On tree hook length formulae, Feynman rules and B-series

by

Bradley Robert Jones

B.Sc., Simon Fraser University, 2012

Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Science

in the

Department of Mathematics

Faculty of Science

© Bradley Robert Jones 2014

SIMON FRASER UNIVERSITY

Fall 2014

APPROVAL

Name: Bradley Robert Jones

Degree: Master of Science (Mathematics)

Title of Thesis: On tree hook length formulae, Feynman rules and B-series

Examining Committee: Dr. Paul Tupper, Associate Professor
Chair

_____

Dr. Karen Yeats, Associate Professor
Senior Supervisor

_____

Dr. Cedric Chauve, Professor
Supervisor

_____

Dr. Tamon Stephen, Associate Professor
Examiner

Date Approved: September 25, 2014

Partial Copyright Licence

SFU

Abstract

This thesis relates similar ideas from enumerative combinatorics, Hopf algebraic quantum field theory and differential analysis. Hook length formulae, from enumerative combinatorics, are equations that can lead to bijections between tree classes and other combinatorial classes. Feynman rules are maps used in quantum field theory to generate integrals from particle interaction diagrams. Here we consider Feynman rules from the Hopf algebra perspective. B-series are powers series that sum over trees and are used in differential analysis to analyze Runge-Kutta methods. The aim of this thesis is to bring together the ideas of the three communities. We show how to use differential equations to obtain new hook length formulae. Some of these new hook length formulae result in new combinatorial bijections. We use hook length formulae to express differential equations combinatorially. We also provide a generalization to hook length. Finally we include a catalogue of known hook length formulae.

To my parents.

"Vix prece finita torpor gravis occupat artus,
mollia cinguntur tenui praecordia libro,
in frondem crines, in ramos bracchia crescunt,
pes modo tam velox pigris radicibus haeret,
ora cacumen habet: remanet nitor unus in illa."

Metamorphoses, OVID

Acknowledgements

Contents

List of Tables

List of Figures

Chapter I

Introduction

There are many cases in the history of mathematics where multiple communities have stumbled upon the same problem and proceeded to solve it in their own ways. Here we consider such a problem appearing in three communities: enumerative combinatorics, quantum field theory and differential analysis. The notions we will compare, and see how they are related to each other, are hook length formulae, Feynman rules and B-series.

The tree factorial, $t!$, for a rooted t is the product of the sizes of the subtrees of t. The reciprocal of the tree factorial is of interest. Knuth [32, § 5.1.4] found, for example, that $\frac{|t|!}{t!}$ counts the number of ways to label a plane tree, t, with increasing labels. $\frac{1}{t!}$ was used as a Feynman rule by Panzer [44, p. 38]. The tree factorial is also used as a statistic in the analysis of Runge-Kutta methods for computing approximate solutions of differential equations [25].

A hook length operator is a generalization of the tree factorial. Hook length operators act on trees and generalize the tree factorial by considering arbitrary functions of the subtrees, instead of just their sizes. We can use a hook length operator to define a formal power series from a hook length operator, B:

$$F_{\mathcal{T},B}(z) = \sum_{t \in \mathcal{T}} B(f) z^{|t|}.$$

This power series is called a hook length series and in the following, we will consider it indexed by a set of trees, $\mathcal{T}$, that form what is known as a simple class of trees. $F_{\mathcal{T},B}(z)$ is thus a weighted generating function as opposed to a counting generating function that enumerates the objects of a combinatorial class.

One interesting application of hook length series, that we will explore in detail, aims at finding functional identities between hook length series and counting generating functions, called hook length formulae. One of the earliest examples of hook length formula was given by Postnikov in 2004 [45]:

$$\sum_{t \in \mathcal{B}_n} n! \prod_{v \in V(t)} \left(1 + \frac{1}{|t_v|}\right) = 2^n (n+1)^{n-1}. \tag{1.0.1}$$

1

The left-hand side of this equation is the coefficient of z^n in a hook length series. The simple tree class is $\mathcal{B}$, the class of binary trees, and the hook length operator is $B(t) = \prod_{v \in V(t)} \left(1 + \frac{1}{|t_v|}\right)$. The right-hand size of the equation is the counting sequence for bicolored labelled forests with n nodes.

A recent line of work in the combinatorial enumeration community is to look for bijective proofs of hook length formulae. When the hook length operator can be used to count specific types of trees — such as the case for the tree factorial counting increasing trees — the identity of a hook length formula induces a bijection between the class of trees counted by the hook length series and the combinatorial class counted by the counting generating function of the hook length formula. For example, Seo [48] found that the hook length operator in Postnikov's formula counts labeled bicolored trees where nodes of one color are labelled increasingly. He developed a bijection from binary trees with this type of labelling to bicolored labelled forests. In Chapter II, we will give numerous examples, some new, of such bijective proofs.

Recently, Kuba and Panholzer [35] discovered an identity of hook length series that induces a recurrence relation on the coefficients of the hook length series.

Theorem (Kuba, Panholzer 2013 [35])**.** *Let $\mathcal{T}$ be a simple tree class and B be a hook length operator. Then for all $k \geq 1$ the hook length series, $F_{\mathcal{T},B}$, satisfies:*

$$B_k = \frac{[z^k]F_{\mathcal{T},B}(z)}{[z^{k-1}]\phi(F_{\mathcal{T},B}(z))}.$$

Here ϕ is a power series which specifies the simple tree class, $\mathcal{T}$. This identity can be used to create hook length formulae. It also categorizes hook length series as the power series that satisfy Kuba and Panholzer's identity. Kuba and Panholzer's identity is equivalent to a differential equation of B-series discovered by Mazza in 2004 [42], which leads us to look at the work of a different community, whose interest is the numerical approximation of differential equations.

B-series, like hook length series, are formal power series indexed by trees. They were developed by Butcher [7] to analyze Runge-Kutta methods to compute numerical solutions to differential equations of the form

$$y'(z) = \phi(y(z)).$$

B-series are power series, denoted $Y_{\phi,a}(z)$, that depend on the ϕ of differential equation and a, a map from trees to reals that is defined from the parameters of a Runge-Kutta method. The B-series $Y_{\phi,1}(z)$ is the solution to the above differential equation. Mazza used hook length operator to produce B-series that are solutions to other differential equations [42].

For a hook length operator we can define an operator, $L_B(1+\theta)$, which is a polynomial of derivatives. This operator leads to a differential equation of B-series.

Theorem (Mazza 2004 [42])**.** *Let B be a hook length operator and $\phi : \mathbb{R} \to \mathbb{R}$. Then the B-series, $F = Y_{\phi,B_!}(z)$, satisfies the differential equation:*

$$F'(z) = L_B(1+\theta)\phi(F(z)).$$

Mazza, without being aware of current development in combinatorics, showed that the B-series, $Y_{\phi,B.!}(z)$ is equal to what we know as the hook length series $F_{\mathcal{T},B}(z)$. In fact, this differential equation of B-series is equivalent to Kuba and Panholzer's recurrence identity. The existence of this differential equation was unknown so far to the combinatorialists investigating hook length formulae. The novelty of this equivalence is that it links the two communities of enumerative combinatorics and differential analysis. It also allows the use of this differential equation to aid in the formation of hook length length formulae.

The last important notion that we study in this thesis comes from a technique called renormalization. Renormalization is a technique in quantum field theory for fixing infinities in computations. In essence it is about "extract[ing] sensible results from apparently ill-defined equations." [30, p. 318] One type of computation that physicists use renormalization for is in Feynman models. Calculations on Feynman models involve applying Feynman rules to Feynman diagrams in order to compute values such as particle trajectories. A Feynman diagram is a graph representing particle interactions and Feynman rules are operators on Feynman diagrams that give probability amplitudes which sum together to give probabilities of particle interactions. See [30, Chapters 6–8].

To aid in the analysis of renormalization, Connes and Kreimer developed a Hopf algebra of rooted trees [13]. In this framework Feynman diagrams are converted into polynomials of trees and Feynman rules are viewed as morphisms from the Connes-Kreimer Hopf algebra to parametrized algebras. One can also work directly with Hopf algebras of Feynman diagrams.

In 2000, [5] Brouder discovered that B-series can be used in the analysis of renormalization. He used these B-series to examine a toy problem of Kreimer. Recently Panzer [44] continued Brouder's analysis and discovered a universal property of the Connes-Kreimer Hopf algebra that relates Feynman rules to hook length operators.

A class of Feynman rules that are of interest to us are Feynman rules that through Panzer's universal property of the Connes-Kreimer Hopf algebra give us hook length operators. We denote each of these operators by L_B^*, where B is a hook length operator.

Using this L_B^* operator, we developed an equivalent statement to Kuba and Panholzer's identity and Mazza's differential equation in the framework of Feynman rules.

Theorem. *Let $\mathcal{T}$ be a simple tree class and B be a hook length operator. Then the hook length series, $F_{\mathcal{T},B}(z)$ satisfies:*

$$F_{\mathcal{T},B}(z) = L_B^*(\phi(F_{\mathcal{T},B}(z))).$$

The importance of the identity in this theorem for mathematical physicists is that it relates hook length formulae to interesting toy models. The simple tree classes and hook length operators can be used to represent Dyson-Schwinger equations and Feynman rules respectively. In this representation hook length formulae correspond to toy models with exact solutions. The identity is also interesting from a combinatorial perspective as it gives a specification of hook length series.

Finally in this thesis we introduce the notion of hook length series of decorated trees. Decorated trees are trees whose nodes have a positive integer size. We extend hook length operators to decorated and

investigate the power series indexed by a simple class of decorated, $\mathcal{T}'$:

$$F_{\mathcal{T}',B}(z) = \sum_{t \in \mathcal{T}'} B(f)z^{|t|}.$$

We call this power series a hook length series of decorated trees.

Simple ordinary tree classes naturally extend to simple decorated tree classes; simple decorated tree classes add control over the size of the nodes. The generalization to decorated trees naturally produces the results from the three communities described above.

Theorem. *Let $\mathcal{T}$ be a simple tree class and B be a hook length operator. Then*

- *for all $k \geq 1$:*

$$B_k = \frac{[z^k]F_{\mathcal{T},B}(z)}{[z^{k-1}]\phi(z, F_{\mathcal{T},B}(z))}.$$

- *$F_{\mathcal{T},B}(z)$ solves the differential equation:*

$$F_{\mathcal{T},B}(z) = L_B(1+\theta)\phi(z, F_{\mathcal{T},B}(z))$$

  and

- *$F_{\mathcal{T},B}(z)$ satisfies:*

$$F_{\mathcal{T},B}(z) = L_B^*(\phi(z, F_{\mathcal{T},B(z)})).$$

Here ϕ is a power series which specifies the simple decorated tree class, $\mathcal{T}$. The ϕ functions for simple decorated tree classes are a generalization of the ϕ functions for simple ordinary tree classes.

The three results of the above theorem are equivalent statements like in the case of ordinary trees. The difference with decorated trees is that decorated trees can have nodes with size greater than 1. This allows us to consider more general tree classes and more kinds of differential equations.

Using decorated tree classes we were able to prove results about more general tree classes and a generalization of the hook length operator. These generalizations enabled us to extend Mazza's differential equation to develop new methods to solve hook length formulae. In particular, these methods allow us to use differential equations with hook length operators that are not rational functions.

We also used our decorated tree classes and Mazza's differential to extend the work of Leroux and Viennot [39, 40] of finding combinatorial solutions to differential equations. Here combinatorial solutions are solutions that arise from weighted generating function of weight combinational classes — in this case hook length series.

This thesis includes new hook length formulae discovered by the author. Six of these hook length formulae are proved by Kuba and Panholzer's recurrence, five are proved using Mazza's differential equation and are eight proved using the new methods developed in this thesis. For two of these hook length formulae we also include bijective proofs. Each of the bijections in the proofs were from the trees classes derived from the hook length series to the combinatorial classes counted in the hook length formulae.

As the final chapter of this thesis we include a catalogue of known hook length formulae. Han's 2008 paper [27] contained most of the known hook length formulae for binary trees and Chen, Gao and Guo [10] had most of the hook length formula for the other tree and forest classes in their 2009 paper. However this catalogue is the most complete resource of known hook length formulae and the only resource to be compiled in a useful catalogue form.

Chapter II

Hook length formulae

In this chapter we begin with an introduction to combinatorial specification. Next we define hook length operators, hook length series and hook length formulae which are the main topic of this work. We also provide an important recurrence relation found by Kuba and Panholzer that can be used to find hook length formulae. We provide some new hook length formulae that were found using the recurrence. We explain how hook length formulae are used to find surprising bijections between combinatorial classes. Finally we generalize hook length to the as yet unconsidered decorated trees and use this generalization to expand Kuba and Panholzer's recurrence.

2.1 Combinatorial classes

This section is an introduction to combinatorial specification based on Flajolet and Sedgewick's book [20]. We will use combinatorial specifiation as a language to define and manipulate combinatorial classes.

A *combinatorial class*, $\mathcal{C}$, is a multiset, $\mathcal{C}$, and a size function, $|\cdot|$, such that for all $n \in \mathbb{N}$, $\mathcal{C}_n = \{c \in \mathcal{C} : |c| = n\}$ is a finite multiset. The members of a combinatorial class are called *combinatorial objects*. Note that a combinatorial class $\mathcal{C}$ can be partitioned as $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$.

An example of a combinatorial class is the class of binary words, $\mathcal{W}$. A binary word is a finite string of 1's and 0's. The size of a binary word is the length of the string. Since there are 2^n strings of 1's and 0's of length n we can see that $\mathcal{W}$ is indeed a combinatorial class.

We say that a map γ between two combinatorial classes, $\mathcal{C}$ and $\mathcal{D}$ is an *(combinatorial) isomorphism* of combinatorial classes if γ is a bijection and $|c| = |\gamma(c)|$ for all $c \in \mathcal{C}$. If such an isomorphism exists we say $\mathcal{C}$ and $\mathcal{D}$ are *(combinatorially) isomorphic* and write $\mathcal{C} \cong \mathcal{D}$. It is important to note that $\mathcal{C} \cong \mathcal{D}$ if and only if $\|\mathcal{C}_n\| = \|\mathcal{D}_n\|$ for all $n \in \mathbb{N}$, where $\|\cdot\|$ denotes cardinality.

For an arbitrary combinatorial class we want to calculate and encode the values $\|\mathcal{C}_n\|$ for all $n \in \mathbb{N}$. To do this we use a structure called a generating function. The *ordinary generating function* of a combinatorial

class, $\mathcal{C}$, is the formal power series:

$$C(z) = \sum_{n \geq 0} \|\mathcal{C}_n\| z^n.$$

By definition two combinatorial classes are isomorphic exactly when they have the same ordinary generating function.

For example, the ordinary generating function for the class of binary words, $\mathcal{W}$, is $W(z) = \sum_{n=0}^{\infty} 2^n z^n$. $W(z)$ is the Taylor expansion of $\frac{1}{1-2z}$ about $z = 0$. Thus we can say that $W(z) = \frac{1}{1-2z}$.

A combinatorial object, c, is *labelled* if it contains $|c|$ elements called atoms each of which is given a unique label from $\{1, \ldots, |c|\}$. A combinatorial class is *labelled* is it contains only labelled objects and *unlabelled* if it contains no labelled objects.

An example of a labelled combinatorial class is the class of permutations, $\mathfrak{S}$. A permutation of size n is word of length n with unique letters from the alphabet $\{1, \ldots, n\}$. Since the number of permutations of size n is $n!$, we see that $\mathfrak{S}$ is indeed a combinatorial class. We can view a permutation as a labelled combinatorial object by identifying each position of the permutation with an atom and labelling the atom with the letter at that position.

For labelled classes it is convenient to use an alternative to an ordinary generating function. The *exponential generating function* of a labelled combinatorial class, $\mathcal{C}$, is the formal power series:

$$C(z) = \sum_{n \geq 0} \|\mathcal{C}_n\| \frac{z^n}{n!}.$$

The exponential generating function of $\mathfrak{S}$ is $S(z) = \sum_{n=0}^{\infty} n! \frac{z^n}{n!} = \frac{1}{1-z}$. Notice that the ordinary generating function of $\mathfrak{S}$ is $\sum_{n=0}^{\infty} n! z^n$ which is not a convergent Taylor expansion.

In general we do not want to have to distinguish between labelled and unlabelled classes, but we still wish to utilize exponential generating functions. To do this we will define the *generating function* of a combinatorial class, $\mathcal{C}$, to be the formal power series:

$$C(z) = \sum_{n \geq 0} \|\mathcal{C}_n\| \zeta^{(n)} \tag{2.1.1}$$

where

$$\zeta^{(n)} = \begin{cases} z^n & \text{if } \mathcal{C} \text{ is an unlabelled classes} \\ \frac{z^n}{n!} & \text{if } \mathcal{C} \text{ is a labelled class.} \end{cases}$$

An m-variate operator Φ taking m combinatorial classes as input and returning a combinatorial class as output is called a *combinatorial operator* if for any 2 lists of m combinatorial classes $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(m)}$ and $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(m)}$ such that $\mathcal{C}^{(1)} \cong \mathcal{D}^{(1)}, \ldots, \mathcal{C}^{(m)} \cong \mathcal{D}^{(m)}$ we have $\Phi(\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(m)}) \cong \Phi(\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(m)})$.

An example of a combinatorial operator is the Cartesian product, $\times$. If we have two unlabelled combinatorial classes, $\mathcal{C}$ and $\mathcal{D}$, and $c \in \mathcal{C}$, $d \in \mathcal{D}$ then we can define the size of $(c, d) \in \mathcal{C} \times \mathcal{D}$ to be $|c| + |d|$. If

7

$\mathcal{A} = \mathcal{C} \times \mathcal{D}$ then its (ordinary) generating function is given by $A(z) = C(z)D(z)$. To see this we expand $A(z)$:

$$\begin{aligned}
A(z) &= \sum_{n \geq 0} \|(\mathcal{C} \times \mathcal{D})_n\| \, z^n \\
&= \sum_{n \geq 0} \|\{(c, d) \in \mathcal{C} \times \mathcal{D} : |c| + |d| = n\}\| \, z^n \\
&= \sum_{n \geq 0} \sum_{k=0}^{n} \|\mathcal{C}_k\| \|\mathcal{D}_{n-k}\| z^n \\
&= \left(\sum_{n \geq 0} \|\mathcal{C}_n\| z^n \right) \left(\sum_{m \geq 0} \|\mathcal{D}_m\| z^m \right) \\
&= C(z)D(z).
\end{aligned}$$

For labelled combinatorial classes we define a slightly different product, $\star$. To define $\star$, we first need to define an operator i_S on labelled combinatorial objects. Let c be a combinatorial object and $S \subset \mathbb{N}^+$ with $\|S\| = |c|$, then $i_S(c)$ is the combinatorial object that is the same as c except for each $i = 1, \ldots, n$ the atom labelled i in c is labelled with the ith greatest element of S in $i_S(c)$.

If $\mathcal{C}$ and $\mathcal{D}$ are labelled classes then we define

$$\mathcal{A} = \mathcal{C} \star \mathcal{D} = \left\{ (i_S(c), i_{\{1, \ldots, |c| + |d|\} \setminus S}(d)) : c \in \mathcal{C}, d \in \mathcal{D}, S \subset \mathbb{N}^+, \|S\| = |c|, \max S \leq |c| + |d| \right\}.$$

The multiset of atoms of $(i_S(c), i_{\{1, \ldots, |c| + |d|\} \setminus S}(d)) \in \mathcal{C} \star \mathcal{D}$ is the disjoint union of the atoms of c and the atoms of d and $|(i_S(c), i_{\{1, \ldots, |c| + |d|\} \setminus S}(d))| = |c| + |d|$. Like for the unlabelled Cartesian product, the (exponential) generating function for $\mathcal{A}$ is given by $A(z) = C(z)D(z)$. The proof of this is similar to the proof for unlabelled classes. We include it here to point out the difference. Expanding $A(z)$ we get

$$\begin{aligned}
A(z) &= \sum_{n \geq 0} \|(\mathcal{C} \star \mathcal{D})_n\| \, z^n \\
&= \sum_{n \geq 0} \left\| \left\{ (i_S(c), i_{\{1, \ldots, |c| + |d|\} \setminus S}(d)) : c \in \mathcal{C}, d \in \mathcal{D}, S \subset \mathbb{N}^+, \|S\| = |c|, \max S \leq |c| + |d| = n \right\} \right\| \frac{z^n}{n!} \\
&= \sum_{n \geq 0} \sum_{k=0}^{n} \|\mathcal{C}_k\| \|\mathcal{D}_{n-k}\| \left\| \left\{ S \subset \mathbb{N}^+ : \|S\| = |c|, \max S \leq n \right\} \right\| \frac{z^n}{n!} \\
&= \sum_{n \geq 0} \sum_{k=0}^{n} \|\mathcal{C}_k\| \|\mathcal{D}_{n-k}\| \frac{n!}{k!(n-k)!} \frac{z^n}{n!} \\
&= \left(\sum_{n \geq 0} \|\mathcal{C}_n\| \frac{z^n}{n!} \right) \left(\sum_{m \geq 0} \|\mathcal{D}_m\| \frac{z^m}{m!} \right) \\
&= C(z)D(z).
\end{aligned}$$

Since the two products have the same effect on the generating functions we shall use $\times$ to denote the product, $\star$, for labelled classes as well.

Type	$\mathcal{C}$	$C(z)$
Elementary class	$\mathcal{C} \cong 1$	$C(z) = 1$
Singleton class	$\mathcal{C} \cong \mathcal{Z}$	$C(z) = z$
Disjoint union	$\mathcal{C} \cong \mathcal{A} + \mathcal{B}$	$C(z) = A(z) + B(z)$
Cartesian product	$\mathcal{C} \cong \mathcal{A} \times \mathcal{B}$	$C(z) = A(z)B(z)$
r copies	$\mathcal{C} \cong r\mathcal{A}$	$C(z) = rA(z)$
r length sequence	$\mathcal{C} \cong \mathcal{A}^r$	$C(z) = A(z)^r$
Sequence	$\mathcal{C} \cong \mathrm{SEQ}(\mathcal{A})$	$C(z) = \frac{1}{1-A(z)}$
Pointing	$\mathcal{C} \cong \Theta(\mathcal{A})$	$C(z) = z\frac{d}{dz}A(z)$
Coefficient extraction	$\mathcal{C} \cong \mathcal{A}_n$	$C(z) = z^n[z^n]A(z)$
Even objects	$\mathcal{C} \cong \mathcal{A}_{\mathrm{even}}$	$C(z) = \frac{A(z)+A(-z)}{2}$
Odd objects	$\mathcal{C} \cong \mathcal{A}_{\mathrm{odd}}$	$C(z) = \frac{A(z)-A(-z)}{2}$
Set (labelled only)	$\mathcal{C} \cong \mathrm{SET}(\mathcal{A})$	$C(z) = e^{A(z)}$
Cycle (labelled only)	$\mathcal{C} \cong \mathrm{CYC}(\mathcal{A})$	$C(z) = \log\left(\frac{1}{1-A(z)}\right)$

Table 2.1.1: Some combinatorial operators and the combinatorial classes 1 and $\mathcal{Z}$.

See Table 2.1.1 for a list of more combinatorial operators.

Two useful combinatorial classes are the elementary class, $1 = \{\mathbb{1}\}$, where $\mathbb{1}$ has size 0 and the singleton class, $\mathcal{Z} = \{\bullet\}$, where $\bullet$ has size 1. $\bullet$ in a labelled class is identified as an atom. Combining these classes with the combinatorial operators of Table 2.1.1 we can build a variety of combinatorial classes. For example the class of binary words can be expressed as $\mathcal{W} \cong \mathrm{SEQ}(2\mathcal{Z})$. We can also express a combinatorial class as the solution of an equation (or system of equations) involving combinatorial operators. This will be used later to define simple tree classes. When a combinatorial class is expressed as a solution of such an equation we call the equation a *(combinatorial) specification* of the class.

A *weighted combinatorial class* is a pair $\mathcal{D} = (\mathcal{C}, \omega)$ of a combinatorial class, $\mathcal{A}$ and a map, $\omega : \mathcal{C} \to \mathbb{K}$, where $\mathbb{K}$ is some ring. We define $\omega(S) = \sum_{s \in S} \omega(S)$ for all $S \subseteq \mathcal{D}$ and define the generating function of $\mathcal{D}$ as:

$$D(z) = \sum_{n \geq 0} \omega(\mathcal{D}_n)\zeta^{(n)}.$$

In this thesis we are primary interested in weighted classes. In particular we are interested in the weighted classes of trees defined in Section 2.2. Using weighted classes allows us to use combinatorial specifications for sets of weighted combinatorial objects.

2.1.1 Simple tree classes

Here we give the definitions for trees and forests. We also define simple tree and simple forest classes.
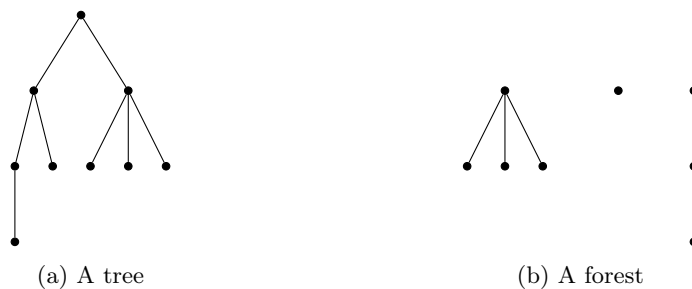
(a) A tree   (b) A forest

Figure 2.1.1: A tree and a forest.

A *(rooted) tree* is a connected acyclic graph with a distinguished vertex called the *root*. We call the vertices of tree, *nodes*. We denote the set of nodes of tree, t, by $V(t)$ and the root by $\text{root}(t)$. We define the size of a tree by $|t| = \|V(t)\|$ and thus we can make combinatorial classes of trees. We can view a tree as a combinatorial object consisting of a node called the root attached to any finite number of rooted trees. In this thesis we will primarily want to view trees this way. We visualize trees as a graph with the root at the top and children drawn below their parents. See Figure 2.1.1a.

A *(rooted) forest* is a finite multiset of rooted trees. We define nodes and size similarly for forests. The class of all forests is denoted $\mathcal{F}$. We can identify a rooted tree, t, with the rooted forest that only contains t. The empty forest, denoted $\mathbb{1}$, is the forest that contains zero trees. We will visualize forests as trees drawn beside each other. See Figure 2.1.1b.

Given a forest f, we say $w \in V(f)$ is a *descendant* of $v \in V(f)$ if there is a path in f from a root to w that contains v. We say w is a *child* of v if w is a descendant of v and w is adjacent to v. Nodes with no children are called *leaves*. The set of leaves of a forest, f, is denoted $\mathscr{D}(f)$. Given a forest f, we say $w \in V(f)$ is an *ancestor* of $v \in V(f)$ if there is a path in f from a root to v that contains w. We say w is a *parent* of v if w is an ancestor of v and w is adjacent to v. Each node in a forest has exactly one parent unless it is a root in which case it has zero parents.

For a rooted forest f and a node $v \in V(f)$ let f_v be the subtree in f whose root is v. f_v contains v and all the descendants of v.

We can label a forest, t, by assigning a unique integer from 1 to $|t|$ to each node. This makes the forest a labelled combinatorial object by associating nodes with atoms. We visualize a labelled forest by writing the label inside the node. See Figure 2.1.2. A labelled node is *increasing* if the label of each descendant of the node is greater than the label of the node. A labelled forest is *increasing* is every node is increasing. See Figure 2.1.2 for an example.

Two forests, f_1 and f_2, are said to have the same *shape* if there is a bijection, $\pi : V(f_1) \to V(f_2)$, such that $v_1 \in V(f_1)$ is a child of $v_2 \in V(f_1)$ in F_1 if and only if $\pi(v_1)$ is a child of $\pi(v_2)$ in f_2. The property of having the same shape defines an equivalence class of forests. Let $\mathfrak{F}$ be this equivalence class and $\mathfrak{T}$ be the equivalence of shapes of trees. For a class of forests, $\mathcal{T}$, let $[f]_{\mathcal{T}}$ be the set of forests with shape f in $\mathcal{T}$. Hook length operators, defined in the next section, are invariant under shape.

