*, *75*(2), 178-186. doi: 10.1086/627245

Shreve, R. L. (1969). Stream Lengths and Basin Areas in Topologically Random Channel Networks. *The Journal of Geology*, *77*(4), 397-414. doi: 10.1086/628366

Strahler, A. N. (1952). Hypsometric (Area-Altitude) Analysis of Erosional Topography. *Geological Society of America Bulletin*, *63*(11), 1117-1141. doi: 10.1130/0016-7606

Strahler, A. N. (1957). Quantitative Analysis of Watershed Geomorphology. *Transactions, American Geophysical Union*, *38*(6), 913-920. doi: 10.1029/TR038i006p00913

Tokunaga, E. (1978). Consideration on the Composition of Drainage Networks and Their Evolution. *Geographical Reports of Tokyo Metropolitan Univ*, *13*, 1-27.

Zaliapin, I., & Kovchegov, Y. (2012). Tokunaga and Horton Self-Similarity for Level Set Trees of Markov Chains. *Chaos, Solitons and Fractals*, *45*(3), 358-372. doi: 10.1016/j.chaos.2011.11.006

Zanardo, S., Zaliapin, I., & Foufoula-Georgiou, E. (2013). Are American Rivers Tokunaga Self-Similar? New Results on Fluvial Network Topology and Its Climatic Dependence. *Journal of Geophysical Research: Earth Surface*, *118*, 1-18. doi: 10.1029/2012JF002392