10
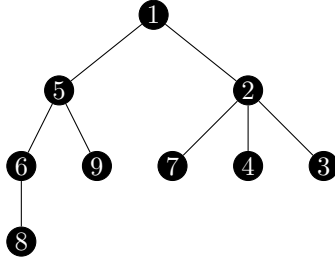
Figure 2.1.2: An increasing tree.
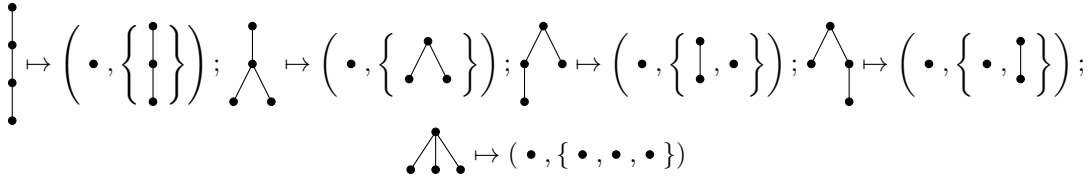
Figure 2.1.3: The plane trees of size 4 and their decompositions.

Now that we have introduced trees and forests we will look at classes of trees and forests. A class of rooted trees, $\mathcal{T}$, is called *simple* if there exists a combinatorial operator, Φ, and an isomorphism $\gamma : \mathcal{T} \to \mathcal{Z} \times \Phi(\mathcal{T})$, called the *decomposition*, such that if $x \in \Phi(\mathcal{T})$ then $x = \{t_1, \ldots, t_j\}$ for some $t_1, \ldots, t_j \in \mathcal{T}$ and $\gamma(t) = (\bullet, \{t_1, \ldots, t_j\})$ if and only if $t_1, \ldots, t_j$ are the subtrees of t whose roots are the children of the root of t.

Most combinatorial operators are isomorphic to an operator of the form of Φ above. Thus we can say that a class of trees is simple if it satisfies the recurrence $\mathcal{T} \cong \mathcal{Z} \times \Psi(\mathcal{T})$ for some operator $\Psi \cong \Phi$ and also has a decomposition.

An example of a class of simple trees is the *class of plane trees*, denoted $\mathcal{O}$. A plane tree is a rooted tree where the sub trees of each node are ordered in a sequence. $\mathcal{O}$ is also the equivalence class of plane embbeddings of rooted trees. The class of plane trees satisfies the equation $\mathcal{O} = \mathcal{Z} \times \mathrm{SEQ}(\mathcal{O})$ showing it is simple. The generating function of $\mathcal{O}$ satisfies $O(z) = \frac{z}{1-O(z)}$ and hence by solving: $O(z) = \frac{1-\sqrt{1-4z}}{2}$. The plane trees of size 4 are shown in Figure 2.1.3.

Remember that from the definition of the size of a tree, the coeffecient of z^n in $\mathcal{O}(z)$ is the number of plane trees with n nodes.

Another simple tree class is the *class of binary trees*, $\mathcal{B}$, introduced in the introduction. Each node in a binary tree has either a left child, a right child, both a left and right child or no children. The class of binary trees satisfies the specification $\mathcal{B} \cong \mathcal{Z} \times (1 + \mathcal{B})^2$ and has the generating function, $B(z) = \frac{1-\sqrt{1-4z}}{2z} - 1$.

Motzkin trees are plane trees where each node has at most two children. They are called Motzkin trees because they are counted by the Motzkin numbers. See [20, §I.5]. The class of Motzkin trees satisfies the specification $\mathcal{M} \cong \mathcal{Z} \times (1 + \mathcal{M} + \mathcal{M}^2)$ and has the generating function, $M(z) = \frac{1-z-\sqrt{1-2z-3z^2}}{2z}$.

r-ary trees are like binary trees except that instead of left and right children there are r types of children. In other words for each node there are r spaces to put children and there are $\binom{r}{m}$ ways for a node to have m children. The class of r-ary trees satisfies the specification $\mathcal{T} \cong \mathcal{Z} \times (1 + \mathcal{T})^r$ and $\|\mathcal{T}_n\| = \frac{1}{n}\binom{rn}{n-1}$.

Complete r-ary trees are plane trees where every node has r children or is a leaf. The class of complete r-ary trees satisfies the specification $\mathcal{T} \cong \mathcal{Z} \times (1 + \mathcal{T}^r)$ and $\|\mathcal{T}_{rm+1}\| = \frac{1}{rm+1}\binom{rm+1}{m}$.

Another important class of trees is the *class of rooted labelled trees*, denoted $\mathcal{R}$. The class of rooted labelled trees satisfies $\mathcal{R} \cong \mathcal{Z} \times \text{SET}(\mathcal{R})$ and thus is simple. The generating function of $\mathcal{R}$ is $\mathbf{W}(z) = \sum_{n \geq 1} \frac{n^{n-1}}{n!} z^n$ and satisfies $\mathbf{W}(z) = ze^{\mathbf{W}(z)}$. We use $\mathbf{W}$ instead of R because $\mathbf{W}$ is related to the Lambert W function, W, which satisfies $z = W(z)e^{W(z)}$. Substituting we can see that $\mathbf{W}(z) = -W(-z)$. See [14].

Another class of rooted labelled trees is the *class of cyclic trees*, $\mathcal{C}$. Cyclic trees are labelled trees where the children of each node are arranged in a cycle. $\mathcal{C}$ satisfies the specification, $\mathcal{C} = \mathcal{Z} \times (1 + \text{CYC}(\mathcal{C}))$. The generating function of $\mathcal{C}$ does not have a nice closed form.

We can define a similar notion for forests. A class of rooted forests, $\mathcal{T}^*$, is called *simple* if there exists a simple tree class $\mathcal{T} = \mathcal{Z} \times \Phi(\mathcal{T})$ and an isomorphism $\gamma : \mathcal{T}^* \to \Phi(\mathcal{T})$ such that $\gamma(f) = \{t_1, \ldots, t_j\}$ if and only if $t_1, \ldots, t_j$ are all the trees contained in f.

Note that if $\mathcal{T}$ is a simple tree class with operator Φ and $\mathcal{T}^*$ is a simple forest class with operator Φ then $\mathcal{T} \cong \mathcal{Z} \times \mathcal{T}^*$. This leads to the specifications: $\mathcal{T} \cong \mathcal{Z} \times \mathcal{T}^*$ and $\mathcal{T}^* \cong \Phi(\mathcal{Z} \times \mathcal{T}^*)$. We will often identify a simple forest class as a solution to the later equation.

One simple class of forests is the *class of rooted labelled forests*, $\mathcal{R}^*$. $\mathcal{R}^*$ satisfies the specifications: $\mathcal{R}^* \cong \text{SET}(\mathcal{R})$ and $\mathcal{R}^* \cong \text{SET}(\mathcal{Z} \times \mathcal{R}^*)$. The generating function of $\mathcal{R}^*$ is $\epsilon(z)$, which is Eisenstein's function that solves the equation $\epsilon(z) = e^{z\epsilon(z)}$ [24, §5.4]. Since $\mathcal{R}$ is the simple tree class associated with $\mathcal{R}^*$ it follows that $\epsilon(z) = \mathbf{W}(z)/z$.

Another simple forest class is the *class of plane forests*, $\mathcal{O}^*$. Here each tree in a forest is a plane tree and the trees of a forest are arranged in a linear order. The class of plane forests satisfies the specifications: $\mathcal{O}^* \cong \text{SEQ}(\mathcal{O})$ and $\mathcal{O}^* \cong \text{SEQ}(\mathcal{Z} \times \mathcal{O}^*)$. The generating function of plane forests is $O^*(z) = \frac{1 - \sqrt{1-4z}}{2z}$, which is the generating function of the Catalan numbers. See [20, p. 6].

One more class of simple forests is the *class of cyclic forests*, $\mathcal{C}^*$. Here each forest is a cycle of cyclic trees. The class of cyclic forests satisfies the specifications: $\mathcal{C}^* \cong 1 + \text{CYC}(\mathcal{C})$ and $\mathcal{C}^* \cong 1 + \text{CYC}(\mathcal{Z} \times \mathcal{C}^*)$. The generating function for cyclic forests does not have a nice closed form.

2.2 Hook length series

In this section we define hook length and hook length series for trees and forests. Then we discuss recent recurrence equations found by Kuba and Panholzer.

We call an operator, $B : \mathcal{F} \to \mathbb{K}$, a *hook length operator* if there exists $B_k \in \mathbb{K}$ for each positive integer, k, such that for all forests, f, $B(f) = \prod_{v \in V(f)} B_{|f_v|}$. Here (and for the rest of the paper) $\mathbb{K}$ is any field,

but we can view $\mathbb{K}$ as either $\mathbb{C}$ or $\mathbb{Q}$.

One property of a hook length operator, B, is that if t is the tree whose root is attached to the trees $t_1, \ldots, t_m$ then $B(t) = B_{|t|} \prod_{i=1}^{m} B(t_i)$ and if f is the forest that contains the trees $t_1, \ldots, t_m$ then $B(f) = \prod_{i=1}^{m} B(t_i)$.

The simplest example of a hook length operator is the tree factorial. The *tree factorial* is defined as $f! = \prod_{v \in V(f)} |f_v|$; here $B_k = k$. The tree factorial is called a factorial because of its recursive expansion. If t is a tree whose root is attached to the trees $t_1, \ldots, t_m$ then $t! = |t| \prod_{i=1}^{m} t_i!$. This is similar to the factorization of the ordinary factorial: $n! = n(n-1)!$.

Another, more useful, hook length operator is $\sigma(f) = \frac{1}{f!}$; here $\sigma_k = \frac{1}{k}$. In his book, Knuth [32, § 5.1.4 Excerise 20] gave an exercise to show that the number of increasing labellings of a plane tree, t, is given by $\frac{|t|!}{t!} = |t|! \sigma(t)$. The proof of this can be given by a simple inductive argument. There is exactly one increasing labelling of a plane tree with one node. If a tree has more than one node then the root is attached to trees, $t_1, \ldots, t_m$. Any increasing labelling of t must have the root labelled 1 and the labels of $t_1, \ldots, t_m$ must be increasing. By induction there are $\frac{|t_i|!}{t_i!}$ increasing labelling of each of these trees. There are $\binom{|t|-1}{|t_1|, \ldots, |t_m|}$ ways to partition the labels to each subtree and thus there are

$$\binom{|t|-1}{|t_1|, \ldots, |t_m|} \prod_{i=1}^{m} \frac{|t_i|!}{t_i!} = \frac{(|t|-1)!}{|t_1|! \cdots |t_m|!} \prod_{i=1}^{m} \frac{|t_i|!}{t_i!}$$

$$= \frac{|t|}{|t|} (|t|-1)! \prod_{i=1}^{m} \frac{1}{t_i!}$$

$$= |t|! \prod_{v \in V(t)} \frac{1}{|t_v|}$$

$$= \frac{|t|!}{t!}$$

increasing labellings of t. Because the multiplicative nature of σ it also follows that the number of increasing labellings of a plane forest, f, is $\frac{|f|!}{f!}$.

Hook length operators are given that name because of the relationship between σ and increasing trees, which is analogous to hook length for partitions that can be used to count standard Young tableaux (increasing partitions) [22].

Another interesting hook length operator is the one given by $B_k = 1 + \frac{1}{k}$. This hook length operator is employed in Postnikov's formula, Equation 1.0.1. Applying this hook length operator to a forest, f, gives the number of labelled trees with shape f that have nodes coloured white or black where white coloured nodes must be increasing.

We shall now combine hook length operators with generating functions to create hook length formulae.

Definition 2.2.1. Let $\mathcal{T}$ be a class of forests and B be a hook length operator. Define $F_{\mathcal{T},B}(z)$ to be the generating function of the weighted combinatorial class $(\mathcal{T}, B)$. In other words:

$$F_{\mathcal{T},B}(z) = \sum_{t \in \mathcal{T}} B(t) \zeta^{(|t|)}.$$

We call $F_{\mathcal{T},B}$ the *tree hook length series* of $\mathcal{T}$ with respect to B.

When B is a general hook length operator we may write $F_{\mathcal{T}}$ instead of $F_{\mathcal{T},B}$.

It is often easier, such as in proofs, to consider the following more general version of the hook length series. We will use the following hook length series to derive the recurrence relation for $F_{\mathcal{T},B}$.

Definition 2.2.2. Given a formal power series $\phi(x) = \sum_{i \geq 0} \phi_i x^i$ define for a forest, f,

$$w_\phi(f) = \prod_{v \in V(f)} \phi_{\deg(v)},$$

where $\deg(v)$ is the number of children of v. We define

$$F_{\phi,B}(z) = \sum_{t \in \mathcal{O}} B(t) w_\phi(t) z^{|t|}.$$

We call $F_{\phi,B}$ the *tree hook length series* of ϕ with respect to B.

$F_{\phi,B}$ is a generalization of $F_{\mathcal{T},B}$ when $\mathcal{T}$ is simple. The following proposition shows this:

Proposition 2.2.3 (Kuba, Panholzer 2013 [35]). *If $\mathcal{T} \cong \mathcal{Z} \times \Phi(\mathcal{T})$ is a simple tree class and ϕ is the power series of Φ then*

$$F_{\mathcal{T},B}(z) = F_{\phi,B}(z).$$

This follows from the fact that B is invariant under shape and the number of trees in $\mathcal{T}$ with the shape of $t \in \mathcal{O}$ is given by $\|[t]_{\mathcal{T}}\| = w_\phi(t)\|[t]_{\mathcal{O}}\|$.

We are primarily interested in hook length formulae. A *hook length formula* is an equation

$$F_{\mathcal{T},B}(z) = g(z)$$

or

$$F_{\phi,B}(z) = g(z)$$

where $\mathcal{T}$ is a tree class, ϕ and g are power series and B is a hook length operator. One reason these formulae are of interest is because they lead to new bijections and combinatorial properties. See Section 2.3.

While studying hook length formulae of binary trees, Han [27] discovered a recurrence relation of hook length series. Kuba and Panholzer extended this relation in 2013.

Theorem 2.2.4 (Kuba, Panholzer 2013 [35]). *Let B be a hook length operator and $\phi(x) = \sum_{i \geq 0} \phi_i x^i$ be a formal power series then for all $k \geq 1$:*

$$B_k = \frac{[z^k]F_{\phi,B}(z)}{[z^{k-1}]\phi(F_{\phi,B}(z))}.$$

Using the fact that $F_\phi = F_{\mathcal{T}}$ when $\mathcal{T}$ is simple, we get the corollary:

Corollary 2.2.5. *Let* $\phi(z) = \sum_{i \geq 0} \phi_i z^i$ *be the formal power series of* $\Phi(\mathcal{Z})$. *If* $\mathcal{T} \cong \mathcal{Z} \times \Phi(\mathcal{T})$ *is a simple tree class then for all* $k \geq 1$:

$$B_k = \frac{[z^k] F_{\mathcal{T}, B}(z)}{[z^{k-1}] \phi(F_{\mathcal{T}, B}(z))}.$$

This recurrence gives a simple method to find hook length formulae given a tree class and a target formula, $F(z)$. Simply apply coefficient extraction to $F(z)$ and $\phi(F(z))$ to obtain a hook length operator. This is not always practical, say if it is difficult to find the coefficient expansion of the composition of ϕ and $F(z)$. This recurrence is also not very useful if we know the hook length operator and the tree class and want to find the hook length series. In Section 4.2 we will present another method using a differential equation for finding hook length formula that is better suited to this situation. The method in Section 4.2 also gives some insight into when nice hook length formulae exist, which the recurrence does not give.

We can also define a similar power series for forests.

Definition 2.2.6. *Let* $\phi(x) = \sum_{i \geq 0} \phi_i x^i$ *be a formal power series. Then we define*

$$G_{\phi, B}(z) = \sum_{t \in \mathcal{O}^*} B(t) w_\phi(t) z^{|t|}.$$

We call $G_{\phi, B}$ the *forest hook length series* of ϕ with respect to B.

As with simple tree classes, if $\mathcal{T}^* \cong \Phi(\mathcal{Z} \times \mathcal{T}^*)$ is a simple forest class then $G_{\phi, B}(z) = F_{\mathcal{T}^*, B}(z)$. Also $G_{\phi, B}(z) = \phi(F_{\phi, B}(z))$.

We also call $G_{\phi, B}(z) = g(z)$ a hook length formula.

This gives the following recurrence.

Theorem 2.2.7 (Kuba, Panholzer 2013 [35]). *Let* B *be a hook length operator and* $\phi(x) = \sum_{i \geq 0} \phi_i x^i$ *be a formal power series with* $\phi_0 \neq 0$ *then for all* $k \geq 1$:

$$B_k = \frac{[z^k] \phi^{-1}(G_{\phi, B}(z))}{[z^{k-1}] G_{\phi, B}(z)}.$$

This theorem comes with an analogous corollary for simple forest classes.

Corollary 2.2.8. *Let* $\phi(z) = \sum_{i \geq 0} \phi_i z^i$ *be the formal power series of* $\Phi(\mathcal{Z})$. *If* $\mathcal{T}^* \cong \Phi(\mathcal{Z} \times \mathcal{T}^*)$ *is a simple forest class then for all* $k \geq 1$:

$$B_k = \frac{[z^k] \phi^{-1}(F_{\mathcal{T}^*, B}(z))}{[z^{k-1}] F_{\mathcal{T}^*, B}(z)}.$$

These two theorems are related to each other in the same way as Theorem 2.2.4 and Corollary 2.2.5, where the theorem is a generalization of the corollary. The difference in lettering between $G_{\phi, B}(z)$ and $F_{\mathcal{T}^*, B}(z)$ is due to notation as the G is needed to distinguish the forest hook length series of a power series from the tree hook length series of a power series, $F_{\phi, B}$; however both $G_{\phi, B}(z)$ and $F_{\mathcal{T}^*, B}(z)$ are hook length series of forest.

2.3 Hook length formulae and isomorphisms

A hook length formula is often an equation involving the generating function, $F_{\mathcal{T},B}$, of a weighted class, $(\mathcal{T}, B)$ and a generating function, G, of another combinatorial class, $\mathcal{G}$. When the hook length operator counts a combinatorial object — for example $\sigma(t)$ counts the number increasing labellings of a plane embedding of t — $F_{\mathcal{T},B}$ is the generating function of the class of those objects, $\mathcal{A}$, with respect to $\mathcal{T}$. The equality of $F_{\mathcal{T},B}$ and G implies that there is an isomorphism between $\mathcal{A}$ and $\mathcal{G}$. The presence of such isomorphisms is not always obvious — as was the case for Postnikov's formula (Equation 1.0.1) — and so the search for such an isomorphism is not started until after the associated hook length formula is found. Here we will investigate some of these isomorphisms as well as some new ones that can be found in Examples 2.3.3, 2.3.4 and 2.3.7.

Another goal of the hook length community is to find other kinds of combinatorial proofs of hook length formulae. Various authors have produced these kinds of proofs [12, 34]. These proofs involve calculating both sides of the equations in different ways. Sagan [47] also came up with probabilistic proofs for some hook length formulae. In this thesis, we are only interested in combinatorial proofs involving bijections as we feel that these are more beautiful.

We will start with four examples using classes of increasing trees. From the discussion in the previous section we can see that $F_{\mathcal{T},\sigma}(z)$ is the exponential generating function of increasing trees with shapes from the tree class $\mathcal{T}$. When this hook length series equals the generating function of another combinatorial class, there is an isomorphism between that class and the class of increasing trees.

Example 2.3.1. Here we present an isomorphism between increasing binary trees and permutations. The generating function for increasing binary tress is the same as the hook length series $F_{\mathcal{B},\sigma}(z)$, where $\mathcal{B}$ is the class of binary trees. We know that $F(z)_{\mathcal{B},\sigma}(z) = \frac{z}{1-z}$, which is the generating function of permutations.

Consider the bijection given by Donaghey [16] that, for $S \subset \mathbb{N}^+$, takes words on the finite alphabet S such that each letter appears exactly once in the word to increasing binary trees of size $\|S\|$ with labels from S. Taking $S = \{1, \dots, n\}$ gives us the isomorphism from permutations to increasing binary trees. Let $w = w_1 w_2 \cdots w_{\|S\|}$ be a word on the finite alphabet S with each letter of S appearing exactly once in w. Let $m = \min S$ and $w_i = m$. Let $l = w_1 \cdots w_{m-1}$ and $r = w_{m+1} \cdots w_{\|S\|}$. Let t be the binary tree whose root is labelled with m and whose children are given as follows. If l is not the empty word then the left child of the root is the image of l from the bijection, otherwise the root has no left child. If r is not the empty word then the right child of the root is the image of r from the bijection, otherwise the root has no right child. Then t is an increasing tree with labels from S. The inverse bijection is obtained by simply reading off the label of the tree in infix order. See Figure 2.3.1 for an example.

Donaghey also investigated the image of the isomorphism when we restrict the class of binary trees. He considered the *class of complete binary trees*, where each node has exactly two children or is a leaf. After applying the isomorphism to this class we obtain half of the class of odd-sized alternating permutations. A permutation, π, is alternating if $\pi(i) < \pi(i+1)$ if and only if $\pi(i+1) > \pi(i+2)$ for all $i = 1, \dots, |\pi| - 2$.

$$7, 11, 10, 12, 5, 1, 6, 4, 9, 2, 8, 3$$

(a) A permutation

$$7, 11, 10, 12, 5 \qquad 1 \qquad 6, 4, 9, 2, 8, 3$$

(b) Split the permutation into the word before 1 and the word after 1

1

$$7, 11, 10, 5 \qquad\qquad 6, 4, 9, 2, 8, 7$$

(c) Connect 1 to the left and right words

(d) Apply the the isomorphism recursively
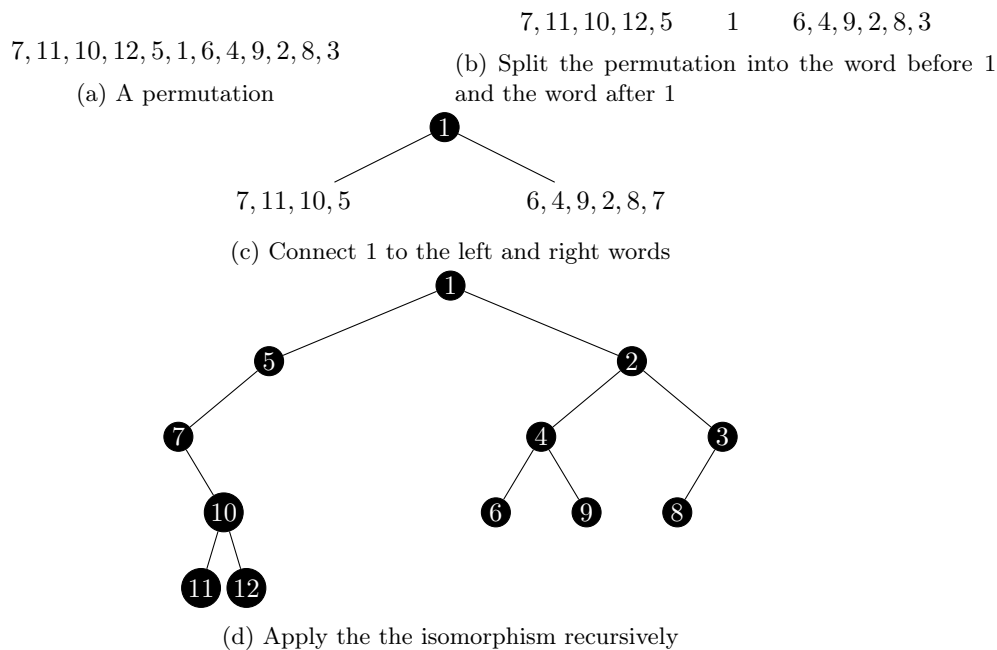
Figure 2.3.1: An example of the isomorphism from permutations to increasing binary trees. Reading off the labels of the resulting tree in infix order we get back the original permutation.

In fact, the bijection gives us the odd-sized alternating permutations with $\pi(1) > \pi(2)$.

We can extend the class of complete binary trees by also including complete binary trees with the rightmost leaf removed. We call this tree class, $\mathcal{B}^{(C)}$, the *class of semicomplete binary trees* as it is an extension of the class of complete binary trees. The class of semicomplete binary trees can be specified with the formula:

$$\mathcal{B}^{(C)} \cong \mathcal{Z} \times (1 + \mathcal{B}^{(C)}_{\text{odd}} \times (1 + \mathcal{B}^{(C)})).$$

This means $\mathcal{B}^{(C)}$ is a simple tree class. This class extends the image of the isomorphism to include all alternating permutations (odd and even sized) that have $\pi(1) > \pi(2)$. These permutations are counted by the Eulerian numbers and their generating function is $\tan z + \sec z - 1$. See [55].

One more interesting restriction is to restrict to the class of binary trees where the right subtree of each node is either a leaf or empty. This class is called the *class of Fibonacci trees* as it is counted by the Fibonacci numbers [52]. Though the class of Fibonacci trees is not simple it satisfies the equation:

$$\mathcal{B}^{(F)} \cong \mathcal{Z} \times (1 + \mathcal{B}^{(F)}) \times (\mathcal{Z} + 1).$$

By applying the isomorphism to increasing Fibonacci trees we do not get an interesting class of permutations; however Stanley [52] gave a different isomorphism mapping the class of increasing Fibonacci trees to the class of involutions.

The isomorphism is as follows. Let t be a Fibonacci tree. For each $v \in V(t)$ let π_v be the identity permutation if v has no right child or the transposition that swaps the label of v with the label of its

(a) A Fibonacci tree

(b) Find the right children

$(19)(25)(3)(4)(5)(67)(7)(8)(9)$

(c) Compose the π_v — here the permutations are written in cyclic notation

$(19)(25)(67) = 953427681$

(d) The resulting transposition
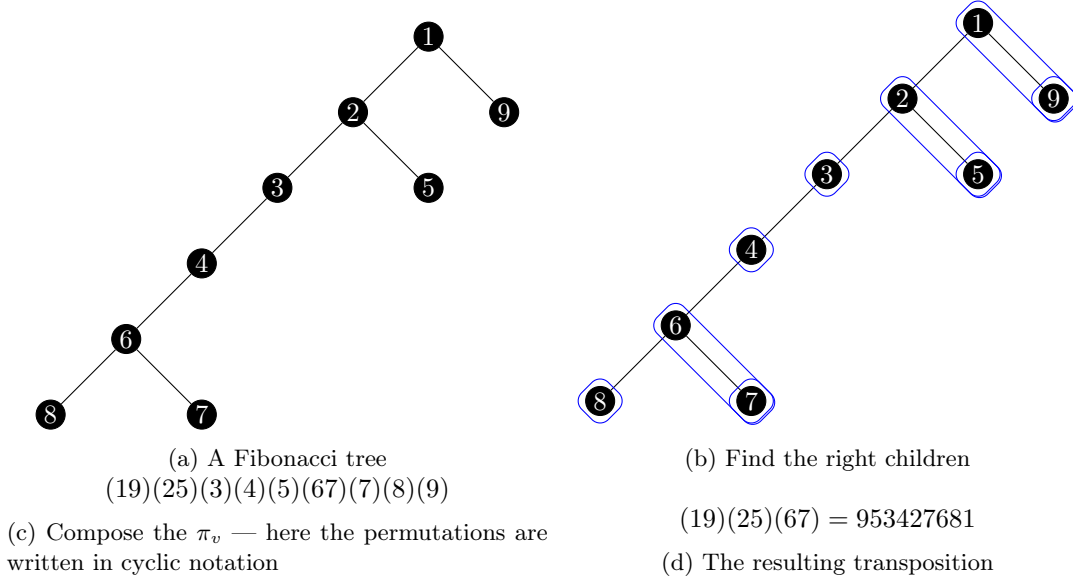
Figure 2.3.2: An example of the isomorphism from increasing Fibonacci trees to involutions. Note that reading off the labels of the tree in infix order gives the permutation 867432519 that is not an involution.

right child if v has a right child. Then $\bigcirc_{v \in V} \pi_v$ is a involution, where $\bigcirc$ denotes iterated composition $(\bigcirc_{i=1}^{n} \phi_i = \phi_1 \circ \phi_2 \circ \cdots \circ \phi_n)$. See Figure 2.3.2 for an example of the isomorphism.

∎

Example 2.3.2. In this example we do not have an isomorphism, but instead have a bijection as the bijection does not preserve size.

Consider the class of increasing plane trees. Chen [9] found that there is a bijection between increasing plane trees with $n + 1$ nodes and matchings on $2n$ vertices. A *matching* is a labelled graph where each vertex is adjacent to exactly one other vertex.

This bijection is ecapsulated in the hook length formula:

$$\sum_{t \in \mathcal{O}_{n+1}} \frac{|t|!}{t!} = (2n-1)!!,$$

since the left-hand side countas increasing plane trees on $n + 1$ nodes and the right-hand side counts the number of matchings on $2n$ nodes.

The bijection is as follows.

If $n = 1$ then we only have the matching of 1 to 2 and the increasing plane tree with root 1 and child 2. So we shall map these to each other.

For $n > 1$ the bijection is as follows. Let M be a matching on $2n$ vertices. Color the vertices of M white if their label is larger than $n + 1$ and view every connected component of M as a rooted tree whose root is the minimum element of that component. Find the connected component, C_1, of M with the

18

smallest label among components with no white vertices (components with no white nodes exist by the pigeonhole principle) and find the connected component, C_2, of M with that the smallest white label. If C_2's root is white then connect the children of the root of C_2 to the right of the children of the root of C_1 and remove C_2 from M (this process is called horizontal merge). If the root, r, of C_2 is not white then either the label of r is less than the label of the root of C_1 or the label of r is greater than the label of the root of C_1. If the label of r is less than label of the root of C_1 then replace the white node with the smallest label in C_2 with C_1 and remove C_1 from M (this process is called vertical merge). If the label of r is greater than label of the root of C_1 then switch the label of the root of C_2 and smallest white label in C_2 and horizontally merge C_1 and C_2 (this process is called lift and merge). Repeat this until there are no white vertices and M will be an increasing plane tree with $n+1$ nodes. See Figure 2.3.3 for a example of this bijection.

The reverse bijection for $n > 1$ is as follows. Let t be an increasing plane trees with $n+1$ nodes. Set $k := n+2$ and set M to be the empty matching. Find the minimum node, v, in t whose left-most child is a leaf. Set $j := k$ and set $k := k+1$. Add v paired with its left-most node to M. For each other child, w, of v from left to right add w paired with k to M and set $k := k+1$. When $n = 2n+1$, M will be a matching with $2n$ vertices.

∎

Example 2.3.3. Consider $\mathcal{R}^*$, the class of labelled unordered forests. Again we know that that the generating function for increasing unordered forests is the same as the hook length series $F_{\mathcal{R}^*,\sigma}(z)$. However in this case the nodes of the forests actually get two labels — one from the labelling of the original forest and the other from the increasing labelling.

Du and Liu [18] found that $F_{\mathcal{R}^*,\sigma}(z) = \frac{1}{1-z}$. This exponential generating function also counts permutations. With two labels, the generating function, $\frac{1}{1-z} = \sum_{n \geq 0} a_n \frac{z^n}{n!n!}$, counts the number of pairs of a permutation and a growing word, which is a word, w, such that for all i, $w_i \leq i$ [10].

The following isomorphism between increasing labelled unordered forest and pairs of a permutation and a growing word is attributed to Thomas by private communication with Chen, Gao and Guo [10]. Let f be an unordered forest where each node, v, of the forest gets a label (a_v, b_v) such that the a_v form a labelling and the b_v form an increasing labelling. Find the node, v, with the label $(a_v, |f|)$. Set $\pi(|f|) = a_v$. If v has a parent then let p be the parent of v and set $w_{|f|} = b_p$. If v has no parent then set $w_{|f|} = |f|$ Repeat the isomorphism on $f \setminus v$. See Figure 2.3.4 for an example.

The inverse isomorphism is as follows. Let π be a permutation of size n and w a growing word of size n. Iterating $i = 1, \ldots, n$ if $w_i = i$ add the tree with a single node labelled $(\pi(i), i)$ to f. If $w_i < i$ then connect a node with label $(\pi(i), i)$ to the node in f whose label is $(\pi(v), w_i)$.

Since the increasing labelling distinguishes each node, the original labelling is unnecessary. In other words changing the original labelling of a bilabelled forest only changes the permutation in the image of Thomas' isomorphism. This implies there is an isomorphism between increasing unordered forests and permutations.

(a) A matching

(b) Color nodes

(c) Horizontally merge 8

(d) Lift and merge 9

(e) Lift and merge 10

(f) Vertically merge 11

(g) Vertically merge 12

(h) Find the node with the least label that has only leaves for children

(i) Relabel node and add children to matching

(j) Find the node with the least label that has only leaves for children

(k) Add children to matching
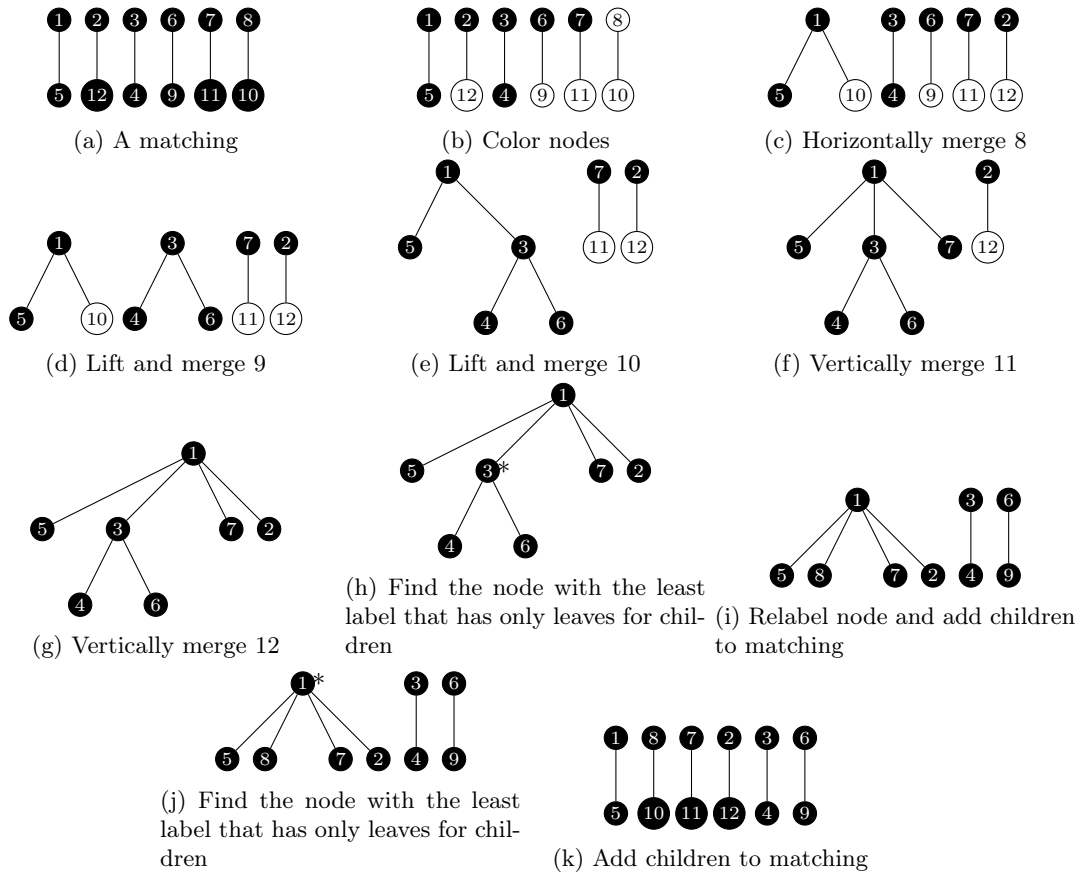
Figure 2.3.3: An example of the bijection from matchings to increasing plane trees. (b)–(g) is the bijection and (h)–(k) is the reverse bijection
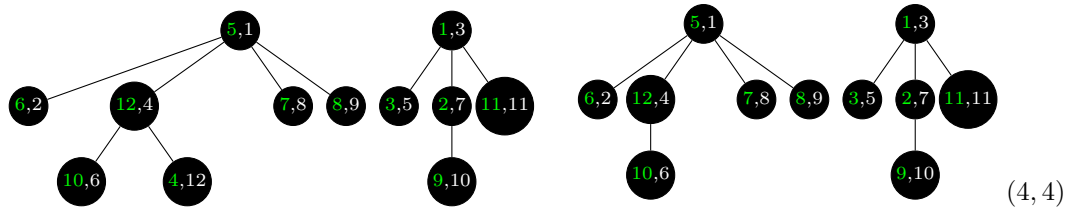
(a) A bilabelled forest with the second labelling increasing

(b) Remove $(4, 12)$ and write 4 paired with its parent's right label

$$((5, 6, 1, 12, 3, 10, 2, 7, 8, 9, 11, 4), (1, 1, 3, 1, 3, 4, 3, 1, 1, 7, 3, 4))$$

(c) Repeat for the remaining forest

$$((6, 1, 12, 3, 10, 2, 7, 8, 9, 11, 4), (1, 3, 1, 3, 4, 3, 1, 1, 7, 3, 4))$$

(d) Make a node with label $(1, 5)$

$$((1, 12, 3, 10, 2, 7, 8, 9, 11, 4), (3, 1, 3, 4, 3, 1, 1, 7, 3, 4))$$
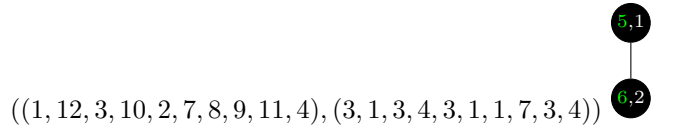
(e) Make a node with label $(6, 2)$ and connect it to the node with label $(\pi_1, 1)$
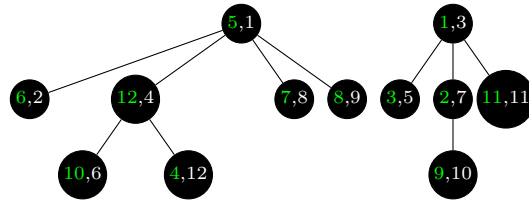
(f) Repeating for each entry gives the original forest

Figure 2.3.4: An example of the isomorphism from bilabelled forest with the second labelling increasing to a permutation/restricted sequence pair. (b)–(c) is the isomorphism and (d)–(f) is the reverse isomorphism.

21

$7, 11, 10, 12, 5, 1, 6, 4, 9, 2, 8, 3$

(a) A permutation

$7, 11, 10, 12, 5 \qquad 1 \qquad 6, 4, 9, 2, 8, 3$

(b) Split the permutation into the word before 1 and the word after 1

$6, 4, 9, 2, 8, 3$

$7, 11, 10, 12, 5$

(c) Make a node with label 1 and connect the left word to that node

(d) Apply the algorithm recursively
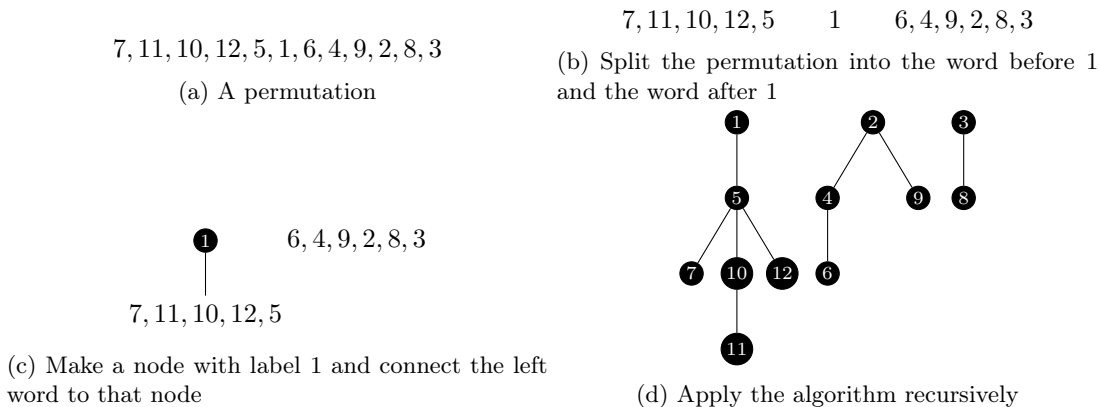
Figure 2.3.5: An example of the isomorphism from permutations to increasing forests. Reading off the labels of the resulting forest in depth-first postfix order gives us the original permutation.

We found a different isomorphism between permutations and increasing unordered forest based on the isomorphism for increasing binary trees. In fact by composing these isomorphisms we get a simple isomorphism between increasing binary trees and increasing unordered forests. Our isomorphism between permutations and increasing unordered forests is as follows. Let w be a finite word with unique letters in $\mathbb{N}^+$. Let w_i be the least letter in w. Let t be the tree whose root is labelled w_i and whose children of the root are the image of $w_1 \cdots w_{i-1}$. Output the increasing forest that is the disjoint union of t and the image of $w_{i+1} \cdots w_{|w|}$. See Figure 2.3.5 for an example.

To reverse the isomorphism simply order the roots of the trees and the children of the nodes in increasing order left to right and then read off the labels in depth-first prefix order.

We will prove that the above map is an isomorphism by showing that the map and the reverse map are inverses. We can see that $\pi = 1$ is mapped to a tree that is a root labelled 1 and vice versa. Let $\pi = \pi_1 \cdots \pi_k 1 \pi_{k+2} \cdots \pi_n$ be a permutation. Then π is mapped to an increasing forest $f = \{t_1, \ldots, t_m\}$ where t_1 has 1 as the root and a forest f_1 as its children. Let π' be the image of f_1 under the reverse map and π'' be the image of $\{t_2, \ldots, t_m\}$ under the reverse map. by induction $\pi' = \pi_1 \cdots \pi_k$ and $\pi'' = \pi_{k+2} \cdots \pi_n$. Therefore the image of f under the reverse map is π. Starting with an increasing forest applying both maps we see that we can also show that we get the original forest back and thus the maps are inverse and hence isomorphisms.

We can also restrict the above isomorphisms to unordered trees. For Thomas' isomorphism of bilabelled forests this would restrict the image to words where $w_i < i$ for all $i > 1$. For our isomorphism of increasing forests the resulting permutations would always end with 1. By linking the front of the permutation to the end of permutation we get unique labelled cycles, see Figure 2.3.6. Thus our isomorphism is also an isomorphism from increasing unordered trees to labelled cycles.

■

Example 2.3.4. A Schröder tree is a rooted tree where the children of each node are arranged in an ordered partition. An ordered partition is a sequence of sets (called *blocks*). The *class of labelled Schröder*
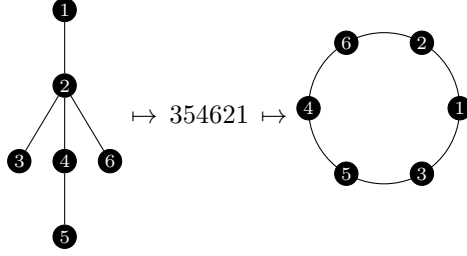
Figure 2.3.6: An example of the isomorphism from increasing plane trees to labelled cycles.

trees, $\mathcal{S}$, is simple and satisfies the specification:

$$\mathcal{S} \cong \mathcal{Z} \times \mathrm{SEQ}(\mathrm{SET}_{\geq 1}(\mathcal{S})).$$

The generating function of increasing Schröder tree is equal to the hook length series $F_{\mathcal{S},\sigma}(z)$. This hook length series is given by:

$$\mathbf{W}\left(\frac{1}{2}\exp\left(\frac{z-1}{2}\right)\right) + \frac{z-1}{2}.$$

This generating function is also the generating function for the class of complete partitions, which is the subject of Schröder fourth problem [53, §6.2].

Chen [9] gave a map that solves Schröder's fourth problem. Here we will extend his bijection to an isomorphism between increasing Schröder trees and phylogenetic trees. *Phylogenetic trees* are trees where each internal node has at least two children and the leaves are labelled. The size of a phylogenetic tree is the number of leaves. The class of phylogenetic trees, $\mathcal{P}$, satisfies the specification:

$$\mathcal{P} \cong \mathcal{Z} + \mathrm{SET}_{\geq 2}(\mathcal{P}).$$

Note that $\mathcal{P}$ is not simple because the $\mathcal{Z}$ is added to, not multiplied by, the combinatorial operator. $\mathcal{P}$ is isomorphic to the class of complete partitions.

Chen's map in [9] maps increasing Schröder trees to forests of increasing trees of height 2. Here we will view these trees as sets of sets instead since it is more intuitive for the completion of the isomorphism. The isomorphism is a composition of two bijections. The first, ϕ, is a variation of Chen's map that takes increasing Schröder trees with n nodes and k blocks to sets of k sets that together have $n + k - 1$ elements. The second, ψ, is a map that takes these sets of sets to phylogenetic trees with n leaves and $k + 1$ internal nodes.

The first bijection, ϕ, is as follows. Let t be an increasing Schröder tree with n nodes and k blocks. Set $S := \emptyset$ and $r := n + k - 1$. First find the maximum labelled node $v \in V(t)$ such that all the children of v are leaves. Let $B_1, \ldots, B_m$ be the labels of the blocks of v. Add the label of v to B_1 and then add B_1 to S. For $i = 2, \ldots, m$, add r to B_i then add B_i to S and set $r = r - 1$. Remove all the children of v from t and relabel v with r. Set $r := r - 1$ and repeat until $r = n$.

The second bijection, ψ, is as follows. Let S be a set of k sets such that $\|\bigcup_{s \in S} s\| = n + k - 1$. Let f be a forest whose nodes are the elements of the sets of S. Mark all the elements of sets in S that are

23

(a) An increasing Schröder tree

(b) Find the node with the greater label whose children are leaves

(c) Extract node and its children

$\{\{10, 12, 15\}; \{7, 16\}; \{13, 22, 23\}; \{4, 6, 14\}; \{2, 3, 11\}; \{1, 8, 20\}; \{5, 9, 18, 21\}; \{17, 19\}\}$

(d) Repeat to get a set of sets

$\{\{10, 12, 15\}; \{7, 16\}; \{13, 22, 23\}; \{4, 6, 14\}; \{2, 3, 11\}; \{1, 8, 20\}; \{5, 9, 18, 21\}; \{17, 19\}\}$

(e) Mark elements

$\{\{7, 16\}; \{13, 22, 23\}; \{4, 6, 14\}; \{2, 3, 11\}; \{1, 8, 20\}; \{5, 9, 18, 21\}; \{17, 19\}\}$

(f) Connect $10, 12, 15$ to 23

(g) Repeat to get a phylogenetic tree
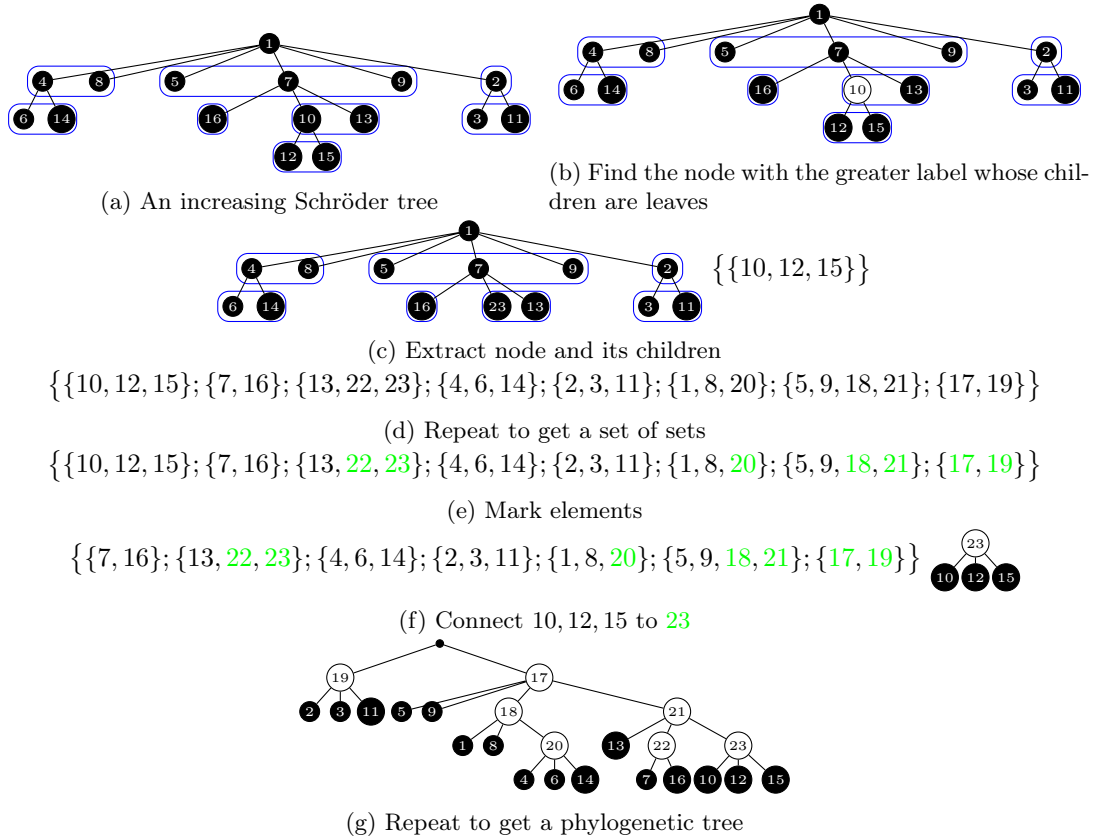
Figure 2.3.7: An example of the isomorphism from increasing Schröder trees to phylogenetic trees. The increasing Schröder tree used in this example was taken from [9].

greater than n. Let v be the greatest marked element among sets in S and let s be the set in S with no marked elements and the greatest label for its minimum element. Connect, in f, each element of s to v so that the elements of s are the children of v. Unmark v, remove s from S and repeat. When there are no marked elements left connect the trees of f to a single root to get a phylogenetic tree.

See Figure 2.3.7 for an example of the isomorphism.

The inverse isomorphism is also the composition of two bijections. The first, ψ^{-1}, is a map from phylogenetic trees with n leaves and $k+1$ internal nodes to sets of k sets that together have $n+k-1$ elements. The second, ϕ^{-1}, is a variation of Chen's map that takes sets of k sets that together have $n+k-1$ elements to increasing Schröder trees with n nodes and k blocks.

The first bijection, ψ^{-1}, is as follows. Let t be a phylogenetic tree with n leaves and $k+1$ internal nodes. Set $r := n+k-1$. Find the unlabelled node, v, whose children are labelled and whose minimum label is greatest. Label v with r and set $r := r-1$. Repeat until every node except the root is labelled. Let c_w be the set of labels of the child of a node $w \in V(t)$. Then $\{c_w : w$ is an internal node of $t\}$ is the desired set of sets.

The second bijection, ϕ^{-1}, is as follows. Let S be a set of k sets such that $\|\bigcup_{s \in S} s\| = n+k-1$. Convert

each set, $s \in S$, into a Schröder tree with the least elements of s as the root and the rest of s as the children of the root all in one block. Mark every node that is labelled greater than n in each tree. Repeat the following process until there are no marked nodes. Find the tree, t_1, with no marked nodes and the maximum labelled root and find the tree, t_2, with the maximum labelled marked node. If t_2 has at least two marked nodes then exchange the label of the root of t_2 with the label of the marked node in t_2 which has the least label. Next replace the marked node which has the largest label in t_2 with t_1 and remove t_1 from the set of trees (this process is called vertical merge). If t_2 has exactly one marked node which is not the root of t_2 then let i be the label of the root of t_2. If i is larger than the label of the root of t_1 then vertically merge t_1 and t_2 and if i is smaller than the label of t_1 then then exchange the label of the root of t_2 and the label of the marked node in t_2 with the least label. Next connect the children of the root of t_2 to the of the children of the root of t_1 preserving blocks and remove t_2 from the set of trees (this process is called horizontal merge). If t_2 has exactly one marked node which is the root of t_2 then horizontally merge t_1 and t_2. The resulting tree is an increasing Schröder tree.

See Figure 2.3.8 for an example of the reverse isomorphism.

The Schröder tree of size one and the phylogenetic tree with one leaf cannot be processed by these maps because they would map to a set of zero sets from ψ and ψ^{-1}. Therefore we extend $\psi \circ \phi$ and $\phi^{-1} \circ \psi^{-1}$ such that the objects of size one map to each other.

We will prove that $\psi \circ \phi$ is an isomorphism by showing that ϕ^{-1} is the inverse of ϕ and ψ^{-1} is the inverse of ψ.

We will prove that ϕ^{-1} is the inverse of ϕ by induction on k.

For $k = 1$ we only consider Schröder trees that are a root whose children are in one block. These types of trees map to sets containing exactly one set and vise versa.

For $k > 1$ suppose t is an increasing Schröder tree with k blocks. Let v be the vertex with the maximum label in t such that such that all the children of v are leaves and P be the partition of the children of v. Let t' be the increasing Schröder that is t with the children of v removed and v labelled with $k + |t| - |P|$. Then by the definition of ϕ, $\phi(t) = \{S_1, \ldots, S_{|P|}\} \cup \phi(t')$ where S_1 is the set containing the labels of the nodes in P_1 and the label of v and for $i > 1$, S_i is the set containing the labels of the nodes in P_i and the integer $k + |t| - i + 1$. Apply the algorithm of ϕ^{-1} to $\phi(t)$ until you need to process $k + |t| - |P| + 1$. At this point you will have the tree, t_v. By the inductive hypothesis $\phi^{-1}(\phi(t')) = t'$. Since when you process $k + |t| - |P| + 1$ replace the greatest element in $\phi(t')$ with t_v, $\phi^{-1}(\phi(t))$ is t' with the greatest labelled node replaced with t_v. Therefore $\phi^{-1}(\phi(t)) = t$.

Let $S = \{S_1, \ldots, S_k\}$ be a set of k sets. Assume that S_1 contains 1 and $S_2, \ldots, S_k$ are decreasing by maximum element. Let S_j be the set in S with the greatest minimum element among subsets of $\{1, \ldots, n\}$. Apply the algorithm of ϕ^{-1} until you need to do an action other than a horizontal merge. At this point you will have a tree, t. Let r be the label of the root of t. The children of the root of t are leaves who are in blocks $S_j \setminus \{r\}$ and $T_1, \ldots, T_l$ for some l and S_j, where T_i is S_i with the largest element removed. Since t is clear (we have only performed horizontal merges), the next action (after

(a) A phylogenetic tree

(b) Label the unlabelled nodes

$$\{\{17, 19\}; \{5, 9, 18, 21\}; \{2, 3, 11\}; \{13, 22, 23\}; \{7, 16\}; \{10, 12, 15\}; \{1, 8, 20\}; \{4, 6, 14\}\}$$

(c) Find the set of sets of siblings

(d) Convert sets to trees and mark nodes

(e) Swap and vertically merge 23

(f) Horizontally merge 22

(g) Swap and vertically merge 21

(h) Vertically merge 20

(i) 'Swap' and vertically merge 19

(j) Horizontally merge 18
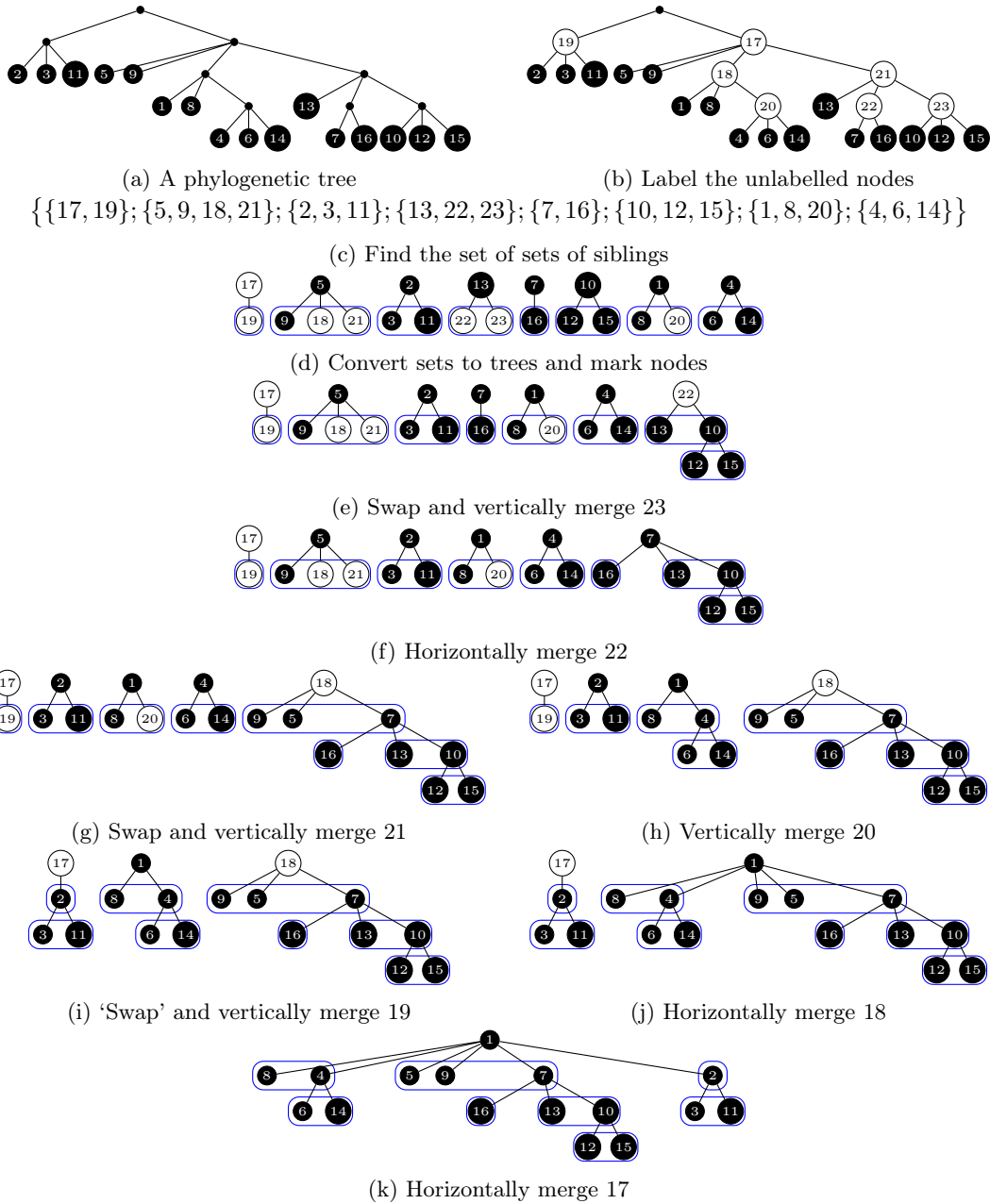
(k) Horizontally merge 17

Figure 2.3.8: An example of the inverse isomorphism from phylogenetic trees to increasing Schröder trees. This example uses the image of the increasing Schröder tree used in Figure 2.3.7.

getting t) of the algorithm is a vertical merge with the largest element of $S \setminus \{S_j, S_1, \ldots, S_l\}$ and t. Let $t' = \phi^{-1}(S \setminus \{S_j, S_1, \ldots, S_l\})$ then $\phi^{-1}(S)$ is t' where the node with the greatest label is replaced with t. Apply the algorithm of ϕ to $\phi^{-1}(S)$ until you add a set containing $k + |t| - l$. At this point you will have $\{S_1, \ldots, S_i, S_j\}$. By the induction hypothesis $\phi(t') = \{S_{l+1}, \ldots, S_k\} \setminus \{S_j\}$. Therefore $\phi\phi^{-1}(S) = \{S_1, \ldots, S_i, S_j\} \cup \{S_{l+1}, \ldots, S_k\} \setminus \{S_j\} = S$.

We will prove that ψ^{-1} is the inverse of ψ by induction on k.

For $k = 1$ we only consider $S = \{s_1\}$ and phylogenetic trees whose only internal vertex is the root. Clearly, $\psi(\{s_1\})$ is the phylogenetic tree that is a root whose children are labelled by s_1 and if t is the phylogenetic tree that is a root with children that are leaves labelled from s_1 then $\psi^{-1}(t) = \{s_1\}$.

For $k > 1$ let $S = \{s_1, \ldots, s_k\}$ where s_1 contains no elements greater than n and whose minimum element is greatest among all such sets. Mark elements according to the isomorphism. Applying the first iteration of the isomorphism, the elements of s_1 are attached as children to a node n_1 in $\bigcup_{i=2}^{k} s_i$ and n_1 is unmarked. Let $t = \psi(S)$ and $t' = \psi(S) \setminus s_1$. We can see that $\psi^{-1}(t') = S \setminus \{s_1\}$ by induction and so $\psi^{-1}(t) = \psi^{-1}(t') \cup \{s_1\} = S$. Therefore $\psi^{-1} \circ \psi = \text{id}$.

Let t be a phylogenetic tree with $k + 1$ internal nodes. Let s_1 be the set of siblings whose minimum is greatest among all sets of siblings. Label the parent of s_1 with $n + k$. Then $\psi^{-1}(t) = \psi^{-1}(t \setminus s_1) \cup \{s_1\}$. By induction $\psi(\psi^{-1}(t \setminus s_1)) = t \setminus s_1$ and so $\psi(\psi^{-1}(t)) = \psi^{-1}(t \setminus s_1) \cup \{s_1\} = t$. Therefore $\psi \circ \psi^{-1} = \text{id}$ and ψ^{-1} is the inverse of ψ.

ϕ is actually a generalization of the bijection in Example 2.3.2. We can view a plane tree as a Schröder tree where each internal vertex is in its own block of size 1. The only difference between ϕ and the bijection in Example 2.3.2 is that in the bijection Example 2.3.2 you process that smallest marked node first and in ϕ^{-1} you process the largest marked node first. The reason we presented the algorithms in these different ways is because it is how Chen first presented the algorithms in [9]. ∎

Now we present three more isomorphisms for a variety of other classes derived from hook length formulae.

Example 2.3.5. Here we will look at a bijective interpretation of Postnikov's hook length formula (Equation 1.0.1) found by Seo [48].

Consider the class, $\mathcal{B}^{(2)}$, of labelled binary trees where each node is colored white or black, and white nodes have increasing labels. In other words the label of a white node is less the labels of each of its descendant The generating function of this class is precisely the hook length series of Postnikov's formula [48]. The right-hand side of Equation 1.0.1 counts the number of bicolored unordered forests with n nodes.

In 2005, Seo [48] found an isomorphism between $\mathcal{B}^{(2)}$ and the class of bicolored unordered forests. Let $t \in \mathcal{B}^{(2)}$. We say a node, v, is *right-minimal* if v has a right subtree and the minimum label of the right subtree of v is less than the minimum label of the left subtree of v or the label of the root. For each right-minimal vertex, v, of t with a nonincreasing label change the color of v to white and switch the left and right subtrees of v. Then return the forest, f, created the following way: For each node, v, in t if v
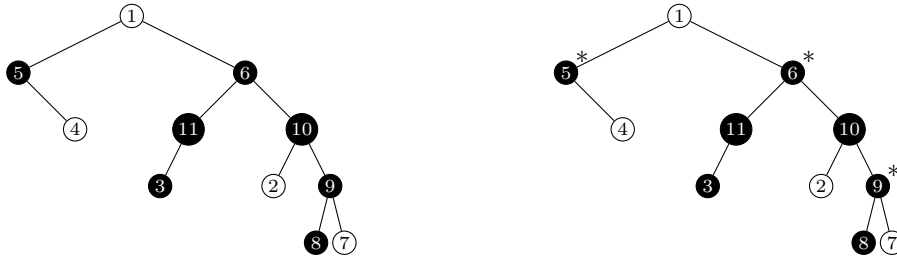
(a) A labelled bicolored binary trees with white nodes increasing

(b) Find nonincreasing nodes that are right minimal
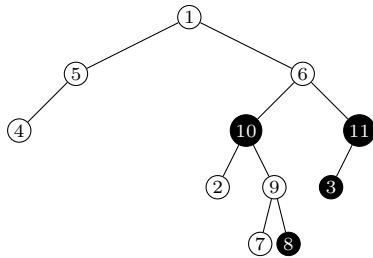
(c) Flip nonincreasing nodes that are right minimal

(d) Interpret the bicolored binary tree as the child-sibling tree of a bicolored plane forest

Figure 2.3.9: An example of the isomorphism from labelled bicolored binary trees with one color increasing to labelled bicolored unordered forests.

has a left child, w, then w is a child of v in f and if v has a right child, x, then x is a sibling of v (i.e. if v is the root of a tree in f then x is the root of another tree and if v is the child of a node y in f then x is a child of y). t is called the sibling-child tree of f. See Figure 2.3.9 for an example of the isomorphism.

The inverse isomorphism is very similar. Let f be a bicolored unordered forest. Order the children of each node so that the value of the minimum label of each subtree is increasing from left to right. Construct the child-sibling binary tree of f. For each nonincreasing white vertex, v, of t change the color of v to black and switch the left and right subtrees of v. Then t is a labelled bicolored binary tree where the white nodes are increasing.

∎

Example 2.3.6. In a follow up paper to their recurrence for hook length formulae (Theorem 2.2.4), Kuba and Panholzer gave combinatorial proofs for new hook length formulae they found [34]. For the hook length formula:

$$\sum_{t \in \mathcal{R}_n} (2n)! \prod_{v \in V(t)} \frac{1}{2|t_v|(2|t_v| - 1)} = n! \left[z^{2n-1}\right] \sqrt{2} \tan\left(\frac{z}{\sqrt{2}}\right),$$

they gave a bijective proof between the class of twice-labelled increasing unordered trees and increasing complete unordered binary trees. In their paper they used increasing complete binary trees with a restricted order for children; we use increasing complete unordered binary trees instead since they are easier to define.

Let $\mathcal{R}^{(2)}$ be the class of rooted unordered trees such that each node has two labels and each label is less

(a) A twice-labelled increasing unordered tree

(b) Shift the labelling

(c) Split tree

(d) Swap labels

(e) Apply the bijection to each tree

(f) Connect the trees to a root with label 1
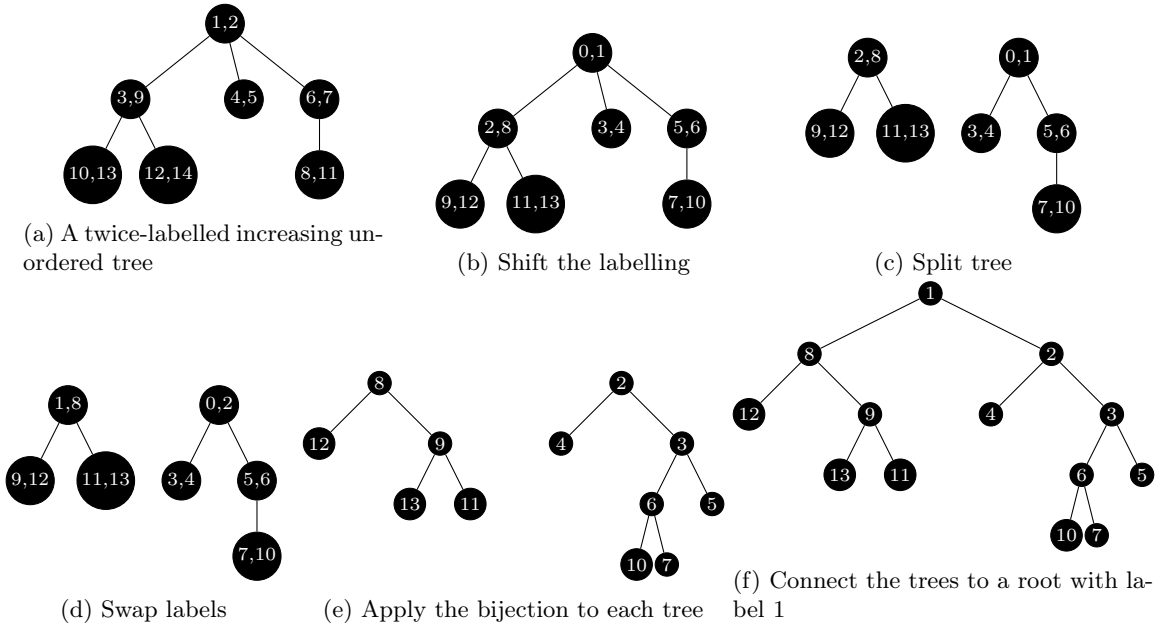
Figure 2.3.10: An example of the bijection from bilabelled increasing unordered trees to increasing unordered complete binary trees. This example was taken from [34].

than all the labels of the descendants of the label's node. We will show that there is a bijection between trees of $\mathcal{R}^{(2)}$ with $2n$ labels and increasing complete unordered binary trees with $2n-1$ nodes.

The bijection is as follows. Let $t \in \mathcal{R}^{(2)}$. Shift the labels of t by subtracting 1 from each label. If t is just a root output a tree with label a_2 where (a_1, a_2) is the label of the root of t and $a_1 < a_2$. Otherwise, let t_1 be the subtree of t with the least label and whose root is a child of the root of t. Let $t_2 = t \setminus t_1$. Let (a_1, a_2) be the label of t_1 and (b_1, b_2) be the label of t_2 so that $a_1 < a_2$ and $b_1 < b_2$. Set the label of t_1 to (b_2, a_2) and set the label of t_2 to (b_1, a_1). Then apply the isomorphism recursively on t_1 and t_2 without shifting labels. Connect the image of t_1 and t_2 as children to a node with label a_2.

■

Example 2.3.7. Let $\mathcal{C}^p$ be the class of unlabelled cyclic trees where each node has an extra edge that points from that node to one of its ancestors if the node is not a leaf or itself if the node is a leaf. This class corresponds to applying the hook length operator given by $B_1 = 1$ and $B_k = k-1$ for all $k > 1$ to the class of unlabelled cyclic trees. We shall see in Example 4.6.3 that the hook length series of labelled cyclic trees with respect to this hook length operator is the same as the ordinary generating function of connected permutations, $F_{\mathbb{C},B}(z) = 1 - \frac{1}{\sum_{n \geq 0} n! z^n}$. A permutation, π, is connected if $\pi(\{1, \ldots, j\}) \neq \{1, \ldots, j\}$ for all $j = 1, \ldots, |\pi| - 1$. As in Example 2.3.3 the equality of the hook length formula of the labelled class leads to an isomorphism for the unlabelled class.

The isomorphism is as follows. First reorder the tree by cycling the children so that the extra edge of each node is pointed to the rightmost subtree of the node. Label the tree with the integers in left-first depth-first prefix order. Let π be the empty word. Then do $(*)$ for the tree.

(∗) If the root of the tree is a leaf then concatenate the label of the root to π. If the root of the tree is not a leaf then concatenate to π the label of the node pointed to by the extra edge of the root. Relabel the node pointed to by the root with 0. Let L be the set of labels of the tree. Relabel the tree with L in left-first depth-first prefix order. Do (∗) for each subtree in left-first order.

We shall now show that the resulting permutation is connected. Let π be the image of the pointed cyclic tree, t. If $|\pi| = 1$ then it is connected; otherwise order the nodes of t as in the isomorphism. Let i be the initial label of the root of the rightmost subtree of the root of t. Since the extra edge of the root of t points to a node in the rightmost subtree, $\pi(1) \geq i$. Suppose $\pi(\{1, \ldots, j\}) = \{1, \ldots, j\}$ for some j. Then $i \leq \pi(1) \leq j$. By the isomorphism $\pi(\{1, \ldots, i-1\}) = \{1, \ldots, i-2, \pi(1)\}$ and thus $\pi(\{i, \ldots, j\}) = \{i-1, \ldots, j\} \setminus \{\pi(1)\}$. By induction the normalization of the rightmost subtree of t is mapped to a connected permutation by the isomorphism and so $j = |\pi|$. Therefore π is connected. See Figure 2.3.11 for an example of the isomorphism.

The reverse isomorphism is as follows. Let π be a connected permutation. Let the positions of π be the nodes of a forest t. Then do (∗∗) for $i = 1$, $j = |\pi|$.

(∗∗) Add an extra edge from node i to node $\pi(i) - i + 1$. Let $c_1, \ldots c_k$ be the connected components of the subword of π from position $i+1$ to position j. For each $a = 1, \ldots, k$ let s_a (respectively e_a) be the starting (respectively ending) position of c_a. Add an edge from i to the start of e_a. Arrange these edges cyclically. Then normalize c_a in π and do (∗∗) for $i = s_a$, $j = e_a$.

The result is a cyclic pointed tree.

By the recursive construction of the isomorphism and the reverse isomorphism, it is simple to see that they are inverses of each other and thus are truly isomorphisms. ∎

Thus we have seen examples of hook length formulae producing combinatorial bijections. All the bijections in this section use the recursive decomposition of simple trees. The use of recursion is overt in the first isomorphism in Example 2.3.1, the second isomorphism in Example 2.3.3 and the isomorphisms in Examples 2.3.6 and 2.3.7. The other bijections are expressed iteratively, but each can be expressed recursively. This is most likely due to the recursive nature of the tree classes.

We discussed earlier how specific isomorphisms in Examples 2.3.1 and 2.3.3 are very similar and how half of the isomorphism in Example 2.3.4 is a generalization of the bijection in Example 2.3.2. However, most of the bijections are ad hoc and do not generalize.

In all of the new isomorphisms in this section, Examples 2.3.3, 2.3.4 and 2.3.7, the hook length series we obtain represented a combinatorial classes with objects that have more sets of labels than the represented classes in the isomorphisms We used these 'unlabelled' classes because it would be redundant to use the labelled versions. Suppose we have two combinatorial classes $\mathcal{C}$ and $l(\mathcal{C})$ where $l(\mathcal{C})$ is the class of labellings of objects of $\mathcal{C}$. Unlabelling can be done when the number of labellings of each objects in $\mathcal{C}$ of size n is $n!$. In this case the ordinary generating function of $\mathcal{C}$ is equal to the exponential generating of $l(\mathcal{C})$. The main reason that the the forests in our examples can be unlabelled is that each of the classes we considered has a unique plane embedding and for each plane tree, t, there are $|t|!$ labellings of

(a) A pointed cyclic tree — we omit drawing the extra edge on the leaves to avoid clutter

(b) Reorder tree by cycling children

(c) Label nodes

(d) Write the label of the root and relabel nodes

(e) Split into sub trees

$8; 2, 4, 1, 3; 5; 13, 6, 9, 7, 12, 10, 11$

(f) Apply the isomorphism to each subtree

(g) Add an extra edge pointing position 1 to position $\pi(1)$

(h) Add edges connecting position 1 to start of connected components

(i) Normalize connected components

(j) Apply isomorphism to each connected component

(k) Redraw the tree vertically to get the original pointed cyclic tree

Figure 2.3.11: An example of the isomorphism from pointed cyclic trees to connected permutations. (b)–(f) is the isomorphism and (g)–(k) is the inverse isomorphism.

t. There are more bijections and isomorphisms involving hook length formulae in the literature. Yang [56] found an involution on increasing plane trees that helps to calculate the hook length formula with $B_k = \frac{(-1)^k}{k}$. Gessel and Seo [23] developed a reverse Prüffer algorithm that acts on a variety of tree classes, particularly r-ary trees, and hook length series of the form $[z^n]F(z) = \frac{c}{n!}\prod_{i=0}^{n-1}(ia+(n-i)b+c)$ for constants a, b, c.

2.4 Some new hook length formulae

Now we will present six new hook length formulae. These hook length formulae were found using the Kuba-Panholzer recurrence and could not be proved using the new methods in Chapter IV because their hook length operators involve factorials or they are hook length formulae of forests.

The first two hook length formulae have hook length series that are $\frac{z}{1-z}$ and $\frac{z}{1-z^r}$. These are generating functions of classes with at most one element of each size. Thus the hook length operators that result in these kinds of hook length series weight the trees of the trees class so that tghe add up to 1. Thus the hook length operator behaves like a probability over the trees of a fixed size. I. Despite this possible application to probability these hook length series have been largely ignored by the hook length community, only appearing for four tree and forest classes. This lack of intereest is probably due to the the fact that it is unknown what these probabilities represent.

The reference to a formula at the top of each example, in this section and in the rest of the thesis, is a reference to the formula number (the **No.** column) in the catalog of hook length formulae of Chapter VI for the hook length formulae featured in the example.

Example 2.4.1 (Formula 6.7.8). Consider $\mathcal{T} \cong \mathcal{Z} \times (1+\mathcal{T})^r$, the class of r-ary trees. Let $B_k = \frac{(r-1)!(k-1)!}{(k+r-2)!}$ and $F(z) = \frac{z}{1-z}$. Then for $k \geq 1$

$$\frac{[z^k]F(z)}{[z^{k-1}](1+F(z))^r} = \frac{1}{[z^{k-1}](\frac{1}{1-z})^r}$$
$$= \frac{1}{(-1)^{k-1}\binom{-r}{k-1}}$$
$$= \frac{(r-1)!(k-1)!}{(k+r-2)!}$$
$$= B_k.$$

Therefore by Corollary 2.2.5 we have that $F(z) = F_{\mathcal{T},B}$ and so for all $n \geq 1$

$$\sum_{t \in \mathcal{T}_n}(r-1)!\prod_{v \in V(t)}\frac{1}{\prod_{i=2}^r(|t_v|+r-i)} = 1. \tag{2.4.1}$$

∎

Example 2.4.2 (Formula 6.8.1). Consider $\mathcal{T} \cong \mathcal{Z} \times (1 + \mathcal{T}^r)$, the class of complete r-ary trees. Let

$$
B_k = \begin{cases} 1 & \text{if } k = 1 \\ \dfrac{(r-1)! \, r^{r-1}}{\prod_{i=0}^{r-2}(k+ir-1)} & \text{if } k > 1 \end{cases}
$$

and $F(z) = \frac{z}{1-z^r}$. For $k = 1$ we have $\frac{[z^1]F(z)}{[z^0](1+F(z)^r)} = 1 = B_1$ and for $k = jr + 1 \geq 2$ we have

$$
\begin{aligned}
\frac{[z^k]F(z)}{[z^{k-1}](1+F(z)^r)} &= \frac{1}{[z^{k-1}]\left(1 + \frac{z^r}{(1-z^r)^r}\right)} \\
&= \frac{1}{[z^{jr}]\left(1 + \frac{z^r}{(1-z^r)^r}\right)} \\
&= \frac{1}{[z^j]\left(1 + \frac{z}{(1-z)^r}\right)} \\
&= \frac{1}{(-1)^{i-2}\binom{-r}{j-2}} \\
&= \frac{(r-1)!}{\prod_{i=0}^{r-2}(j+i)!} \\
&= \frac{(r-1)!}{\prod_{i=0}^{r-2}(\frac{k-1}{r}+i)!} \\
&= \frac{(r-1)! \, r^{r-1}}{\prod_{i=0}^{r-2}(k-1+ir)!} \\
&= B_k.
\end{aligned}
$$

If $k \neq jr + 1$ for all $j \in \mathbb{N}$ then $[z^k]F(z) = [z^k]F_{\mathcal{T},B}(z) = 0$. Therefore by Corollary 2.2.5 we have that $F(z) = F_{\mathcal{T},B}$ and so for all $n = jr + 1 \geq 1$

$$
\sum_{t \in \mathcal{T}_n}(r-1)! \, r^{r-1} \prod_{v \in V(t)\backslash\varnothing(t)} \frac{1}{\prod_{i=0}^{r-2}(|t_v| + ir - 1)} = 1. \tag{2.4.2}
$$

∎

Example 2.4.3 (Formula 6.2.17). Consider $\mathcal{B} \cong \mathcal{Z} \times (1+\mathcal{B})^2$, the class of binary trees. Let $B_k = \frac{((k-1)!)^2}{(2k-1)!}$. Note that B_k is the Beta function, $B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$, with $a = b = k$ [15]. Let $F(z) = \sum_{n=1}^{\infty} \frac{2^n}{(n+1)!n!} z^n$.

For $k = 1$ we have $\frac{[z^1]F(z)}{[z^0](1+F(z))^2} = 1 = B_1$ and for $k \geq 2$ we have:

$$\frac{[z^k]F(z)}{[z^{k-1}](1+F(z))^2} = \frac{[z^k]F(z)}{[z^{k-1}](F(z)^2 + 2F(z))}$$

$$= \frac{\frac{2^k}{(k+1)!k!}}{\sum_{i=1}^{k-2} \frac{2^i}{(i+1)!i!} \frac{2^{k-i-1}}{(k-i)!(k-i-1)!} + \frac{2^k}{k!(k-1)!}}$$

$$= \frac{\frac{2}{(k+1)!k!}}{\sum_{i=1}^{k-2} \frac{1}{(i+1)!i!(k-i)!(k-i-1)!} + \frac{2}{k!(k-1)!}}$$

$$= \frac{\frac{2}{(k+1)!k!}}{\sum_{i=0}^{k-1} \frac{1}{(i+1)!i!(k-i)!(k-i-1)!}}$$

$$= \frac{2}{(k+1)\sum_{i=0}^{k-1} \binom{k}{i+1}\binom{k}{k-1-i}}$$

$$= \frac{2}{(k+1)\binom{2k}{k-1}}$$

$$= \frac{((k-1)!)^2}{(2k-1)!}$$

$$= B_k.$$

Therefore by Corollary 2.2.5 we have that $F(z) = F_{\mathcal{B},B}$ and so for all $n \geq 1$

$$\sum_{t \in \mathcal{B}_n} \frac{(2n)!}{2^n} \prod_{v \in V(t)} \frac{((|t_v| - 1)!)^2}{(2|t_v| - 1)!} = \frac{1}{n+1}\binom{2n}{n}. \tag{2.4.3}$$

Note that $(2n)![z^n]F(z)$ is the number of walks in the quarter plane using the steps $\{(-1,-1), (-1,0), (1,1)\}$ that end on the vertical axis [4]. $\blacksquare$

Example 2.4.4 (Formula 6.11.5)**.** Consider $\mathcal{R} \cong \mathcal{Z} \times \text{SET}(\mathcal{R})$, the class of rooted labelled unordered trees. Let $B_k = \frac{1}{k\binom{k+a-2}{a-1}}$. Let $F(z) = a\log\left(\frac{a}{a-z}\right)$. For $k \geq 1$ we have:

$$\frac{[z^k]F(z)}{[z^{k-1}]e^{F(z)}} = \frac{a[z^k]\log\left(\frac{1}{1-z/a}\right)}{[z^{k-1}]\left(\frac{1}{1-z/a}\right)^a}$$

$$= \frac{a^{1-k}\frac{1}{k}}{(-a)^{1-k}\binom{-a}{k-1}}$$

$$= \frac{1}{k\binom{k+a-2}{a-1}}$$

$$= B_k.$$

Therefore by Corollary 2.2.5 we have that $F(z) = F_{\mathcal{R},B}$ and so for all $n \geq 1$

$$\sum_{t \in \mathcal{R}_n} \frac{a^{n-1}}{n!} \prod_{v \in V(t)} \frac{1}{|t_v|\binom{|t_v|+a-2}{a-1}} = (n-1)!. \tag{2.4.4}$$

This formula simultaneously generalizes Mazza's formula with $B_k = \frac{1}{k^2}$ (Formula 6.11.2) and the classic σ formula (Formula 6.11.1) for rooted labelled unordered trees. We should also note that this follows

from Chen, Gao and Guo's formula for unordered labelled forest (Formula 6.19.6) since $G_{\mathcal{R}^*,B}(z) = \exp(F_{\mathcal{R},B}(z))$. ∎

The next two hook length formulae are forest hook length formulae that result in the hook length series $\frac{1}{1-z}$.

Example 2.4.5 (Formula 6.18.11). Consider $\mathcal{O}^* \cong \text{SEQ}(\mathcal{Z} \times \mathcal{O}^*)$, the class of plane forests. Suppose we want to find the hook length operator, B, such that $F_{\mathcal{O}^*,B}(z) = \frac{1}{1-z}$. By Corollary 2.2.8 we have that:

$$
\begin{aligned}
B_k &= \frac{[z^k]\left(1 - {}^1/F_{\mathcal{O}^*,B}(z)\right)}{[z^{k-1}]F_{\mathcal{O}^*,B}(z)} \\
&= \frac{[z^k]z}{1} \\
&= \begin{cases} 1 & \text{if } k = 1 \\ 0 & \text{if } k > 1. \end{cases}
\end{aligned}
$$

This hook length operator returns 1 if the forest contains only trees that are singletons and 0 otherwise. The resulting hook length formulae is not surprising or deep since the class of plane forests whose trees are only singletons is clearly isomorphic to the class of sequences. However it is important to note that this is the only hook length operator that will give this hook length series for this forest class (as is the case for any hook length operator and hook length series). ∎

Example 2.4.6 (Formula 6.20.2). Consider $\mathcal{C}^* \cong 1 + \text{CYC}(\mathcal{Z} \times \mathcal{C}^*)$, the class of cyclic forests. Suppose we want to find the hook length operator, B, such that $F_{\mathcal{C}^*,B}(z) = \frac{1}{1-z}$. By Corollary 2.2.8 we have that:

$$
\begin{aligned}
B_k &= \frac{[z^k]\left(1 - e^{1-F_{\mathcal{C}^*,B}(z)}\right)}{[z^{k-1}]F_{\mathcal{C}^*,B}(z)} \\
&= \frac{[z^k]\left(1 - e^{-\frac{z}{1-z}}\right)}{1} \\
&= \sum_{i=1}^{k} \frac{(\text{-}1)^{i-1}}{i!}\binom{k-1}{i-1}.
\end{aligned}
$$

Therefore we have:

$$
\sum_{f \in \mathcal{C}_n^*} \prod_{v \in f} \left(\sum_{i=1}^{|f_v|} \frac{(\text{-}1)^{i-1}}{i!}\binom{k-1}{i-1} \right) = 1. \tag{2.4.5}
$$

∎

2.5 Decorated trees

In this section we will discuss decorated trees. This is the first time hook length has been generalized for decorated trees. We do so here as it seems a natural progression to extend Han's recurrence for binary

$$\left|\cdot\right| \mapsto \left(\bullet,\left(\mathbb{1},\left\{\left|\cdot\right|\right\}\right)\right); \; \bigwedge \mapsto (\bullet,(\mathbb{1},\{\bullet,\bullet\})); \; \left|\cdot\right|^{2} \mapsto (\bullet,(\bullet,\{\bullet\})); \; \left|\cdot\right|_{2} \mapsto (\bullet,(\mathbb{1},\{\bullet_{2}\}));$$

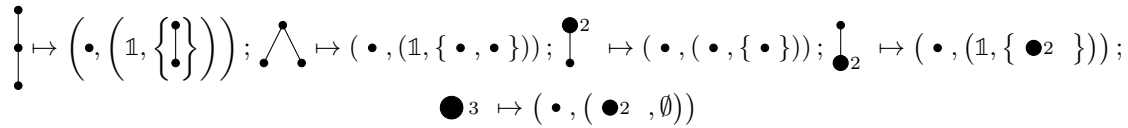$$\bullet_{3} \mapsto (\bullet,(\bullet_{2},\emptyset))$$

Figure 2.5.1: The decorated plane trees of size 3 and their decompositions. The labels to the right of each node indicates the size of the node. Nodes with no label have size 1.

trees to the Kuba-Panholzer recurrence for simple tree classes and then to a recurrence for decorated tree classes. Using decorated tree classes will also allow us to easily extend hook length to even more general tree classes and to expand the types of differential equations we can study in Chapter IV.

A *decorated tree* is a rooted tree whose nodes have any positive integer size. The size of a decorated tree is then the sum of the sizes of its nodes. A *decorated forest* is a multiset of decorated trees. The class of all decorated forests is denoted $\mathcal{F}'$.

The name decorated tree comes from quantum field theory where they are defined in more generality (see [44, §2.5].) We will discuss more about the use of decorated trees in quantum field theory in Section 3.1.

We say a class of decorated trees, $\mathcal{T}'$, is *simple* if there exists a bivariate combinatorial operator, Φ, and an isomorphism $\gamma : \mathcal{T}' \to \mathcal{Z} \times \Phi(\mathcal{Z},\mathcal{T}')$ such that if $x \in \Phi(\mathcal{Z},\mathcal{T}')$ then $x = (\bullet^{i},\{t_{1},\ldots,t_{j}\})$ for some $i \in \mathbb{N}$ and $t_{1},\ldots,t_{j} \in \mathcal{T}'$, and $\gamma(t) = (\bullet,(\bullet^{i-1},\{t_{1},\ldots,t_{j}\}))$ if and only if the root of t has size i and $t_{1},\ldots,t_{j}$ are the subtrees of t whose roots are the children of the root of t. Here $\bullet^{i}$ refers to the sequence of i singletons and has size i.

Let $\mathcal{O}'$ be the *class of decorated plane trees*. A decorated plane tree is a plane tree where each node has any positive integer size. $\mathcal{O}'$ is a simple decorated tree class with operator $\Phi(\mathcal{Z},\mathcal{T}) \cong \mathrm{SEQ}(\mathcal{Z}) \times \mathrm{SEQ}(\mathcal{T})$. See Figure 2.5.1 for the decorated plane trees of size 3 and their decompositions.

We define f_{v} for $v \in V(f)$ of a decorated forest, f, the same as for ordinary forests. This gives us the definition of a hook length operator, $B : \mathcal{F}' \to K$ such that $B(f) = \prod_{v \in V(f)} B_{|f_{v}|}$. We also get hook length series.

Definition 2.5.1. Let $\phi(z,x) = \sum_{i,j \geq 0} \phi_{i,j} z^{i} x^{j}$ be a bivariate formal power series and B be a hook length operator then define

$$F_{\phi,B}(z) = \sum_{t \in \mathcal{O}'} w_{\phi}(t)B(t)z^{|t|},$$

where $w_{\phi}(f) = \prod_{v \in V(f)} \phi_{|v|-1,\deg(v)}$. We call $F_{\phi,B}$ the *decorated hook length series* of ϕ with respect to B.

Like for regular trees we get a recurrence of decorated hook length series. This recurrence is more general than the KP recurrence (Theorem 2.2.4) since the nodes of the trees can have any size and ϕ is bivariate. It is the most general form of the recurrence in this thesis and can be thought of as the main theorem of the thesis.

Theorem 2.5.2. *Let $\phi(z,x)$ be a bivariate formal power series and B be a hook length operator then*

$F_{\phi,B}$ satisfies the recurrence:

$$B_k = \frac{[z^k]F_{\phi,B}(z)}{[z^{k-1}]\phi(z, F_{\phi,B}(z))}, \forall k \geq 1.$$

Proof. This proof is similar to Kuba and Panholzer's proof of Theorem 2.2.4 (Theorem 1 of [35]).

We proceed by induction on k.

For $k = 1$, $[z^1]F_{\phi,B}(z) = w_\phi(\bullet)B(\bullet) = \phi_{0,0}B_1 = B_1[z^0]\phi(z, F_{\phi,B}(z))$.

For $k > 1$,

$$[z^k]F_{\phi,B}(z) = \sum_{t\in\mathcal{O}'_k} w_\phi(t)B(t)$$

$$= \sum_{i=1}^{k}\sum_{j\geq 1}\phi_{i,j}B_k \sum_{\substack{n_1+\cdots+n_j=k-i-1 \\ n_1,\ldots,n_j\geq 1}} \sum_{t_1\in\mathcal{O}'_{n_1},\ldots,t_j\in\mathcal{O}'_{n_j}} \prod_{l=1}^{j}(w_\phi(t_l)B(t_l))$$

$$= B_k\sum_{i=1}^{k}\sum_{j\geq 1}\phi_{i,j} \sum_{\substack{n_1+\cdots+n_j=k-i-1 \\ n_1,\ldots,n_j\geq 1}} \prod_{l=1}^{j}\left(\sum_{t_l\in\mathcal{O}'_{n_l}} w_\phi(t_l)B(t_l)\right)$$

$$= B_k\sum_{i=1}^{k}\sum_{j\geq 1}\phi_{i,j} \sum_{\substack{n_1+\cdots+n_j=k-i-1 \\ n_1,\ldots,n_j\geq 1}} \prod_{l=1}^{j}[z^{n_l}]F_{\phi,B}(z)$$

$$= B_k[z^{k-1}]\phi(z, F_{\phi,B}(z)).$$

$\square$

It is worth mentioning that as with regular trees, the above theorem can be applied to simple decorated tree classes.

Definition 2.5.3. Let $\mathcal{T}' = \mathcal{Z} \times \Phi(\mathcal{Z}, \mathcal{T}')$ and define

$$F_{\mathcal{T}',B}(z) = \sum_{t\in\mathcal{T}'} B(t)z^{|t|},$$

to be the *hook length series* of $\mathcal{T}'$ with respect to B. Note that $F_{\mathcal{T}',B} = F_{\phi,B}$. As a result we get the following recurrence as a corollary to Theorem 2.5.2.

Corollary 2.5.4. *Let $\mathcal{T}' = \mathcal{Z} \times \Phi(\mathcal{Z}, \mathcal{T}')$ be a simple decorated tree class and B be a hook length then $F_{\mathcal{T}',B}$ satisfies the recurrence:*

$$B_k = \frac{[z^k]F_{\mathcal{T}',B}(z)}{[z^{k-1}]\phi(z, F_{\mathcal{T};,B}(z))}, \forall k \geq 1.$$

Now we present an example of a decorated tree formula.

Example 2.5.5. This is a new hook length formula.

Consider the set of decorated plane trees, $\mathcal{O}'$. Suppose we want to want to find which hook length operator gives the hook length series $\frac{z}{1-z}$. Since $\phi(z, x) = \frac{1}{1-z} \frac{1}{1-x}$ for $\mathcal{O}'$ by Theorem 2.5.2 we know that:

$$
\begin{aligned}
B_k &= \frac{[z^k]\frac{z}{1-z}}{[z^{k-1}]\left(\frac{1}{1-z}\frac{1}{1-\frac{z}{1-z}}\right)} \\
&= \frac{1}{[z^{k-1}]\frac{1}{1-2z}} \\
&= 2^{1-k}.
\end{aligned}
$$

Therefore we have that

$$
\sum_{t \in \mathcal{O}'_n} 2^n \prod_{v \in V(t)} \frac{1}{2^{|t_v|}} = 1. \tag{2.5.1}
$$

$\blacksquare$

2.6 General hook length operators

We do not have to restrict our definition of hook length operator to depend on only the size of the subtrees. Here we will generalize the hook weight part, B_n, of the hook length operator definition to take subtrees (instead of sizes of subtrees) to the ring, $\mathbb{K}$.

We call an operator, $B : \mathcal{F} \to \mathbb{K}$, a *general hook length operator* if there exists $B_t \in \mathbb{K}$ for each tree, $t \in \mathcal{F}$, such that for all forests, f, $B(f) = \prod_{v \in V(f)} B_{f_v}$.

However, in order to use a recurrence such as in Theorem 2.2.4, we need B_t to depend on the size of the tree. To alleviate this, we will consider classes of the form: $\mathcal{T} \cong \mathcal{Z} \times \Phi(\mathcal{Z}, \mathcal{T}, \mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(l)})$ and general hook length operators that behave as an ordinary hook length operator for trees in $\mathcal{T}$. In other words for all $s, t \in \mathcal{T}_n$, $B_s = B_t$. However B_t may be a value different from $g(|t|)$ for $t \in \mathcal{T}^{(i)}$. This leads us to the recurrence in the following corollary.

Corollary 2.6.1. *Let $\mathcal{T}$ be a class of decorated trees satisfying: $\mathcal{T} \cong \mathcal{Z} \times \Phi(\mathcal{Z}, \mathcal{T}, \mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(l)})$, where each $\mathcal{T}^{(i)}$ is a class of decorated trees and let B be a general hook length operator where $s, t \in \mathcal{T}_n$ implies $B_s = B_t$. Define*

$$
F(z) = \sum_{t \in \mathcal{T}} B(t) \zeta^{(|t|)},
$$

$$
F^{(i)}(z) = \sum_{t \in \mathcal{T}^{(i)}} B(t) \zeta^{(|t|)}
$$

and ϕ to be the formal power series of Φ. Then F satisfies

$$
B_k = \frac{[z^k]F(z)}{[z^{k-1}]\phi(z, F(z), F^{(1)}(z), \ldots, F^{(l)}(z))} \quad \forall k \geq 1,
$$

where ϕ is the formal power series of Φ.

To see that this follows from Theorem 2.5.2, consider $\tilde{\varphi}(z,x) = \phi(z, x, F^{(1)}(z), \ldots, F^{(l)}(z))$. Then we merely need to show that $F(z) = F_{B,\tilde{\varphi}}(z)$. Since B is invariant over size of t, define $B_n = B_t$ for any $t \in \mathcal{T}_n$. By computing the power series expansion:

$$F(z) = \sum_{t \in \mathcal{T}} B(t)\zeta^{(|t|)}$$

$$= \sum_{i,j,j_1,\ldots,j_n} \phi_{i,j,j_1,\ldots,j_l} \sum_{t_1,\ldots,t_j \in \mathcal{T}} \sum_{r=1}^{l} \sum_{t_1^{(r)},\ldots,t_{j_r}^{(r)} \in \mathcal{T}^{(r)}}$$

$$B_{1+i+\sum_{s=1}^{j}|t_s|+\sum_{r=1}^{l}\sum_{s=1}^{j_r}|t_s^{(r)}|} \prod_{s=1}^{j} B(t_s) \prod_{r=1}^{l}\prod_{s=1}^{j_r} B(t_s^{(r)}) z^{1+i+\sum_{s=1}^{j}|t_s|+\sum_{r=1}^{l}\sum_{s=1}^{j_r}|t_s^{(r)}|}$$

$$= \sum_{i,j,j_1,\ldots,j_n} \phi_{i,j,j_1,\ldots,j_l} \sum_{t_1,\ldots,t_j \in \mathcal{T}} \sum_{k \geq 0}$$

$$B_{1+i+\sum_{s=1}^{j}|t_s|+k} \prod_{s=1}^{j} B(t_s) \left([z^k] \prod_{r=1}^{l} (F^{(r)}(z))^{j_r} \right) z^{1+i+\sum_{s=1}^{j}|t_s|+k}$$

$$= \sum_{i,j,j_1,\ldots,j_n} \phi_{i,j,j_1,\ldots,j_l} \sum_{k \geq 0} [z^k] \left(\prod_{r=1}^{l} (F^{(r)}(z))^{j_r} \right) \sum_{t_1',\ldots,t_j' \in \mathcal{O}} B_{1+i+\sum_{s=1}^{j}|t_s'|+k}$$

$$\prod_{s=1}^{j} (w_{\tilde{\varphi}}(t')B(t_s')) z^{1+i+k+\sum_{s=1}^{j}|t_s'|}$$

$$= \sum_{t' \in \mathcal{O}'} w_{\tilde{\varphi}}(t')B(t')z^{|t'|}$$

$$= F_{B,\tilde{\varphi}}(z).$$

In their paper [35] with the recurrence from Theorem 2.2.4, Kuba and Panholzer generalized their recurrence to more general hook length operators on trees. These included: hook length operators which depends of the height of the tree and hook length operators that treat the root of a tree differently. All of these hook length operators are covered by our definition of a general hook length operator and each of their recurrences follows from Corollary 2.6.1.

For example let $B : \mathcal{F} \to \mathbb{K}$ be a general hook length operator such that $B_t = B_{|t|}^{h(t)}$ where $h(t)$ is the height of t and let $\mathcal{T} \cong \mathcal{Z} \times \Phi(\mathcal{T})$ be a simple tree class. Let $\mathcal{T}^{(h)}$ be the class of trees in $\mathcal{T}$ with height h. Then $\mathcal{T}^{(1)} \cong \phi_0 \mathcal{Z}$ and $\mathcal{T}^{(h)} \cong \mathcal{Z} \times \Phi(\mathcal{T}^{(h-1)})$. By Corollary 2.6.1

$$F_{\mathcal{T}^{(1)},B}(z) = \phi_0 B_1^{(1)} z$$

and

$$B_k^{(h)} = \frac{[z^k]F_{\mathcal{T}^{(h)},B}(z)}{[z^{k-1}]\phi(F_{\mathcal{T}^{(h-1)},B}(z))}.$$

One could also specify a different Φ in the specification of each $\mathcal{T}^{(h)}$.

We shall investigate general hook length operators further in Section 4.5.

Chapter III

The Connes-Kreimer Hopf algebra and hook length

In this chapter we introduce the Connes-Kreimer Hopf algebra of rooted trees. We will also investigate a universal property of the Hopf algebra and its effect on hook length operators.

3.1 Hopf algebra of rooted trees

This section introduces a Hopf algebra of rooted trees developed by Connes and Kreimer in 1998 [13]. A Hopf algebra in short is a structure that is an algebra and coalgebra and has a special morphism called the antipode. The algebra part of the Connes-Kreimer Hopf algebra was first developed by Butcher in 1972 [7] to analyze the Runge-Kutta method. Connes and Kreimer used their Hopf algebra to analyze renormalization in quantum physics. Connes and Kreimer were unaware that Butcher had developed their algebra previously; however Brouder showed that the two algebras were the same in 2000 [5].

Here we will define a Hopf algebra and also the Connes-Kreimer Hopf algebra in parallel. We will do so by following Panzer's Masters thesis [44, Chapter 2]. We will also present a universal property of the Connes-Kreimer Hopf algebra that has an interesting relationship with hook length operators. This relationship will be investigated in the next section.

We can identify any vector space, A, with $\mathbb{K} \otimes A$ or $A \otimes \mathbb{K}$ via scalar multiplication. This is assumed for the definitions of unit and counit. $\mathrm{Hom}(A, B)$ will always be used as the set of vector space morphisms between the vector spaces, A and B. We will frequently use map to mean vector space morphism. We define $\mathrm{End}(A) = \mathrm{Hom}(A, A)$ and $\mathrm{id} \in \mathrm{End}(A)$ as the identity map.

Definition 3.1.1. An *algebra* is a vector space, A, with a map, $m \in \mathrm{Hom}(A \otimes A, A)$, called the *product* such that

$$m \circ (\mathrm{id} \otimes m) = m \circ (m \otimes \mathrm{id}). \ (m \text{ is a associative})$$

An algebra is *unital* if there exists a map, $u \in \mathrm{Hom}(\mathbb{K}, A)$, called the *unit* such that

$$m \circ (u \otimes \mathrm{id}) = \mathrm{id} = m \circ (\mathrm{id} \otimes u).$$

A morphism of algebras, A and B, is a map, $g \in \mathrm{Hom}(A, B)$, such that $g \circ m_A = m_B \circ (g \otimes g)$. g is a morphism of unital algebras as well if A and B are unital and $g \circ u_A = u_B$.

We want to put an algebra structure on the class of rooted trees. Let $H_\mathcal{R}$ be the set of linear combinations of unordered rooted forests with coefficients in $\mathbb{K}$. Define $m \in \mathrm{Hom}(H_\mathcal{R} \otimes H_\mathcal{R}, H_\mathcal{R})$ such that for all $f_1, f_2 \in \mathcal{F}$,

$$m(f_1, f_2) = f_1 \sqcup f_2$$

and extend linearly. Here $\sqcup$ denotes disjoint union. Define $u \in \mathrm{Hom}(\mathbb{K}, H_\mathcal{R})$ to be the map that takes $1 \in \mathbb{K}$ to $\mathbb{1} \in H_\mathcal{R}$. This makes $H_\mathcal{R}$ a unital algebra.

Definition 3.1.2. A *coalgebra* is a vector space, A, with a map, $\Delta \in \mathrm{Hom}(A, A \otimes A)$, called the *coproduct* such that

$$(\mathrm{id} \otimes \Delta) \circ \Delta = (\Delta \otimes \mathrm{id}) \circ \Delta. \ (\Delta \text{ is coassociative})$$

An algebra is *counital* if there exists a map $\epsilon \in \mathrm{Hom}(A, \mathbb{K})$ called the *counit* such that

$$(\epsilon \otimes \mathrm{id}) \circ \Delta = \mathrm{id} = (\mathrm{id} \otimes \epsilon) \circ \Delta.$$

A morphism of coalgebras, A and B, is a map $g \in \mathrm{Hom}(A, B)$ such that $\Delta_B \circ g = (g \otimes g) \circ \Delta_A$. g is a morphism of counital coalgebras as well if A and B are counital and $\epsilon_B \circ g = u_A$.

We shall now define a coalgebra structure on $H_\mathcal{R}$.

Given a forest, f, and a subset of its nodes, $W \subset V(f)$, we say W is *independent* if no two nodes of W are descendants of each other. We define $I(f)$ as the set of all independent subsets of f. For $W \in I(f)$ we shall define $f_W = \{f_v : v \in W\}$.

Define $\Delta \in \mathrm{Hom}(H_\mathcal{R}, H_\mathcal{R} \otimes H_\mathcal{R})$ such that for $f \in \mathcal{F}$ we have

$$\Delta(f) = \sum_{W \in I(f)} f_W \otimes (f \setminus f_W)$$

and extend linearly. Define $\epsilon \in \mathrm{Hom}(H_\mathcal{R}, \mathbb{K})$ to be the map that takes $y \in H_\mathcal{R}$ to be the coefficient of $\mathbb{1}$ in y. This makes $H_\mathcal{R}$ a counital coalgebra.

Definition 3.1.3. A vector space, A, that is a algebra under m and a coalgebra under Δ is called a *bialgebra* if

$$\Delta \circ m = (m \otimes m) \circ \tau \circ (\Delta \otimes \Delta) \ (m \text{ is a morphism of coalgebras}) \qquad (3.1.1)$$

where $\tau(a \otimes b \otimes c \otimes d) = a \otimes c \otimes b \otimes d$. If H is unital and counital then we must also have that

$$\Delta \circ u = u \otimes u \ (u \text{ is a morphism of coalgebras}) \qquad (3.1.2)$$

To show that m is a morphism of coalgebras it suffices to show that Equation 3.1.1 holds for every forest. Let $f_1, f_2 \in \mathcal{F}$ then:

$$
\begin{aligned}
(\Delta \circ m)(f_1 \otimes f_2) &= \Delta(f_1 \sqcup f_2) \\
&= \sum_{W \in I(f_1 \cup f_2)} (f_1 \sqcup f_2)_W \otimes ((f_1 \sqcup f_2) \setminus (f_1 \sqcup f_2)_W) \\
&= \sum_{W_1 \in I(f_1)} \sum_{W_2 \in I(f_2)} ((f_1)_{W_1} \sqcup (f_2)_{W_2}) \otimes ((f_1 \setminus (f_1)_{W_1}) \sqcup (f_2 \setminus (f_2)_{W_2})) \\
&= m \left(\sum_{W_1 \in I(f_1)} (f_1)_{W_1} \otimes \sum_{W_2 \in I(f_2)} (f_2)_{W_2} \right) \otimes \\
&\quad\; m \left(\sum_{W_1 \in I(f_1)} (f_1 \setminus (f_1)_{W_1}) \otimes \sum_{W_2 \in I(f_2)} (f_2 \setminus (f_2)_{W_2}) \right) \\
&= ((m \otimes m) \circ \tau) \\
&\quad\; \left(\sum_{W_1 \in I(f_1)} (f_1)_{W_1} \otimes \sum_{W_1 \in I(f_1)} (f_1 \setminus (f_1)_{W_1}) \otimes \right. \\
&\qquad\qquad \left. \sum_{W_2 \in I(f_2)} (f_2)_{W_2} \otimes \sum_{W_2 \in I(f_2)} (f_2 \setminus (f_2)_{W_2}) \right) \\
&= ((m \otimes m) \circ \tau \circ (\Delta \otimes \Delta))(f_1 \otimes f_2).
\end{aligned}
$$

To show that u is a morphism of coalgebras, we only need to check that Equation 3.1.2 holds for $1 \in \mathbb{K}$:

$$
(\Delta \circ u)(1) = \Delta(\mathbb{1}) = \mathbb{1} \otimes \mathbb{1} = u(1) \otimes u(1).
$$

Therefore A is a bialgebra.

Definition 3.1.4. A bialgebra, A, is called a *Hopf algebra* if there exists $S \in \operatorname{End}(A)$ such that:

$$
m \circ (\operatorname{id} \otimes S) \circ \Delta = u \circ \epsilon = m \circ (S \otimes \operatorname{id}) \circ \Delta.
$$

We call S the *antipode* of A.

To easily show that $H_{\mathcal{R}}$ is a Hopf algebra we need a few more definitions.

Definition 3.1.5. A family of subspaces $(A^n)_{n \in \mathbb{N}}$ of a bialgebra, A, with product, m, and coproduct, Δ is a *filtration* if:

1. $A^n \subseteq A^{n+1} \quad \forall n \in \mathbb{N}$,

2. $A = \sum_{n \in \mathbb{N}} A^n$,

3. $\Delta(A^n) \subseteq \sum_{i=0}^{n} A^i \otimes A^{n-i} \quad \forall n \in \mathbb{N}$ and

4. $m(A^n \otimes A^m) \subseteq A^{n+m} \quad \forall n \in \mathbb{N}$.

A bialgebra is said to be *connected* if there exists a filtration $(A^n)_{n \in \mathbb{N}}$ with $A^0 = \mathbb{K} \otimes \mathbb{1}$.

Theorem 3.1.6. *Every connected bialgebra is a Hopf algebra.*

To prove this theorem we merely need to find S. For this classic result we shall follow Panzer [44, §2.1.2]. He computed $S = \sum_{n \in \mathbb{N}} (u \circ \epsilon - \mathrm{id})^{\star n}$ where $u^{\star n+1} = m \circ (u^{\star n} \otimes u) \circ \Delta$ for $n \geq 1$, $u^{\star 1} = u$ and $u^{\star 0} = u \circ \epsilon$. The filtration of the connected bialgebra turns the infinite sum in the definition of S into a finite sum, ensuring the existence of S.

We show that $H_{\mathcal{R}}$ is a Hopf algebra by showing that it is connected. For $n \in \mathbb{N}$ let $\mathcal{F}_n$ be the set of forests with n nodes and $H_{\mathcal{R}}^n = \sum_{i=0}^{n} \mathbb{K} \otimes \mathcal{F}_i$. Clearly $H_{\mathcal{R}}^n \subseteq H_{\mathcal{R}}^{n+1}$ and $H_{\mathcal{R}} = \sum_{n \in \mathbb{N}} A^n$. Let f be a forest with n nodes then

$$\Delta(f) = \sum_{W \in I(f)} f_W \otimes (f \setminus f_W) \in \sum_{i=0}^{n} H_{\mathcal{R}}^i \otimes H_{\mathcal{R}}^{n-i}.$$

If $f_1 \in \mathcal{F}^n$ and $f_2 \in \mathcal{F}^m$ then

$$f_1 \otimes f_2 = f_1 \sqcup f_2 \in \mathcal{F}^{m+n} \subset H_{\mathcal{R}}^{n+m}.$$

Therefore $(H_{\mathcal{R}}^n)_{n \in \mathbb{N}}$ is a filtration. Since $H_{\mathcal{R}}^0 = \mathbb{K} \otimes \mathbb{1}$, $H_{\mathcal{R}}$ is connected and thus a Hopf algebra. This Hopf algebra is called the Connes-Kreimer Hopf algebra [13, 44].

An important operator on the Connes-Kreimer Hopf algebra is the grafting operator. We define the *grafting operator*, $B_+ : H_{\mathcal{R}} \to H_{\mathcal{R}}$, as the algebra morphism such that for a forest $t_1 \cdots t_n$, $B_+(t_1 \cdots t_n)$ is the tree whose root is attached to $t_1, \ldots, t_n$.

Any forest, f, can be written as compositions and products of B_+ applied to $\mathbb{1}$ (uniquely up to permutation). For some examples, see Figure 3.1.1. To prove this simply use induction on the size of the forest. This decomposition is similar to the decomposition for simple trees discussed in Section 2.1.1. In fact, B_+ equations can be interpreted as simple tree specifications by replacing $B_+(X)$ with $\mathcal{Z} \times X$. This provides a link between simple tree classes and combinatorial Dyson-Schwinger equations. For more on Dyson-Schwinger equations see [57, 44].

Theorem 3.1.7 (Panzer 2011 [44, Theorem 2.4.6]). *Let A be a commutative unital algebra and $L \in \mathrm{End}(A)$. Then there exists a unique morphism of unital algebras, $L_\rho : H_{\mathcal{R}} \to A$, such that:*

$$L_\rho \circ B_+ = L \circ L_\rho. \tag{3.1.3}$$

If A is a bialgebra and $\Delta_A \circ L = (\mathrm{id}_A \otimes L) \circ \Delta_A + L \otimes 1_A$ then L_ρ is a morphism of coalgebras and bialgebras. If A is also a Hopf algebra then L_ρ is a morphism of Hopf algebras.

The L_ρ basically replaces B_+ with L and $\mathbb{1}$ with 1_A, the multiplicative identity in A, in the decomposition

$$B_+(B_+(\mathbb{1})B_+(B_+(\mathbb{1})B_+(B_+(\mathbb{1})B_+(\mathbb{1}))B_+(\mathbb{1})))$$

(a) Decomposition of a tree

$$B_+(B_+(\mathbb{1})B_+(\mathbb{1}))B_+(\mathbb{1})B_+(B_+(\mathbb{1})B_+(\mathbb{1})B_+(B_+(\mathbb{1})))$$

(b) Decomposition of a forest

$$a_1B_+(B_+(\mathbb{1})B_+(B_+(\mathbb{1})B_+(B_+(\mathbb{1})B_+(\mathbb{1}))B_+(\mathbb{1}))) +$$
$$a_2B_+(B_+(\mathbb{1})B_+(\mathbb{1}))B_+(\mathbb{1})B_+(B_+(\mathbb{1})B_+(\mathbb{1})B_+(B_+(\mathbb{1}))) + a_3\mathbb{1}$$
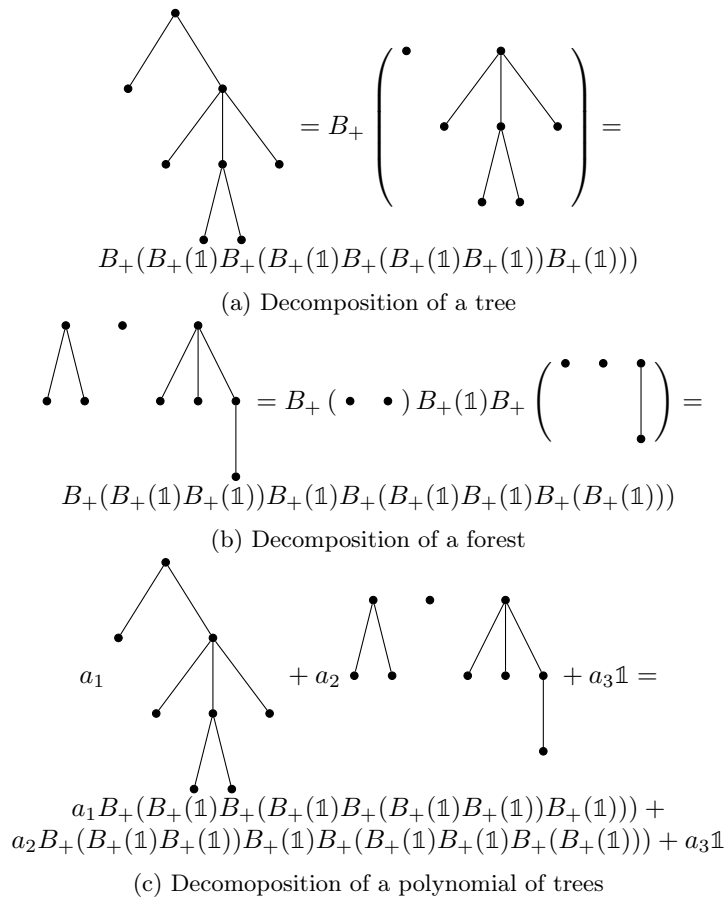
(c) Decomoposition of a polynomial of trees

Figure 3.1.1: Examples of how to decompose elements of $H_{\mathcal{R}}$ into compositions and products of B_+ applied to $\mathbb{1}$.

of a forest. For example

$$L_\rho \left(\begin{array}{c} \end{array} \right) = L \circ L_\rho \left(\begin{array}{c} \end{array} \right) = L(L(1_A)L(L(1_A)L(L(1_A)L(1_A))L(1_A))).$$

In the following section we will investigate a map, $L_B^* \in \mathrm{End}(\mathbb{K}[z])$, which is defined using a hook length operator, B and the morphism, $(L_B^*)_\rho$, resulting from Theorem 3.1.7.

We can also build a Hopf algebra for decorated trees. Panzer mentions this briefly [44, §2.5], though he considers a more general type of tree. By extending the definitions of product, unit, coproduct and counit from $H_\mathcal{R}$ we can derive the Hopf algebra, $H_{\mathcal{R}'}$, of decorated trees.

Instead of having one grafting function, B_+, $H_{\mathcal{R}'}$ has a grafting function for each possible size of the new root. For $a \in \mathbb{N}^+$ we denote the grafting function, $B_+^{(a)} : H_{\mathcal{R}'} \to H_{\mathcal{R}'}$, to be such that $B_+^{(a)}(t_1 \cdots t_n)$ is the decorated tree with root of size a attached to the trees $t_1, \ldots, t_n$.

These grafting functions are used in quantum field theory to express the loop number of the inserted graph in the associated Feynman diagram [57].

An important property of $H_{\mathcal{R}'}$ is that it also satisfies a universal property similar to the universal property of $H_\mathcal{R}$ (Theorem 3.1.7).

Theorem 3.1.8 (Panzer 2011 [44, §2.5]). *Let A be a commutative unital algebra and $L_a \in \mathrm{End}(A)$ for $a \in \mathbb{N}$. Then there exists a unique morphism of unital algebras, $L_{\rho'} : H_{\mathcal{R}'} \to A$, such that for all $a \in \mathbb{N}$:*

$$L_{\rho'} \circ B_+^{(a)} = L_a \circ L_{\rho'}. \tag{3.1.4}$$

If A is a bialgebra and for $a \in \mathbb{N}$, $\Delta_A \circ L_a = (\mathrm{id}_A \otimes L_a) \circ \Delta_A + L_a \otimes 1_A$ then $L_{\rho'}$ is a morphism of coalgebras and bialgebras. If A is also a Hopf algebra then $L_{\rho'}$ is a morphism of Hopf algebras.

The L_ρ and L'_ρ maps of this section play the role of Feynman rules. See [44] for more.

3.2 The L_B^* operator

In this section we study a family of maps we call L_B^* that has not been studied previously though Panzer studies some particular examples. These maps are useful because each map, L_B^*, leads to a specification of a weighted combinatorial class $(\mathcal{T}, B)$, whose generating function is $F_{\mathcal{T},B}$. We will give some examples of L_B^* maps and explain how they are related to Feynman rules. We will also explore how the maps behave under the universal property. This generalizes a result by Panzer [44].

For a hook length operator, B, we define $L_B^* \in \mathrm{End}(\mathbb{K}[z])$ such that $L_B^*(z \mapsto z^n) = z \mapsto B_{n+1}z^{n+1}$ for all $n \in \mathbb{N}$. With a general hook length operator we shall write L^*.

For the hook length, σ, with $\sigma_k = \frac{1}{k}$, $L_\sigma^* = \int_0$ where $\int_0 (z \mapsto x(z)) = z \mapsto \int_0^z x(\varepsilon) d\varepsilon$. It turns out that $\int_0$ is a cocycle of the Hopf algebra of polynomials, see [44, Lemma 2.6.2], and $(L_\sigma^*)_\rho$ is a renormalized Feynman rule [44, §3.5].

Because of the nice property that defines L_B^*, we can use the map in a specification for hook length series. We show this in the following theorem. This theorem is a rephrasing of Kuba and Panholzer's recurrence (Theorem 2.2.4) in the language of renormalization.

Theorem 3.2.1. *Let $\mathcal{T} = \mathcal{Z} \times \Phi(\mathcal{T})$ be a simple tree class and $\phi(x)$ be the formal power series of Φ then*

$$F_{\mathcal{T}} = L^* \phi(F_{\mathcal{T}}).$$

Proof. Since for any formal power series X, $[z^0] L^*(X(z)) = 0$ and because $[z^0] F_{\mathcal{T}}(z) = 0$, it follows that $[z^0] F_{\mathcal{T}} = [z^0] L^* \phi(F_{\mathcal{T}})$.

By Corollary 2.2.5, for $n \geq 1$ we have:

$$[z^n] F_{\mathcal{T},B}(z) = B_n [z^{n-1}] \phi(F_{\mathcal{T},B}(z)) = [z^n] B_n z \phi(F_{\mathcal{T},B}(z)) = [z^n] L_B^* \phi(F_{\mathcal{T},B}(z)).$$

Therefore $F_{\mathcal{T}} = L^* \phi(F_{\mathcal{T}})$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

Despite giving a specification of $(\mathcal{T}, B)$ this theorem does not result in any new applications.

Using Corollary 2.6.1 we can extend the above theorem to our more general setting of Section 2.6.

Corollary 3.2.2. *Let $\mathcal{T}$ be a class of decorated trees satisfying: $\mathcal{T} \cong \mathcal{Z} \times \Phi(\mathcal{Z}, \mathcal{T}, \mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(l)})$, where each $\mathcal{T}^{(i)}$ is a class of decorated trees and let B be a general hook length operator where $s, t \in \mathcal{T}_n$ implies $B_s = B_t$. Define*

$$F(z) = \sum_{t \in \mathcal{T}} B(t) \zeta^{(|t|)},$$

$$F^{(i)}(z) = \sum_{t \in \mathcal{T}^{(i)}} B(t) \zeta^{(|t|)}$$

and ϕ to be the formal power series of Φ. Then F satisfies

$$F(z) = L_B^* \phi(z, F(z), F^{(1)}(z), \ldots, F^{(l)}(z)),$$

where ϕ is the formal power series of Φ.

3.2.1 Examples of L_B^*

Here we give some examples of L_B^* for different hook lengths.

Let θ be the operator $z \frac{d}{dz}$. It turns out that $p(\theta)(z^n) = p(n) z^n$ for any polynomial $p(\theta) \in \mathbb{K}[\theta]$. To show this we just need to compute $\theta^k(z_n)$ for $k \in \mathbb{N}^+$. We have that $\theta^0(z_n) = 1$ and that $\theta^k(z_n) = n^k z^n$ implies

$$\theta^{k+1}(z_n) = \theta(\theta^k(z)) \theta(n^k z^n) = z n^k n z^{n-1} = n^{k+1} z^n.$$

Hence the statement is true by induction.

Theorem 3.2.3. *Let B be a hook length operator with $B_k = k^q$ with $q \in \mathbb{N}$ then*

$$L_B^* : [z \mapsto x(z)] \mapsto [z \mapsto z(1+\theta)^q x(z)].$$

Proof. L_B^* given above is linear because it is the composition of linear equations

Let $n \in \mathbb{N}$ then

$$
\begin{aligned}
L_B^*(z \mapsto z^n) &= z \mapsto z(1+\theta)^q z^n \\
&= z \mapsto z(1+n)^q z^n &&\text{since } (1+\theta)(z^n) = (1+n)z^n \\
&= z \mapsto B_{n+1} z^{n+1}
\end{aligned}
$$

as desired. □

Theorem 3.2.4. *Let B be a hook length operator with $B_k = k^{\text{-}q}$ with $q \in \mathbb{N}^+$ then*

$$L_B^* : [z \mapsto x(z)] \mapsto \left[z \mapsto z \left(z^{\text{-}1} \int_0^{\cdot} \right)^q x(z) \right].$$

Proof. L_B^* given above is linear because it is the composition of linear equations

Let $n \in \mathbb{N}$ then

$$
\begin{aligned}
L_B^*(z \mapsto z^n) &= z \mapsto z \mapsto z \left(z^{\text{-}1} \int_0^{\cdot} \right)^q z^n \\
&= z \mapsto z(1+n)^{\text{-}q} z^n &&\text{since } z^{\text{-}1} \int_0^{\cdot} z^n = (1+n)^{\text{-}1} z^n \\
&= z \mapsto B_{n+1} z^{n+1}
\end{aligned}
$$

as desired. □

By taking sums of the above results we can derive L_B^* for any rational $B_k = \frac{P(k)}{Q(k)}$.

This last example will be used in Section 4.6 for a new method to solve hook length formulae we call the scaled method.

Theorem 3.2.5. *If B and C are hook lengths with $B_k = r^{k-1}C_k$ for all $k \in \mathbb{N}^+$ then*

$$L_B^*(z \mapsto x(z)) = L_C^*(z \mapsto x(rz)).$$

Proof. We must prove that L_B^* given in the theorem satisfies $L_B^*(z \mapsto z^n) = z \mapsto B_{n+1} z^{n+1}$ and that L_B^* is linear.

Let $x, y : \mathbb{K}[z] \to K[z]$, $a, b \in K$ then

$$
\begin{aligned}
L_B^*(z \mapsto (ax+by)(z)) &= L_C^*(z \mapsto (ax+by)(rz)) \\
&= aL_C^*(z \mapsto x(rz)) + bL_C^*(z \mapsto y(rz)) &&\text{(because } L_C^* \text{ is linear)} \\
&= aL_B^*(z \mapsto x(z)) + bL_B^*(z \mapsto y(z)).
\end{aligned}
$$

So L_B^* is linear.

Let n be an integer, then

$$L_B^*(z \mapsto z^n) = L_C^*(z \mapsto r^n z^n) = z \mapsto r^n C_{n+1} z^{n+1} = z \mapsto B_{n+1} z^{n+1}.$$

$\square$

3.2.2 The universal property and L_B^*

In this subsection we will investigate how the L_B^* acts under the universal property of Theorem 3.1.7. For convenience we will denote $(L_B^*)_\rho$ by $L_{B,\rho}^*$.

The following theorem is a generalization of a result by Panzer [44, Lemma 3.1.1]. Panzer proved it for the case where $B = \sigma$ and $L_B^* = \int_0$.

Theorem 3.2.6. *Let B be a hook length operator. For any forest, f, we have*

$$L_{B,\rho}^*(f) = B(f)z^{|f|}.$$

Proof. Proof by induction on $|f|$.

In the base case $f = \mathbb{1}$ and $L_{B,\rho}^*(\mathbb{1}) = 1 = B(\mathbb{1})z^0$.

Suppose that the statement holds for all forests on fewer vertices.

If f is a forest then the statement holds for all $t \in \pi_0(f)$ so

$$L_{B,\rho}^*(f) = \prod_{t \in \pi_0(f)} L_{B,\rho}^*(t) = \prod_{t \in \pi_0(f)} \left(B(t)z^{|t|}\right) = \left(\prod_{t \in \pi_0(f)} B(t)\right) z^{\sum_{t \in \pi_0(f)} |t|} = B(f)z^{|f|}.$$

If f is a tree then there exists a forest, f', such that $B_+(f') = f$. Since $|f| = |f'| + 1$ we have

$$L_{B,\rho}^*(f) = (L_{B,\rho}^* \circ B_+)(f') = (L_B^* \circ L_{B,\rho}^*)(f') = L_B^*(B(f')z^{|f'|}) = B_{|f'|+1}B(f')z^{|f'|+1}$$
$$= B_{|f|}B(f')z^{|f|} = B(f)z^{|f|}.$$

$\square$

The above theorem implies that for any class of forests, $\mathcal{T}$, $F_{\mathcal{T},B}(z) = \sum_{f \in \mathcal{T}} L_{B,\rho}^*(f)$. $L_{B,\rho}^*$ can also be seen as the hook length operator, C, with $C_k = B_k z$.

Because the universal property also applies to the Hopf algebra of decorated trees, we get a similar result as the previous theorem for $H_{\mathcal{R}'}$. Define $L_{B,\rho'}^* = L_{\rho'}$ where $L_a = L_B^*$ for all $a \in \mathbb{N}$.

Theorem 3.2.7. *Let B be a hook length operator. For any decorated forest, f, we have*

$$L_{B,\rho'}^*(f) = B(f)z^{|f|}.$$

48

Proof. Proof by induction on $|f|$.

In the base case $f = \mathbb{1}$ and $L^*_{B,\rho'}(\mathbb{1}) = 1 = B(\mathbb{1})z^0$.

Suppose that the statement holds for all decorated forests on fewer vertices.

If f is a decorated forest then the statement holds for all $t \in \pi_0(f)$ so

$$L^*_{B,\rho'}(f) = \prod_{t \in \pi_0(f)} L^*_{B,\rho'}(t) = \prod_{t \in \pi_0(f)} \left(B(t)z^{|t|} \right) = \left(\prod_{t \in \pi_0(f)} B(t) \right) z^{\sum_{t \in \pi_0(f)} |t|} = B(f)z^{|f|}.$$

If f is a decorated tree then there exists a decorated forest, f', and positive integer, a, such that $B_+^{(a)}(f') = f$ and $|f| = |f'| + a$. Thus we have

$$L^*_{B,\rho'}(f) = (L^*_{B,\rho'} \circ B_+^{(a)})(f') = (L^*_B \circ L^*_{B,\rho'})(f') = L^*_B(B(f')z^{|f'|}) = B_{|f'|+1}B(f')z^{|f'|+a}$$
$$= B_{|f|}B(f')z^{|f|} = B(f)z^{|f|}.$$

$\square$

In Panzer's thesis [44, Chapter 3], he was interested in Feynman rules. He defined *Feynman rules* as morphisms, $\phi : H_{\mathcal{R}} \to A$, where A is some commutative algebra. In particular he was interested in Feynman rules, ϕ, that satisfied the universal property for some morphism, $L : A \to A$. In other words $\phi \circ B_+ = L \circ \phi$ and $\phi = L_\rho$. In particular Panzer considered $L^*_{\sigma,\rho} = \int_0$.

He also considered the more complicated Feynman rule, $_z\phi_s : H_{\mathcal{R}} \to A$, which satisfies

$$_z\phi_s \circ B_+ = \int_0^\infty g(\epsilon)(s\epsilon)^{-z} {}_z\phi_{s\epsilon} d\epsilon$$

for a given $g : A \to A$. $_z\phi_s$ is also the hook length operator defined by $B_k = s^{-z}G(kz)$ where $G(z) = \int_0^\infty g(\epsilon)d\epsilon$ [44, Proposition 3.2.2].

Because of the universal property (Theorem 3.1.7), Panzer's Feynman rules behave like hook length operators. From Theorems 3.2.6 and 3.2.7, it follows that hook length operators are Feynman rules. Since simple tree specifications can be written in terms of B_+, these specifications can represent Dyson-Schwinger equations. See [44, §3.6]. Thus we can find the associated hook length formula for a given Feynman rule and Dyson-Schwinger equation to give new toy models for quantum physics. Furthermore if our hook length formulae involve a closed form function then these toy models will have exact solutions.

For combinatorialists the Feynman rule language is useful because of the specification-like form of the hook length formulae and the ability to add parameters to hook length operators.

Chapter IV

Differential equations of hook length series

This chapter explores how hook length series can be expressed as solutions to particular differential equations. We will begin with Butcher's work analyzing Runge-Kutta methods, which led him to solutions that were hook length series with the operator σ. Then we will explore Mazza's generalization of Butcher's solution that can solve more differential equations and encapsulates more hook length series. We will use Mazza's differential equation to come up with new hook length formulae. Next we will generalize Mazza's equation for decorated trees and the more general tree classes of Section 2.6. We use our generalization to develop new methods to find hook length formulae, which we use to find some new formulae. Finally we shall show how to use our generalization to produce combinatorial solutions of differential equations.

4.1   B-series and Runge-Kutta methods

Consider the dynamical system:

$$y'(z) = \psi(z, y(z)), \quad y(z_0) = y_0, \tag{4.1.1}$$

where $\psi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is smooth.

To solve this dynamical system we may use B-series. See [25] for a reference. To define B-series first we must define the tree diferential. The *tree differential* of a forest, f, is defined as $\delta_f = \prod_{v \in V(f)} \frac{\partial^{\deg(v)}}{\partial z^{\deg(v)}}$. For some examples see Table 4.1.1.

Definition 4.1.1. Let $\alpha(f)$ be the number of increasing labellings of the forest, f. Given a functional
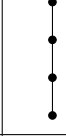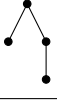
t				
δ_t	$D_0 D_1^3$	$D_0^2 D_1 D_2$	$D_0^2 D_1 D_2$	$D_0^3 D_3$

Table 4.1.1: Some examples of tree differentials. Here $D_n = \frac{d^n}{dz^n}$.

$a : \mathfrak{T} \to \mathbb{R}$, we call the formal power series:

$$Y_{\psi,a}(z) = y_0 + \sum_{t \in \mathfrak{T}} a(t)\alpha(t)\delta_t(\psi(y_0))\frac{(z - z_0)^{|t|}}{|t|!} \tag{4.1.2}$$

a *B-series*.

The term B-series was coined by Hairer and Wanner in 1974 [26]. They are called B-series in honour of Butcher because of his power series approach to Runge-Kutta method analysis [6, 7].

When $\psi(z, y)$ is independent of z ($\psi(z, y) = \psi(y)$), the B-series, $Y(z)$, with $a(t) = 1$ for all t solves the dynamical system in Equation 4.1.1 exactly [25]. In fact this B-series is related to hook length series by $Y(z) = y_0 + F_{\phi,\sigma}(z - z_0)$ where $\phi(x) = \psi(x + y_0)$. See [42] for the details of this result. Mazza proves it for hook length series in general where the hook length operator B is related to the functional given by $a(t) = B(t)t!$.

An interesting property of B-series is how they compose.

Theorem 4.1.2 (Hairer and Wanner [26]). *Suppose $a, b : \mathfrak{T} \to \mathbb{R}$ then*

$$Y_{(Y_{\psi,b}),a}(z) = Y_{\psi,ab}(z),$$

where $(ab)(t) = a(t)b(t)$.

Since hook length series are B-series, this property also holds for hook length series.

Though B-series give an exact solution to the differential equation, it is not always practical as the B-series solution is an infinite series solution. A related way to solve this system numerically is to use a Runge-Kutta method [8, 25].

Choose $a_{i,j}, b_i, c_i \in \mathbb{R}$ then the following is an *s-stage Runge-Kutta method*:

$$k_1 = \psi(z_0, y_0)$$
$$k_2 = \psi(z_0 + c_2 h, y_0 + a_{2,1} k_1 h)$$
$$\cdots$$
$$k_s = \psi(z_0 + c_s h, y_0 + (a_{s,1} k_1 + \cdots + a_{s,s-1} k_{s-1})h)$$
$$y(h) = y_0 + (b_1 k_1 + \cdots + b_s k_s)h.$$

Usually the c_i satisfy $c_i = \sum_{j=1}^{i} a_{i,j}$ [25]. Choosing specific coefficients $(a_{i,j}, b_i, c_i)$ the Runge-Kutta method gives $y(h)$ as a numerical for differential equations of the form in Equation 4.1.1. Some examples of Runge-Kutta methods are given in Figure 4.1.1.

When ψ is independent of z, the numerical solution, $y(h)$, of a Runge-Kutta method is in fact a B-series. Extend $a_{j,k}$ such that $a_{j,k} = 0$ whenever $j > s$ or $k \geq s$. Set the tree function a to be

$$a(t) = \sum_{j=1}^{s} b_j \Phi_j(t)$$

where

$$\Phi_j(B(t_1, \ldots, t_m)) = \sum_{1 \leq k_1, \ldots, k_m < j} \prod_{i=1}^{m} a_{j,k_i} \Phi_{k_i}(t_i)$$

and $\Phi_j(\bullet) = 1$. Then $y(h) = Y_{a,\phi}(h)$ because from [25, §II.2] we have:

$$y(h) = y_0 + \sum_{t \in \mathfrak{T}} \alpha(t) \delta_t(\psi(y_0)) \left(\sum_{j=1}^{s} b_j \Phi_j(t) \right) \frac{h^{|t|}}{|t|!}.$$

4.2 Mazza's differential equation

In this section we will look at a more general family of differential equations than in the previous section. We will show that this family of differential equations is equivalent to the Kuba-Panholzer recurrence.

In order to extend the differential equation, first we need a special operator. For a hook length operator, B, we define the operator, $L_B(x) = \sum_{n \geq 0} L_n x^n$, such that for all $k \in \mathbb{N}^+$, $L_B(k) = kB_k$. Recall that $P(1 + \theta)(z^n) = P(1 + n)z^n$ and so

$$L_B(1 + \theta)(z^n) = L_B(1 + n)z^n = (n+1)B_{n+1}z^n.$$

The L_B operator is closely related to the L_B^* operator of Section 3.2. Applying $\int_0$ to $L_B(1 + \theta)$ we get

$$\int_0 L_B(1 + \theta)(z^n) = \int_0 (n+1)B_{n+1}z^n = B_{n+1}z^{n+1} = L_B^*(z^n).$$

Therefore $\int_0 L_B(1 + \theta) = L_B^*$.

Using the L_B operator we can derive the following differential equation.

Theorem 4.2.1 (Mazza 2004 [42]). *Let B be a hook length operator and $\phi(z) = \sum_{i=1}^{n} \phi_n z^n$ be a formal power series then the B-series,*

$$Y(z) = Y_{B\cdot!, \phi}(z) = \sum_{t \in \mathfrak{T}} B(t) t! \alpha(t) \delta_t(\phi(0)) \frac{z^{|t|}}{|t|!},$$

satisfies the differential equation:

$$Y' = L_B(1 + \theta)\phi(F). \tag{4.2.1}$$

52

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{2,1} & & & & \\
c_3 & a_{3,1} & a_{3,2} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

(a) The standard representation of a Runge-Kutta method.

$$
\begin{array}{c|cc}
0 & & \\
1/2 & 1/2 & \\
\hline
 & 0 & 1
\end{array}
$$

(b) Runge-Kutta method due to Runge [46].

$$
\begin{array}{c|cccc}
0 & & & & \\
1/3 & 1/3 & & & \\
2/3 & -1/3 & 1 & & \\
1 & 1 & -1 & 1 & \\
\hline
 & 1/8 & 3/8 & 3/8 & 1/8
\end{array}
$$

(c) Runge-Kutta method due to Kutta [36].

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
1/5 & 1/5 & & & & & & \\
3/10 & 3/40 & 9/40 & & & & & \\
4/5 & 44/45 & -56/13 & 32/9 & & & & \\
8/9 & 19372/6561 & -25360/2187 & 64448/6561 & -212/729 & & & \\
1 & 9017/3168 & -355/33 & 46732/5247 & 49/176 & -5103/18656 & & \\
1 & 35/384 & 0 & 500/1113 & 125/192 & -2187/6784 & 11/84 & \\
\hline
 & 5179/57600 & 0 & 7571/16695 & 393/640 & -92097/339200 & 187/2100 & 1/40
\end{array}
$$

(d) Runge-Kutta method due to Dormand and Prince [17].

Figure 4.1.1: Matrices of coefficients for Runge-Kutta methods. The coefficients are represented as a matrix in the form of (a). These examples were taken from [25]

. The methods due to Runge and Kutta are among the earliest Runge-Kutta methods and the Runge-Kutta method due to Dormand and Prince is a more modern method that is more accurate.

As we saw in the previous section, the B-series, Y, in the above theorem is equal to the hook length series $F_{\phi,B}$. Thus Equation 4.2.1 is a differential equation of hook length series as exhibited in following corollary.

Corollary 4.2.2. *Let B be a hook length operator and $\mathcal{T} \cong \mathcal{Z} \times \Phi(\mathcal{T})$ be a simple tree class. Then $F = F_{\mathcal{T},B}$ satisfies the differential equation:*

$$F' = L_B(1 + \theta)\phi(F).$$

From the equality of Y and $F_{\phi,B}$, Theorem 4.2.1 follows directly from Theorem 2.2.4 since $[z^n]Y'(z) = (n+1)[z^{n+1}]Y(z)$ and $[z^n]L_B(1+\theta)\phi(Y(z)) = (n+1)B_{n+1}[z^n]\phi(Y(z))$.

When using the diferential equation to derive hook length formulae we use the initial condition $F(0) = 0$ since tree classes, in this thesis, do not contain the empty tree.

Though the two relations are equivalent, one advantage of Mazza's differential equation over the Kuba-Panholzer recurrence (Theorem 2.2.4) is that it gives a criterion for the existence of a hook length formula: solvability of the differential equation.

Also, if we are able to solve the differential equation, we can obtain a hook length formulae given a tree class and a hook length operator. The hook length formula obtained will also generally be in a closed form. We can use and have used the differential equation to find new novel hook length formulae that may be difficult to find using the Kuba-Panholzer recurrence.

Finally the differential equation is a simpler method to verify hook length formulae than the Kuba-Panholzer recurrence, since it requires differentiation instead of coefficient extraction.

Example 4.2.3. In this example we shall verify Postnikov's formula (Equation 1.0.1) using Mazza's differential equation. Let $\mathcal{B} \cong \mathcal{Z} \times (1 + \mathcal{B})^2$ be the class of binary trees. Let $B_k = 1 + \frac{1}{k}$. Postnikov's formula implies that

$$F_{\mathcal{B},B}(z) = \frac{\mathbf{W}(2z)}{2z} - 1.$$

Since $L_B(k) = k + 1$, by Corollary 4.2.2 $F_{\mathcal{B},B}(z)$ satisfies the differential equation:

$$F'_{\mathcal{B},B}(z) = 2(1 + F_{\mathcal{B},B}(z))^2 + 2z(1 + F_{\mathcal{B},B}(z))F'_{\mathcal{B},B}(z).$$

Substituting $F_{\mathcal{B},B}(z) = \frac{\mathbf{W}(2z)}{2z} - 1$ and using $\mathbf{W}'(z) = \frac{\mathbf{W}(z)}{z(1-\mathbf{W}(z))}$ we get:

$$\frac{\mathbf{W}(2z)^2}{2z^2(\mathbf{W}(2z) - 1)} = \frac{\mathbf{W}(2z)^2}{2z^2} + \frac{\mathbf{W}(2z)^3}{2z^2(1 - \mathbf{W}(2z))}.$$

Bringing the right hand side of the equation under the same denominator, we can see that the equation is satisfied and Postnikov's equation holds. ∎

One apparent problem with Equation 4.2.1 is that it only works if kB_k is a polynomial. Otherwise $L(k)$ is not a well-defined operator. However we can fix this if L_B is rational in k.

Let $L_B(k) = \frac{g(k)}{h(k)}$ $\left(B_k = \frac{g(k)}{kh(k)} \right)$ for polynomials g, h. Then

$$[z^n]h(1+\theta)L_B(1+\theta)\phi(F) = [z^n]L_B(1+\theta)\phi(F)$$
$$= h(n+1)\frac{g(n+1)}{h(n+1)}[z^n]\phi(F)$$
$$= g(n+1)[z^n]\phi(F)$$
$$= [z^n]g(1+\theta)\phi(F).$$

Therefore Y satisfies:

$$h(1+\theta)Y' = g(1+\theta)\phi(Y)$$

by Theorem 4.2.1.

This means that for rational hook lengths, L_B behaves as a rational operator. We will see how to handle more complicated hook length operators in Section 4.6 with our new methods.

As with the Kuba-Panholzer recurrence we can derive a differential equation for forest hook length series.

Corollary 4.2.4. *Let B be a hook length operator and $\phi(z) = \sum_{i \geq 0} \phi_i z^i$ be a formal power series with $\phi_0 \neq 0$. Then $G_{\phi,B} = G$ satisfies the differential equation:*

$$G'(z) = \phi'(\phi^{-1}(G(z)) \cdot L_B(1+\theta)G(z). \tag{4.2.2}$$

Proof. Let ϕ and B be as above. Equation 4.2.2 is equivalent to the following differential equation:

$$(\phi^{-1}(G(z)))' = L_B(1+\theta)G(z).$$

Taking the coefficient extraction of both sides we get:

$$[z^{k+1}](\phi^{-1}(G(z)))' = [z^{k+1}]L_B(1+\theta)G(z)$$
$$(k+1)[z^{k+1}]\phi^{-1}(G(z)) = (k+1)B_{k+1}[z^k]G(z),$$

which is the recurrence in Theorem 2.2.7. □

4.3 Hypergeometric differential equations

The *hypergeometric function*, $_2\mathbf{F}_1(a, b; c; z) = \sum_{n \geq 0} \frac{\Gamma(n+a)\Gamma(n+b)\Gamma c}{n!\Gamma(n+c)} z^n$, also called the Gauss function [49, §1.1], is a classical function in analysis. The hypergeometric function, $F(z) = {}_2\mathbf{F}_1(a, b; c; z)$ solves the differential equation [49, §1.2]:

$$z(1-z)\frac{d^2}{dz^2}F(z) + (c - (1+a+b)z)\frac{d}{dz}F(z) - aby = 0. \tag{4.3.1}$$

This differential equation is called the *hypergeometric differential equation*.

One may notice that Equation 4.3.1 looks similar to the types of equations in Theorem 4.2.1. In fact for $B_k = \frac{(k-a)(k-b)}{k(k-c)}$ and $\phi(x) = (1+x)$ the differential equation of Theorem 4.2.1 is equivalent to

$$z(1-z)\frac{d^2}{dz^2}F_{\phi,B}(z) + (c - (1+a+b)z)\frac{d}{dz}F_{\phi,B}(z) - aby = 1.$$

Furthermore $_2\mathbf{F}_1(a,b;c;z) = F_{\phi,B}(z) + 1$. This shall be confirmed in a more general setting.

Define

$$_r\mathbf{F}_s(a_1,\ldots,a_r;b_1\ldots,b_s;z) = \sum_{n\geq 0}\frac{\prod_{i=1}^r\left(\frac{\Gamma(n+a_i)}{\Gamma(a_i)}\right)}{n!\prod_{j=1}^s\left(\frac{\Gamma(n+b_j)}{\Gamma(b_i)}\right)}z^n.$$

This function is called the *general hypergeometric function*.

Theorem 4.3.1. *Given the general hypergeometric function, $_r\mathbf{F}_s(a_1,\ldots,a_r;b_1\ldots,b_s;z)$, let*

$$G(k) = \prod_{i=1}^r(k+a_i-1)$$

and

$$H(k) = \prod_{i=1}^s(k+b_i-1).$$

Then $F(z) = {}_r\mathbf{F}_s(a_1,\ldots,a_r;b_1\ldots,b_s;z)$ solves the differential equation

$$H(1+\theta)(F') = G(1+\theta)(F). \tag{4.3.2}$$

Proof. For this proof we shall denote $_r\mathbf{F}_s(a_1,\ldots,a_r;b_1\ldots,b_s;z)$ with $F(z)$.

Let $\mathcal{L}$ be the *class of rooted lines* which satisfies $\mathcal{L} \cong \mathcal{Z} \times (1+\mathcal{L})$. Let

$$B_k = \frac{\prod_{i=1}^r(k+a_i-1)}{k\prod_{i=1}^s(k+b_i-1)}$$

then by the definitions of $F_{\mathcal{L},B}$ and $_r\mathbf{F}_s(a_1,\ldots,a_r;b_1\ldots,b_s;z)$, $F_{\mathcal{L},B}(z) = F(z) - 1$. By Corollary 4.2.2, $F_{\mathcal{L},B}$ solves the differential equation

$$F'_{\mathcal{L},B} = L_B(1+\theta)(1+F_{\mathcal{L},B})$$

and so F solves the differential equation

$$(F(z) - 1)' = L_B(1+\theta)(F(z)).$$

Since $L_B(k) = \frac{G(k)}{H(k)}$ we get that F solves

$$H(1+\theta)(F') = G(1+\theta)(F)$$

as desired. $\qquad\square$

When $r = 2, s = 1$, Equation 4.3.2 is equivalent to the hypergeometric differential equation (Equation 4.3.1).

Furthermore, Equation 4.3.2 is equivalent to the *general hypergeometric equation*,

$$\left(z\frac{d}{dz}(z\frac{d}{dz} + b_1 - 1)\cdots(z\frac{d}{dz} + b_s - 1) - z(z\frac{d}{dz} + a_1)\cdots(z\frac{d}{dz} + a_r)\right) F(z) = 0,$$

which appears in [49, §2.1] as the differential equation of the generalized Gauss function.

The results in this sections are not new. However, the proof of Theorem 4.3.1 views the general hypergeometric function as a series of weighted lines, which gives the general hypergeometric differential equation a slight combinatorial flavour. In Section 4.7 we will turn this idea around and look at combinatorially solutions to different differential equations.

4.4 New hook length formulae using Mazza's differential equation

In this section we will see how we can use Mazza's differential equation (Equation 4.2.1) to find new hook length formulae. This section contains five new hook length formulae proved using the differential equation.

Differential equations have been used in the hook length formula community to find hook length formulae [48, 23, 54]. The differential equations they used involve solutions that are power series. Differential equations were also used with increasing tree classes [39, 1]. However none of the combinatorialists investigating hook length formulae developed the differential equation to the generalization of Equation 4.2.1.

Example 4.4.1 (Formula 6.10.1). Consider $\mathcal{T} \cong \mathcal{Z} \times \text{SEQ}(\mathcal{T})^2$, the *class of fat plane trees*. A fat plane tree is a plane tree in which each node is given a color from 0 to the degree of the node. This class of trees is not widely used. We consider it here to demonstrate that how our methods can be applied to any simple tree class class with a computable combinatorial operator. The combinatorial operator for fat plane trees, $\text{SEQ}(\mathcal{Z})^2$, has the generating function $1 + 2z + 3z^2 + 4z^3 + \cdots$.

By Corollary 4.2.2 we have that $F = F_{\mathcal{T},\sigma}$ satisfies the differential equation:

$$F'(z) = \frac{1}{(1 - F(z))^2}.$$

The solution to this differential equation with $F(0) = 0$ is

$$F(z) = 1 - (1 - 3z)^{1/3}.$$

Therefore we have that:

$$\sum_{t \in \mathcal{T}_n} n! \prod_{v \in V(t)} \frac{1}{|t_v|} = \prod_{i=1}^{n-1}(3i - 1). \tag{4.4.1}$$

∎

57

Example 4.4.2 (Formula 6.11.6)**.** Consider the class, $\mathcal{R}$, of rooted unordered labelled trees. Let $B_k = 1 + \frac{1}{k}$. By Corollary 4.2.2 we have that $F = F_{\mathcal{R},B}$ satisfies the differential equation:

$$F'(z) = 2e^{F(z)} + z\frac{d}{dz}(e^{F(z)}).$$

Solving this differential equation using Maple, along with $F(0) = 0$, we get that:

$$F(z) = -2\log\left(\frac{1}{2}\left(1 + \sqrt{1-4z}\right)\right)$$

$$= 2\log\left(\frac{1}{1 - \frac{1-\sqrt{1-4z}}{2}}\right).$$

So,

$$\sum_{t\in\mathcal{R}_n} \frac{1}{2n!} \prod_{v\in V(t)} \left(1 + \frac{1}{|t_v|}\right) = \binom{2n+1}{n+1}. \tag{4.4.2}$$

$F(z)$ is also twice the exponential generating function for cycles of labelled Catalan objects and $F(z)$ is also twice the generating function of planar embedded trees (see [38] or [2, §3.1] for the definition). An open question is to find a bijection between one of these classes and the the class of bicolored labelled plane tree with one color increasing that the hook length formula counts. ∎

Example 4.4.3 (Formula 6.11.7)**.** Consider again $\mathcal{R} \cong \mathcal{Z} \times \mathrm{SET}(\mathcal{R})$, the class of rooted unordered labelled trees. Let $B_k = \frac{2}{k} - 1$. By Corollary 4.2.2 we have that $F = F_{\mathcal{R},B}$ satisfies the differential equation:

$$F'(z) = e^{F(z)} - z\frac{d}{dz}(e^{F(z)}).$$

Solving this differential equation using Maple, along with $F(0) = 0$, we get that:

$$F(z) = \log\left(z + \sqrt{z^2 + 1}\right).$$

Therefore

$$\sum_{t\in\mathcal{R}_{2n+1}} \prod_{v\in V(t)} \left(\frac{2}{|t_v|} - 1\right) = (-1)^n(2n-1)!!. \tag{4.4.3}$$

∎

Example 4.4.4 (Formula 6.14.2)**.** Consider $\mathcal{T} \cong \mathcal{Z} \times \mathrm{SET}_{\mathrm{even}}(\mathcal{T})$, the class of rooted unordered labelled trees with only vertices of even degree. Let $B = \frac{2}{k} - 1$. By Corollary 4.2.2 we have that $F = F_{\mathcal{T},B}$ satisfies the differential equation:

$$F'(z) = \cosh(F(z)) - z\frac{d}{dz}(\cosh(F(z))).$$

Solving this differential equation using Maple, along with $F(0) = 0$, we get that:

$$F(z) = \log\left(z + \sqrt{z^2 + 1}\right).$$

Therefore,

$$\sum_{t\in\mathcal{T}_{2n+1}} \prod_{v\in V(t)} \left(\frac{2}{|t_v|} - 1\right) = (-1)^n(2n-1)!!. \tag{4.4.4}$$

An interesting fact about this hook length formulae is that it is the same as the formula in Example 4.4.3 except with a different tree class. This because the trees of $\mathcal{R}$ that have a node with an odd degree have weights that cancel each other out. ∎

Example 4.4.5 (Formula 6.17.1). Consider $\mathcal{T} \cong \mathcal{Z} \times \text{SEQ}(\text{SET}_{\geq 1}(\mathcal{T}))$, the class of Schröder trees. By Corollary 4.2.2 we have that $F = F_{\mathcal{T},\sigma}$ satisfies the differential equation:

$$F'(z) = \frac{1}{2 - e^{F(z)}}.$$

Plugging $F(z) = \mathbf{W}\left(\frac{1}{2}\exp\left(\frac{z-1}{2}\right)\right) + \frac{z-1}{2}$ into the differential equation we get:

$$\frac{1}{2}\frac{\mathbf{W}\left(\frac{1}{2}\exp\left(\frac{z-1}{2}\right)\right)}{1 - \mathbf{W}\left(\frac{1}{2}\exp\left(\frac{z-1}{2}\right)\right)} + \frac{1}{2} = \frac{1}{2 - 2\mathbf{W}\left(\frac{1}{2}\exp\left(\frac{z-1}{2}\right)\right)},$$

which is satisfied. The hook length series, F, is also the generating function for phylogenetic trees. As we saw in Example 2.3.4 there is a bijection from increasing Schröder trees to phylogenetic trees. ∎

4.5 The differential equation for decorated trees and general hook length operators

Now we shall return to decorated trees. Using Theorem 2.5.2 we can see that a hook length series of decorated trees satisfies a similar differential equation.

Corollary 4.5.1. *Let $\phi(z,x)$ be a bivariate formal power series and B be a hook length, then $F_{\phi,B}$ satisfies the differential equation:*

$$F'_{\phi,B}(z) = L_B(1+\theta)\phi(z, F_{\phi,B}(z)).$$

Proof. Let $n \in \mathbb{N}$. Then $[z^n]F'_{\phi,B}(z) = (n+1)[z^{n+1}]F_{\phi,B}(z)$ and

$$[z^n]L_B(1+\theta)\phi(z, F_{\phi,B}(z)) = (n+1)B_{n+1}[z^n]\phi(z, F_{\phi,B}(z)).$$

By Theorem 2.5.2, $[z^{n+1}]F_{\phi,B}(z) = B_{n+1}[z^n]\phi(z, F_{\phi,B}(z))$ and therefore $F_{\phi,B}$ satisfies the differential equation. □

Example 4.5.2. This example, found by Leroux and Viennot [40], shows the simplicity of σ when ϕ is the product of two univariate functions of different variables. In this case the differential equation is separable.

Let $\phi(x)$ be a formal power series and let $\phi^*(z,x) = f(z)\phi(x)$. Let $F = F_{\phi,\sigma}$ and $F^* = F_{\phi^*,\sigma}$.

From Theorem 4.2.1 we have that F satisfies $F' = \phi(F)$ and from Corollary 4.5.1 we have that F^* satisfies

$$(F^*(z))' = \phi^*(z, F^*(z)) = f(z)\phi(F^*(z)). \tag{4.5.1}$$

Since $\frac{d}{dz}F(\int_0 f(z)) = F'(\int_0 f(z))f(z) = f(z)\phi(F(\int_0 f(z)))$, $F(\int_0 f(z))$ satisfies Equation 4.5.1. Since it is also the case that $F(\int_0 f(0)) = F(0) = 0 = F^*(0)$, we have that $F^*(z) = F(\int_0 f(z))$. ∎
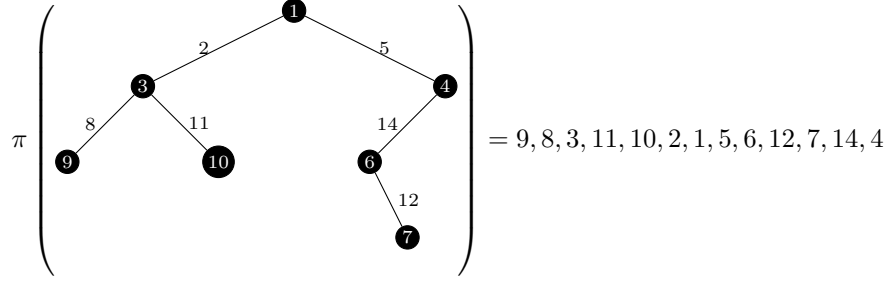
Figure 4.5.1: An example of the isomorphism π. The integers adjacent to the edges of the tree are the labels of the adjacent edges.

Next we have a specific example of a hook length formula of a decorated tree class. Note that the hook length operator in this example does not factor.

Example 4.5.3. This is a new hook length formula.

Let $\phi(z,x) = (1 + zx)^2$. Then $F = F_{\phi,\sigma}$ satisfies $F' = (1 + zF)^2$. Solving this DE we get $F(z) = \frac{\tanh(z)}{1 - z\tanh(z)}$. ∎

$(2n-1)![z^{2n-1}]F(z)$ in the above example counts the number of binary trees with n nodes where each node and each edge have a label with an added restriction. The restriction is that the label of each node is less than the labels of the nodes and edges below it (i.e. the tree is increasing). Note that $[z^{2n}]F(z) = 0$ so $F(z)$ is the exponential generating function of this tree class.

From the OEIS, $F(z)$ is also the exponential generating function of sequence A113583 [50]. This sequence counts the number of odd-sized permutations of length n that no local that have no local minimum at even positions, i.e. permutations π with $\pi_{2i} > \pi_{2i-1}$ or $\pi_{2i} > \pi_{2i+1}$ for all $i \leq \frac{n-1}{2}$ [37].

The isomorphism is given by

$$\pi(t) = \pi \left(\begin{array}{c} a \\ b \diagdown c \\ t_1 \quad t_2 \end{array} \right) = \pi(t_1)bac\pi(t_2)$$

where t is a binary tree with nodes and edges labelled with any $S \subseteq \mathbb{N}^+$ such that the label of each node is less than the labels below it. The isomorphism gives a permutation with no local minimum at even positions because the labels of the edges are put in even positions and the label of an edge is always less than the label of its parent which is either directly before or directly after the label of the edge. For an example see Figure 4.5.1.

The inverse isomorphism is

$$t(\pi) = t(\pi'bac\pi'') = \begin{array}{c} a \\ b \diagdown c \\ t(\pi') \quad t(\pi'') \end{array}$$

where a is the least element of $\pi'bac\pi''$ and π is a word with unique characters in $\mathbb{N}^+$ such that $\pi_{2i} > \pi_{2i-1}$ or $\pi_{2i} > \pi_{2i+1}$.

This isomorphism is an extension of Donaghey's isomorphism between binary increasing trees and permutations in Example 2.3.1. The proof that the isomorphism is indeed an isomorphism is similar.

We can also apply Corollary 2.6.1 to get a differential equation for general tree hook length series.

Corollary 4.5.4. *Let $\mathcal{T}$ be a class of decorated trees satisfying: $\mathcal{T} \cong \mathcal{Z} \times \Phi(\mathcal{Z}, \mathcal{T}, \mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(l)})$, where each $\mathcal{T}^{(i)}$ is a class of decorated trees and let B be a general hook length operator where $s, t \in \mathcal{T}_n$ implies $B_s = B_t$. Define*

$$F(z) = \sum_{t \in \mathcal{T}} B(t) \zeta^{(|t|)},$$

$$F^{(i)}(z) = \sum_{t \in \mathcal{T}^{(i)}} B(t) \zeta^{(|t|)}$$

and ϕ to be the formal power series of Φ. Then F satisfies the differential equation

$$F'(z) = L_B(1 + \theta)\phi(z, F(z), F^{(1)}(z), \ldots, F^{(l)}(z)),$$

where $\phi(z)$ is the formal power series of $\Phi(\mathcal{Z})$.

This differential equation is the foundation for the methods of Section 4.6. But now we shall look at three examples of new hook length formulae found using this corollary that do not use the new methods directly.

The hook length operators in the following examples are all general hook length operators as in Section 2.6. If a general hook length operator is an ordinary hook length operator — it depends only on the size of the subtrees — in subclasses of our tree class then we can break up our tree class into these subclasses. By applying the hook length operator to each subclass we can stay in the diferential equation framework and we will obtain a system of differential equations after applying Corollary 4.5.4. This is the main idea of the systems method of Section 4.6.2, which is used for ordinary hook length operators. Note that this technique was used in the example at the end of Section 2.6. The following examples are a preview of this method.

Example 4.5.5. This is a new hook length formula.

Let $\mathcal{B} \cong \mathcal{Z} \times (1 + \mathcal{B})^2$ and $B_t = \frac{1}{|t|}$ if the root of t has less than two children and $B_t = \frac{a}{|t|}$ if the root of t has exactly two children. We can split $\mathcal{B}$ into two classes: $\mathcal{B}^{(1)} \cong \mathcal{Z} \times (1 + 2\mathcal{B}^{(1)} + 2\mathcal{B}^{(2)})$, trees whose root has less than two children and $\mathcal{B}^{(2)} \cong \mathcal{Z} \times (\mathcal{B}^{(1)} + \mathcal{B}^{(2)})^2$, trees whose root has exactly two children. Let $F_1 = F_{\mathcal{B}^{(1)}, B}$ and $F_2 = F_{\mathcal{B}^{(2)}, B}$.

By Corollary 4.5.4, $F_1'(z) = 1 + 2F_1(z) + 2F_2(z)$ and $F_2'(z) = a(F_1(z) + F_2(z))^2$. Using Maple's `dsolve` function and simplifying we can see that $F_{\mathcal{B}, B}(z) = F_1(z) + F_2(z) = \frac{e^{2\alpha z} - 1}{\alpha + 1 + (\alpha - 1)e^{2\alpha z}}$, where $\alpha = \sqrt{1 - a}$.

In the special case where $a = 2$ we get: $F_{\mathcal{B}, B}(z) = \frac{\tan z}{1 - \tan z}$. This gives the hook length formula:

$$\sum_{t \in \mathcal{B}} z^{|t|} \prod_{v \in V(t) \setminus \mathscr{B}(t)} \frac{\deg(v)}{|t_v|} = \frac{\tan z}{1 - \tan z}. \tag{4.5.2}$$
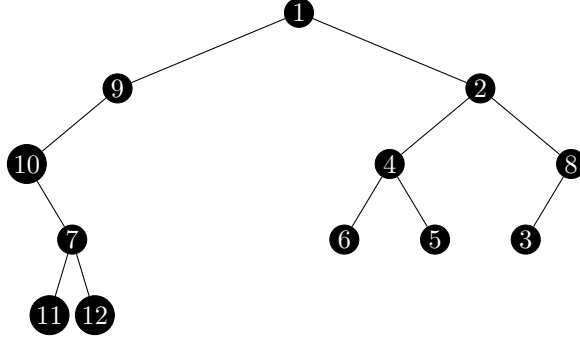
∎

61

Figure 4.5.2: An example of a labelled binary tree where the label of each node is greater than the label of its ancestors of degree 2.

Example 4.5.6. This is a new hook length formula.

Let $\mathcal{M} \cong \mathcal{Z} \times (1 + \mathcal{M} + \mathcal{M}^2)$ and

$$B_t = \begin{cases} a - \frac{a-2}{|t|} & \text{if } \deg(\text{root}(t)) = 1 \\ \frac{1}{|t|} & \text{otherwise.} \end{cases}$$

We split $\mathcal{M}$ into two classes: $\mathcal{M}^{(1)} \cong \mathcal{Z} \times \mathcal{M}$, the class of trees where the root has exactly one child and $\mathcal{M}^{(2)} \cong \mathcal{Z} \times (1 + \mathcal{M}^2)$, the class of trees where the root has zero or two children. Let $F_1 = F_{\mathcal{M}^{(1)},B}$, $F_2 = F_{\mathcal{M}^{(2)},B}$ and $F = F_{\mathcal{M},B}$.

By Corollary 4.5.4, $F_1'(z) = 2F(z) + azF'(z)$ and $F_2'(z) = 1 + F(z)^2$. Since $F(z) = F_1(z) + F_2(z)$, therefore

$$F'(z) = 1 + 2F(z) + F(z)^2 + azF'(z).$$

Plugging $F(z) = \frac{a}{a + \log(1-az)} - 1$ into the equation we can see that it is satisfied.

Taking $a = 2$ we can view this hook length as counting the number of labelled binary plane trees where the label of each node is greater than the label of its ancestors of degree 2. An example of such a tree is given in Figure 4.5.2. The counting sequence of this class was found by the author and added to the OEIS [50] as sequence A227917.

∎

Example 4.5.7 (Formula 6.5.9)**.** This is a new hook length formula.

Consider the class of Fibonacci trees, $\mathcal{T}$, which satisfies the combinatorial specification: $\mathcal{T} \cong \mathcal{Z} \times (\mathcal{T} + 1) \times (1 + \mathcal{Z})$. In this specification the last $\mathcal{Z}$ term indicates a node and not extra size for a node, therefore $\mathcal{T}$ is not a simple tree class. When dealing with hook length operators where $B_1 \neq 1$ we need to take special consideration of the last $\mathcal{Z}$ term. Let $\mathcal{T}_1$ be the class that only contains the tree that is just a leaf. Then $\mathcal{T} \cong \mathcal{Z} \times (\mathcal{T} + 1) \times (1 + \mathcal{T}_1)$. Let B_k be a hook length operator then $F_{\mathcal{T}_1,B}(z) = B_1 z$ and so by Corollary 4.5.4 $F = F_{\mathcal{T},B}$ satisfies the differential equation:

$$F'(z) = L_B(1 + \theta)((F(z) + 1)(1 + B_1 z)).$$

Let $B_k = 1 + \frac{1}{k}$. Then $F = F_{\mathcal{T},B}$ satisfies the differential equation:

$$F'(z) = 2(F(z) + 1)(1 + 2z) + z\frac{d}{dz}((F(z) + 1)(1 + 2z)).$$

Solving this we get $F(z) = \frac{1-2z}{((1-2z)^2(1+z))^{4/3}}$. $F(z)$ counts the number of labelled Fibonacci trees where each node is colored white or black and white nodes are increasing. ∎

4.6 New methods of finding hook length formulae

This section discusses new methods, using the differential equation, to find hook length formulae. In particular we will show how to use the differential equation for hook length operators that are piecewise using the leafless method (Section 4.6.1) or systems method (Section 4.6.2). By exploiting Theorem 3.2.5 we will also show how to use the differential equation for hook length operators with factors of r^k using the scaled method (Section 4.6.3).

4.6.1 Leafless method

The method of this subsection is called the leafless method and can be used when B_k is of the following form:

$$B_k = \begin{cases} a & \text{if } k = 1 \\ g(k) & \text{if } k > 1 \end{cases}$$

for some $a \in \mathbb{K}$ and $g : \mathbb{N}^+ \to \mathbb{K}$ with $g(1) \neq a$ if defined. The method is called the leafless method because we essentially ignore the leaves of the tree when constructing the differential equation.

For a tree class, $\mathcal{T} \cong \mathcal{Z} \times \Phi(\mathcal{T})$, we can split the class into two classes $\mathcal{T}_1$, the trees of size 1 and $\mathcal{T}^{(2)}$ the trees with size greater than 1.

Let Ψ be the combinatorial operator such that $\Psi(\mathcal{C}) = \{c \in \Phi(\mathcal{C}) : |c| > 0\}$ then $\psi(x) = \phi(x) - \phi_0$. Also

$$\mathcal{T}^{(2)} \cong \mathcal{Z} \times \Psi(\mathcal{T}),$$

$$\mathcal{T}_1 \cong \phi_0 \mathcal{Z}$$

and

$$\mathcal{T} \cong \mathcal{T}^{(2)} + \mathcal{T}_1.$$

Therefore by Corollary 4.5.4, $F_{\mathcal{T}',B}$ satisfies the differential equation:

$$F'_{\mathcal{T}^{(2)},B}(z) = L_B(1 + \theta)\psi(F_{\mathcal{T},B}).$$

Since $\psi(x) = \phi(x) - \phi_0$, $F_{\mathcal{T},B} = F_{\mathcal{T}^{(2)},B} + F_{\mathcal{T}_1,B}$ and $F_{\mathcal{T}_1,B} = B_1\phi_0 z$, we have

$$F'_{\mathcal{T},B}(z) - B_1\phi_0 = L_B(1 + \theta)(\phi(F_{\mathcal{T},B}) - \phi_0).$$

The following three examples illustrate how to use the leafless method.

Example 4.6.1 (Formula 6.6.4)**.** This is a new hook length formula.

Consider $\mathcal{M} \cong \mathcal{Z} \times (1 + \mathcal{M} + \mathcal{M}^2)$, the class of Motzkin trees. Let

$$
B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{k-1} & \text{if } k > 1. \end{cases}
$$

We need the leafless method in this example because $\frac{1}{k-1}$ is undefined for $k = 1$. Using the leafless method we get that $L_B(k) = \frac{k}{k-1}$. Then $F_{\mathcal{M},B}$ satisfies the differential equation:

$$
z \frac{d^2}{dz^2} F_{\mathcal{M},B}(z) = F_{\mathcal{M},B}(z) + F_{\mathcal{M},B}(z)^2 + z \frac{d}{dz}(F_{\mathcal{M},B}(z) + F_{\mathcal{M},B}(z)^2).
$$

Let $F(z) = \frac{z}{1-z}$. Plugging $F = F_{\mathcal{M},B}$ into the differential equation above we can see that it is satisfied. Since $F(0) = 0 = F_{\mathcal{M},B}(0)$ and $F'(0) = 1 = F'_{\mathcal{M},B}(0)$, $F = F_{\mathcal{M},B}$. Therefore for all $n \geq 1$

$$
\sum_{t \in \mathcal{M}_n} \prod_{v \in V(t) \setminus \varnothing(t)} \frac{1}{|t_v| - 1} = 1. \tag{4.6.1}
$$

∎

Example 4.6.2 (Formula 6.16.2)**.** This is a new hook length formula.

Consider $\mathcal{C} = \mathcal{Z} \times (1 + \text{CYC}(\mathcal{C}))$, the class of cyclic trees. Let

$$
B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{k} - 1 & \text{if } k > 1. \end{cases}
$$

Again we need the leafless method in this example because $\frac{1}{k} - 1$ equals 0 when $k = 1$, but $B_1 = 1$. Using the leafless method we get that $F = F_{\mathcal{C},B}$ satisfies the differential equation:

$$
\frac{d}{dz} F(z) - 1 = -z \frac{d}{dz}(-\log(1 - F(z))).
$$

Using Maple, with $F(0) = 0$, we get that $F(z) = 1 + z - \sqrt{1 + z^2}$. Therefore

$$
\sum_{t \in \mathcal{C}_{2n}} \frac{2^{2n-1}}{n!} \prod_{v \in V(t) \setminus \varnothing(t)} \left(\frac{1}{|t_v|} - 1 \right) = \frac{(-1)^n}{n} \binom{2n}{n}. \tag{4.6.2}
$$

∎

Example 4.6.3 (Formula 6.16.3)**.** This is a new hook length formula.

Again consider the class of cyclic trees, $\mathcal{C}$. Let

$$
B_k = \begin{cases} 1 & \text{if } k = 1 \\ B_k = k - 1 & \text{if } k > 1. \end{cases}
$$

Then we can show that

$$
F_{\mathcal{C},B}(z) = 1 - \frac{1}{\sum_{n \geq 0} n! z^n}. \tag{4.6.3}
$$

64

We need the leafless method in this example because $k - 1$ equals 0 when $k = 1$, but $B_1 = 1$. Using the leafless method we get that $F_{\mathcal{C},B}$ satisfies the differential equation

$$\frac{d}{dz}F_{\mathcal{C},B}(z) - 1 = z^2\frac{d^2}{dz^2}(-\log(1 - F_{\mathcal{C},B}(z))) + 2z\frac{d}{dz}(-\log(1 - F_{\mathcal{C},B}(z))).$$

Since $\sum_{n\geq 0} n!z^n = {}_2\mathbf{F}_0(1,1;;z)$, substituting Equation 4.6.3 into the differential equation we get:

$$\frac{\frac{d}{dz}{}_2\mathbf{F}_0(1,1;;z)}{{}_2\mathbf{F}_0(1,1;;z)^2} - 1 = z^2\frac{{}_2\mathbf{F}_0(1,1;;z)\frac{d^2}{dz^2}{}_2\mathbf{F}_0(1,1;;z) - (\frac{d}{dz}{}_2\mathbf{F}_0(1,1;;z))^2}{{}_2\mathbf{F}_0(1,1;;z)^2} + 2z\frac{\frac{d}{dz}{}_2\mathbf{F}_0(1,1;;z)}{{}_2\mathbf{F}_0(1,1;;z)}.$$

From Theorem 4.3.1 we know that ${}_2\mathbf{F}_0(1,1;;z)$ satisfies the differential equation
$\frac{d}{dz}{}_2\mathbf{F}_0(1,1;;z) = {}_2\mathbf{F}_0(1,1;;z) + 3z\frac{d}{dz}{}_2\mathbf{F}_0(1,1;;z) + z^2\frac{d^2}{dz^2}{}_2\mathbf{F}_0(1,1;;z)$. We can use this to simplify to:

$$\frac{\frac{d}{dz}{}_2\mathbf{F}_0(1,1;;z)\left(-1 - z^2\frac{d}{dz}{}_2\mathbf{F}_0(1,1;;z) + (1-z){}_2\mathbf{F}_0(1,1;;z)\right)}{{}_2\mathbf{F}_0(1,1;;z)^2} = 0.$$

Since

$$
\begin{aligned}
-1 - z^2\frac{d}{dz}{}_2\mathbf{F}_0(1,1;;z) + (1-z){}_2\mathbf{F}_0(1,1;;z) &= -1 - z^2\frac{d}{dz}\sum_{n\geq 0}n!z^n + (1-z)\sum_{n\geq 0}n!z^n \\
&= -1 - z^2\sum_{n\geq 1}nn!z^{n-1} + (1-z)\sum_{n\geq 0}n!z^n \\
&= -1 - \sum_{n\geq 2}(n-1)(n-1)!z^n + \\
&\qquad \sum_{n\geq 0}n!z^n - \sum_{n\geq 1}(n-1)!z^n \\
&= -\sum_{n\geq 2}(n-1)(n-1)!z^n + \sum_{n\geq 2}n!z^n - \sum_{n\geq 2}(n-1)!z^n \\
&= \sum_{n\geq 2}(-(n-1)(n-1)! + n(n-1)! - (n-1)!)z^n \\
&= 0
\end{aligned}
$$

and ${}_2\mathbf{F}_0(1,1;;z)$ is nonzero, we have that the equation is satisfied. This hook length series is also the ordinary generating function of connected permutations. The hook length, B, counts the number of trees where each internal node has an extra edge that points to one of its descendants. See Example 2.3.7 for the isomorphism between these two classes. $\blacksquare$

4.6.2 System method

Suppose B_k is piecewise with $P_1 \cup \cdots \cup P_m = \mathbb{N}^+$ and $B_k = B_k^{(i)}$ whenever $k \in P_i$. Furthermore, suppose that $L_{B^{(i)}} \neq L_{B^{(j)}}$ for all $i \neq j$. Then we cannot use the differential equation from Corollary 4.2.2 to find hook length formulae.

However, if the target tree class, $\mathcal{T}$, can easily be seperated into tree classes $\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(m)}$ such that $\mathcal{T}^{(i)} \cong \mathcal{Z} \times \Phi_i(\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(m)})$ and the sizes of trees in each $\mathcal{T}_i$ lie in P_i then we can use Corollary 2.6.1 to

find the hook length formulae. From the theorem we know that $F_{\mathcal{T}^{(i)},B}$ solves the differential equation

$$F'_{\mathcal{T}^{(i)},B} = L_B^{(i)}(1+\theta)\Phi_i(F_{\mathcal{T}^{(1)},B},\ldots,F_{\mathcal{T}^{(m)},B}).$$

This leaves us with a system of differential equations. If we solve this system we can get $F_{\mathcal{T},B}$ by adding the $F'_{\mathcal{T}^{(i)},B}$ together. We call this method the system method.

Note that the system method is a generalization of the leafless method where $P_1 = \{1\}$, $P_2 = \mathbb{N}^+ \setminus \{1\}$, $\mathcal{T}_1 \cong \phi_0 \mathcal{Z}$ and $\mathcal{T}_2 \cong \mathcal{Z} \times (\Phi(\mathcal{T}) - \phi_0)$.

We could not find any new hook length formulae that use the system method without using the scaled method of the next subsection. The example we provide shows how to use the system to reprove an old hook length formula. Example 4.6.6 shows how to use the system method in combination with the scaled method.

Example 4.6.4. The following example was found by Han in 2008 by using a precursor to the Kuba-Panholzer recurrence [27]. Here we shall use the system method instead.

Consider $\mathcal{B} \cong \mathcal{Z} \times (1+\mathcal{B})^2$, the class of binary trees and the hook length operator B given by:

$$B_k = \begin{cases} 1 & \text{if } k \text{ is odd} \\ \text{-}\frac{1}{k} & \text{if } k \text{ is even.} \end{cases}$$

We shall split the class of binary trees into two classes, $\mathcal{B}_{\text{odd}}$ and $\mathcal{B}_{\text{even}}$, the classes of odd and even sized binary trees respectively. It follows that that $\mathcal{B}_{\text{odd}}$ satisfies the specification:

$$\mathcal{B}_{\text{odd}} \cong \mathcal{Z} \times ((1+\mathcal{B}_{\text{even}})^2 + \mathcal{B}_{\text{odd}}^2)$$

and $\mathcal{B}_{\text{even}}$ satisfies the specification:

$$\mathcal{B}_{\text{even}} \cong \mathcal{Z} \times 2\mathcal{B}_{\text{odd}} \times (1+\mathcal{B}_{\text{even}}).$$

From the system method we get that $F_o = F_{\mathcal{B}_{\text{odd}},B}$ and $F_e = F_{\mathcal{B}_{\text{even}},B}$ satisfy the differential equations:

$$F'_o(z) = (1+F_e(z))^2 + F_o(z)^2 + z\frac{d}{dz}((1+F_e(z))^2 + F_o(z)^2)$$

and

$$F'_e(z) = \text{-}2F_o(z)(1+F_e(z)).$$

Han showed that $F_{\mathcal{B},B}(z) = \frac{z-z^2}{1+z^2}$. By substituting $F_o(z) = \frac{z}{1+z^2}$ and $F_e(z) = \text{-}\frac{z^2}{1+z^2}$ in the equations we can see that they are satisfied. ∎

4.6.3   Scaled method

Suppose we have a hook length operator, B, with $B_k = r^{k-1}C_k$, where C is also hook length operator. Then we cannot use Mazza's differential equation because $L_B(k) = r^{k-1}L_C(k)$ and

$$L_B(x) = \sum_{i=0}^{\infty}\sum_{j=0}^{i}\binom{m}{i}\frac{(\ln r)^i}{ri!}c_{m-i}x^m$$

which does not give an easily solvable differential equation.

However we know from Theorem 3.2.5 that $L_B^*(x(z)) = L_C^*(x(rz))$. Thus

$$\int_0 L_B(1+\theta)(x(z)) = L_B^*(x(z)) = L_C^*(x(rz)) = \int_0 L_C(1+\theta)(x(rz))$$

and so

$$L_B(1+\theta)(x(z)) = L_C(1+\theta)(x(rz)).$$

Therefore we know that for a power series, ϕ, $F_{\phi,B} = F$ solves the differential equation

$$F'(z) = L_B(1+\theta)(\phi(F(z))) = L_C(1+\theta)(\phi(F(rz))).$$

We call this method the scaled method since we scale the hook length series to r.

The following four examples show how to use the scaled method. They all use either the leafless method or system method in combination with the scaled method.

Example 4.6.5 (Formula 6.9.4). This is a new hook length formula.

Let

$$B_k = \begin{cases} 1 & \text{if } k = 1 \\ 2^{2-k} & \text{if } k > 1. \end{cases}$$

Then $r = \frac{1}{2}$ and

$$C_k = \begin{cases} 1 & \text{if } k = 1 \\ 2 & \text{if } k > 1. \end{cases}$$

Then by the leafless method $F_{\mathcal{O},C}$ satisfies the differential equation:

$$F'_{\mathcal{O},C}(z) - 1 = 2\left(\frac{F_{\mathcal{O},C}(z)}{1 - F_{\mathcal{O},C}(z)} + z\frac{d}{dz}\left(\frac{F_{\mathcal{O},C}(z)}{1 - F_{\mathcal{O},C}(z)}\right)\right).$$

By the scaled method we have that $F = F_{\mathcal{O},B}$ satisfies the differential equation:

$$F'(z) - 1 = 2\left(\frac{F(z/2)}{1 - F(z/2)} + z\frac{d}{dz}\left(\frac{F(z/2)}{1 - F(z/2)}\right)\right).$$

Plugging in $F(z) = \frac{z}{1-z}$ we get:

$$\frac{1}{(1-z)^2} - 1 = 2\left(\frac{\frac{z/2}{1-z/2}}{1 - \frac{z/2}{1-z/2}} + \frac{z}{2(1-z/2)^2(1 - \frac{z/2}{1-z/2})^2}\right).$$

Simplifying we see that the differential equation is satisfied and $F = F_{\mathcal{O},B}$. Therefore

$$\sum_{t \in \mathcal{O}_n} 4^n \prod_{v \in V(t) \setminus \mathscr{D}(t)} \frac{1}{2^{|t_v|}} = 1 \tag{4.6.4}$$

∎

67

Example 4.6.6 (Formula 6.9.4). This is a new hook length formula.

Consider the class of plane trees, $\mathcal{O} \cong \mathcal{Z} \times \mathrm{SEQ}(\mathcal{O})$. Let

$$B_k = \begin{cases} 1 & \text{if } k \leq 2 \\ \frac{1}{2} & \text{if } k \equiv 1,2 \mod 4, k > 2 \\ -\frac{1}{2} & \text{if } k \equiv 0,3 \mod 4. \end{cases}$$

We can rewrite this hook length as:

$$B_k = \begin{cases} 1 & \text{if } k = 1 \\ 1 & \text{if } k = 2 \\ \frac{1}{2}\iota^{k-1} & \text{if } k > 2 \text{ is odd} \\ \frac{1}{2\iota}\iota^{k-1} & \text{if } k > 2 \text{ is even} \end{cases}$$

where $\iota = \sqrt{-1}$ is the imaginary number. We can split the class of plane trees into three classes $\mathcal{O}_2$, the class of trees of size 2, $\mathcal{O}_o$ the class of odd sized trees and $\mathcal{O}_e$, the class of even sized trees of size greater than 2. Then $\mathcal{O}_2 \cong \mathcal{Z}^2$, $\mathcal{O}_o \cong \mathcal{Z} \times \mathrm{SEQ}_{\mathrm{even}}(\mathrm{SEQ}(\mathcal{O}_e + \mathcal{O}_2) \times \mathcal{O}_o) \times \mathrm{SEQ}(\mathcal{O}_e + \mathcal{O}_2)$ and $\mathcal{O}_e \cong \mathcal{Z} \times \mathrm{SEQ}_{\mathrm{odd}}(\mathrm{SEQ}(\mathcal{O}_e+\mathcal{O}_2) \times \mathcal{O}_o) \times \mathrm{SEQ}(\mathcal{O}_e+\mathcal{O}_2)$. We can see that $O_o(z) = z\frac{1 - O_e(z) - O_e(z)}{(1-O_e(z)-O_2(z))^2 - O_o(z)^2}$ and $O_e(z) = z\frac{O_o(z)}{(1-O_e(z)-O_2(z))-O_o(z)^2}$, where O_o, O_e and O_2 are the generating functions of $\mathcal{O}_o$, $\mathcal{O}_e$ and $\mathcal{O}_2$ respectively. Let $F_2 = F_{\mathcal{O}_2,B}$, $F_o = F_{\mathcal{O}_o,B}$ and $F_e(\mathcal{O}_e, B)$.

Using the system method together together with the scaled method and the leafless method on $\mathcal{O}_o$, we get the following differential equations:

$$F_2(z) = z^2,$$

$$F_o'(z) - 1 = \frac{1}{2}\left(\frac{1 - F_e(\iota z) - F_2(\iota z)}{(1 - F_e(\iota z) - F_2(\iota z))^2 - F_o(\iota z)^2} + z\frac{d}{dz}\left(\frac{1 - F_e(\iota z) - F_2(\iota z)}{(1 - F_e(\iota z) - F_2(\iota z))^2 - F_o(\iota z)^2} \right) \right)$$

and

$$F_e'(z) = \frac{1}{2\iota}\left(\frac{F_o(\iota z)}{(1 - F_e(\iota z) - F_2(\iota z))^2 - F_o(\iota z)^2} + z\frac{d}{dz}\left(\frac{F_o(\iota z)}{(1 - F_e(\iota z) - F_2(\iota z))^2 - F_o(Iz)^2} \right) \right).$$

Substituting $F_o(z) = \frac{z}{1+z^2}$ and $F_e(z) = -\frac{z^4}{1+z^2}$ we can see that the equations are satisfied. Since $F_2(z) + F_o(z) + F_e(z) = z^2 + \frac{z}{1+z^2} - \frac{z^4}{1+z^2} = \frac{z+z^2}{1+z^2}$, we get that $F_{\mathcal{O},B}(z) = \frac{z+z^2}{1+z^2}$.

In this example it is actually simpler to prove the formula using the Kuba-Panholzer recurrence Now we shall illustrate this. Let $F(z) = \frac{z+z^2}{1+z^2}$ then

$$[z^k]F(z) = \begin{cases} 1 & \text{if } k \equiv 1,2 \mod 4 \\ -1 & \text{if } k \equiv 0,3 \mod 4 \end{cases}$$

and

$$[z^{k-1}]\frac{1}{1-F(z)} = [z^{k-1}]\frac{1}{1-\frac{z+z^2}{1+z^2}} = [z^{k-1}]\frac{1+z^2}{1-z} = \begin{cases} 1 & \text{if } k \leq 2 \\ 2 & \text{if } k > 2. \end{cases}$$

So $\frac{[z^k]F(z)}{[z^{k-1}]\frac{1}{1-F(z)}} = B_k$ and by Theorem 2.2.4, we have that $F(z) = F_{\mathcal{O},B}(z)$. ∎

Example 4.6.7 (Formula 6.10.2)**.** This is a new hook length formula.

Consider $\mathcal{T} \cong \mathcal{Z} \times \mathrm{SEQ}(\mathcal{T})^2$, the class of fat plane trees and let

$$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{2^{k-3}(k+2)} & \text{if } k > 1 \end{cases}$$

Then $r = \frac{1}{2}$ and

$$C_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{4}{k+2} & \text{if } k > 1 \end{cases}$$

By the scaled and leafless methods we have that $F = F_{\mathcal{T},B}$ satisfies the differential equation:

$$z F''(z) + 3F'(z) - 3 = 4 \left(\frac{1}{(1 - F(z/2))^2} - 1 + z \frac{d}{dz} \left(\frac{1}{(1 - F(z/2))^2} - 1 \right) \right).$$

Plugging in $F(z) = \frac{z}{1-z}$ we get:

$$\frac{2z}{(1-z)^3} + \frac{3}{(1-z)^2} - 3 = 4 \left(\frac{1}{\left(1 - \frac{z/2}{1-z/2}\right)^2} - 1 + \frac{z}{(1 - z/2)^2 \left(1 - \frac{z/2}{1-z/2}\right)^3} \right).$$

Simplifying we see that the differential equation is satisfied and $F = F_{\mathcal{T},B}$. Therefore

$$\sum_{t \in \mathcal{T}_n} 8^n \prod_{v \in V(t) \setminus \mathscr{D}(v)} \frac{1}{2^{|t_v|}(|t_v| + 2)} = 1. \tag{4.6.5}$$

∎

Example 4.6.8 (Formula 6.16.4)**.** This is a new hook length formula.

Consider $\mathcal{C} \cong \mathcal{Z} \times (1 + \mathrm{CYC}(\mathcal{C}))$, the class of cyclic trees and

$$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{k-1}{2^{k-1}-1} & \text{if } k > 1 \end{cases}.$$

This example is more complicated than the previous examples because 2^{k-1} is not a factor of B_k, but is instead inside a factor.

However we can write $L_B(k)$ as $L_B(k) = \frac{k(k-1)}{L_C(k)-1}$, where $C_k = 2^{k-1}$.

By the leafless $F = F_{\mathcal{C},B}$ satisfies the differential equation:

$$L_C(1 + \theta)F'(z) - F'(z) = -2z \frac{d}{dz} \log(1 - F(z)) - z^2 \frac{d^2}{dz^2} \log(1 - F(z)).$$

Since $L_C(1 + \theta)g(z) = g(2z)$ we get that F satisfies the differential equation:

$$F'(2z) - F'(z) = -2z \frac{d}{dz} \log(1 - F(z)) - z^2 \frac{d^2}{dz^2} \log(1 - F(z)).$$

Plugging in $F(z) = \frac{z}{1-z}$ we get:

$$\frac{1}{1-2z} - \frac{1}{1-z} = \frac{2z}{(1-z)(1-2z)} + \frac{z^2(3-4z)}{(1-z)^2(1-2z)^2}.$$

Simplifying we see that the differential equation is satisfied and $F = F_{\mathcal{C},B}$. Therefore

$$\sum_{t \in \mathcal{C}_n} \prod_{v \in V(t)} \frac{|t_v| - 1}{2^{|t_v|-1} - 1} = n!. \tag{4.6.6}$$

∎

4.7 Representing differential equations combinatorially

In this section we use Mazza's differential equation and Corollary 4.5.4 to represent some differential equation solutions as hook length series. This can be considered solving the differential equations combinatorially.

Finding combinatorial solutions to differential equations was indirectly done by Butcher when he was developing his analysis of the Runge-Kutta methods [6, 7]. Leroux and Viennot [39] independently set out to find combinatorial solutions for the same differential equation as Butcher,

$$y'(z) = H(z, y(z)).$$

They discovered that when H only depends on $y(z)$, the solution for y is the class of increasing labellings of the class $\mathcal{T} \cong \mathcal{Z} \times H(\mathcal{T})$. In this section we will extend their findings for all of the differential equations that Corollary 4.5.4 covers.

Leroux and Viennot also found combinatorial interpretations of systems of differential equations. They interpreted higher-order differential equation by reducing them to systems of differential equation in the form of Section 4.7. This is different from how we will handle higher-order differential equations since higher derivatives can be generated by the L_B operator. We will not be discussing systems of differential equations.

4.7.1 More on the theta operator

Before we solve differential equations combinatorially, first we need to investigate the theta operator, $\theta = z\frac{d}{dz}$. Our goal is to be able to convert differential equations without θ into equations with θ so we can use Corollary 4.5.4. The following propositions will lead us to this goal.

Proposition 4.7.1. *For $n \in \mathbb{N}$:*

$$\theta^n = \sum_{i=0}^{n} \begin{Bmatrix} n \\ i \end{Bmatrix} z^i \frac{d^i}{dz^i} \tag{4.7.1}$$

$$(1+\theta)^n = \sum_{i=0}^{n} \begin{Bmatrix} n+1 \\ i+1 \end{Bmatrix} z^i \frac{d^i}{dz^i} \tag{4.7.2}$$

where $\left\{{a \atop b}\right\}$ are the Stirling numbers of the second kind.

Proof. Both equations follow from the recurrence $\left\{{a+1 \atop b}\right\} = b\left\{{a \atop b}\right\} + \left\{{a \atop b-1}\right\}$ for all $a, b \geq 1$.

For Equation 4.7.1, we can see that: $\theta^0 = 1 = \left\{{0 \atop 0}\right\} z^0 \frac{d^0}{dz^0}$.

Assume that the result holds for n. Then,

$$\theta^{n+1} = z \frac{d}{dz} \sum_{i=0}^{n} \left\{{n \atop i}\right\} z^i \frac{d^i}{dz^i}$$

$$= \sum_{i=0}^{n} \left(i \left\{{n \atop i}\right\} z^i \frac{d^i}{dz^i} + \left\{{n \atop i}\right\} z^{i+1} \frac{d^{i+1}}{dz^{i+1}} \right)$$

$$= \sum_{i=0}^{n} \left(i \left\{{n \atop i}\right\} + \left\{{n \atop i-1}\right\} \right) z^i \frac{d^i}{dz^i} + \left\{{n \atop n}\right\} z^n \frac{d^n}{dz^n}$$

$$= \sum_{i=0}^{n} \left\{{n+1 \atop i}\right\} z^i \frac{d^i}{dz^i} + \left\{{n+1 \atop n+1}\right\} z^n \frac{d^n}{dz^n}$$

$$= \sum_{i=0}^{n+1} \left\{{n+1 \atop i}\right\} z^i \frac{d^i}{dz^i}.$$

Similarly for Equation 4.7.2, we get: $(\theta + 1)^0 = 1 = \left\{{1 \atop 1}\right\} z^0 \frac{d^0}{dz^0}$.

Assume that the result holds for n. Then,

$$(1 + \theta)^{n+1} = \left(1 + z \frac{d}{dz}\right) \sum_{i=0}^{n} \left\{{n+1 \atop i+1}\right\} z^i \frac{d^i}{dz^i}$$

$$= \sum_{i=0}^{n} \left(\left\{{n+1 \atop i+1}\right\} z^i \frac{d^i}{dz^i} + i \left\{{n \atop i+1}\right\} z^i \frac{d^i}{dz^i} + \left\{{n+1 \atop i+1}\right\} z^{i+1} \frac{d^{i+1}}{dz^{i+1}} \right)$$

$$= \sum_{i=0}^{n} \left((i+1) \left\{{n+1 \atop i+1}\right\} + \left\{{n+1 \atop i}\right\} \right) z^i \frac{d^i}{dz^i} + \left\{{n+1 \atop n+1}\right\} z^n \frac{d^n}{dz^n}$$

$$= \sum_{i=0}^{n} \left\{{n+2 \atop i+1}\right\} z^i \frac{d^i}{dz^i} + \left\{{n+2 \atop n+2}\right\} z^n \frac{d^n}{dz^n}$$

$$= \sum_{i=0}^{n+1} \left\{{n+2 \atop i+1}\right\} z^i \frac{d^i}{dz^i}.$$

$\square$

We use the previous proposition to obtain the following identity.

Proposition 4.7.2. *For $n \in \mathbb{N}$:*

$$z^n \frac{d^n}{dz^n} = \sum_{i=0}^{n} s(n+1, i+1)(1+\theta)^i \tag{4.7.3}$$

where $s(a,b)$ are the signed Stirling numbers of the first kind.

Proof. Proof by induction on n.

For $n = 0$, $z^0 \frac{d^0}{dz^0} = 1 = s(1,1)(1+\theta)^0$.

For $n \geq 1$,

$$z^n \frac{d^n}{dz^n} = (1+\theta)^n - \sum_{i=0}^{n-1} \left\{ \begin{matrix} n+1 \\ i+1 \end{matrix} \right\} z^i \frac{d^i}{dz^i} \qquad \text{by Equation 4.7.2}$$

$$= (1+\theta)^n - \sum_{i=0}^{n-1} \left\{ \begin{matrix} n+1 \\ i+1 \end{matrix} \right\} \sum_{j=0}^{i} s(i+1,j+1)(1+\theta)^j \qquad \text{by induction}$$

$$= s(n+1,n+1)(1+\theta)^n - \sum_{j=0}^{n-1} \left(\sum_{i=j}^{n-1} s(i+1,j+1) \left\{ \begin{matrix} n+1 \\ i+1 \end{matrix} \right\} \right) (1+\theta)^j$$

$$= s(n+1,n+1)(1+\theta)^n + \sum_{j=0}^{n-1} s(n+1,j+1)(1+\theta)^j$$

$\square$

Proposition 4.7.3. *For $n \in \mathbb{N}$ and $f : \mathbb{K} \to \mathbb{K}$, n-times differentiable,*

$$z^n f(z) \frac{d^n}{dz^n} = \sum_{i=0}^{n} z^i \frac{d^i}{dz^i} g_i(z) \qquad (4.7.4)$$

for some $g_i(z) : \mathbb{K} \to \mathbb{K}$, i-times differentiable.

Before we prove this proposition we shall prove the following lemma:

Lemma 4.7.4. *For $n \in \mathbb{N}$ and $f : \mathbb{K} \to \mathbb{K}$, n-times differentiable,*

$$f(z) \frac{d^n}{dz^n} = \sum_{i=0}^{n} \frac{d^i}{dz^i} g_i(z) \qquad (4.7.5)$$

for some $g_i(z) : \mathbb{K} \to \mathbb{K}$, i-times differentiable.

Proof. The proof is by induction on n.

If $n = 0$ then $f(z) \frac{d^0}{dz^0} = f(z) = \frac{d^0}{dz^0} f(z)$.

If $n \geq 1$ then

$$f(z) \frac{d^n}{dz^n} = \frac{d}{dz} f(z) \frac{d^{n-1}}{d^{n-1}} + f'(z) \frac{d^{n-1}}{d^{n-1}} \qquad \text{by product rule}$$

$$= \frac{d}{dz} \sum_{i=0}^{n-1} \frac{d^i}{dz^i} g_i(z) + \sum_{i=0}^{n-1} \frac{d^i}{dz^i} h_i(z) \qquad \text{by the induction hypothesis}$$

$$= \frac{d^0}{dz^0} h_0(z) + \sum_{i=1}^{n-1} \frac{d^i}{dz^i} (f_{i-1}(z) + h_i(z)) + \frac{d^n}{dz^n} g_{n-1}(z).$$

By $\qquad$ $\square$

proof of Proposition 4.7.3. The proof is by induction on n.

If $n = 0$ then $z^0 f(z)\frac{d^0}{dz^0} = f(z) = z^0 \frac{d^0}{dz^0} f(z)$.

If $n \geq 1$ then

$$
\begin{aligned}
z^n f(z)\frac{d^n}{dz^n} &= z^n \frac{d}{dz} f(z)\frac{d^{n-1}}{d^{n-1}} + z^n f'(z)\frac{d^{n-1}}{d^{n-1}} && \text{by product rule} \\
&= z^n \frac{d}{dz} \sum_{i=0}^{n-1} \frac{d^i}{dz^i} k_i(z) + z^n f'(z)\frac{d^{n-1}}{d^{n-1}} && \text{by Lemma 4.7.4} \\
&= z^n \frac{d^n}{dz^n} k_{n-1}(z) + \sum_{i=0}^{n-2} z^n \frac{d^{i+1}}{dz^{i+1}} k_i(z) + z^n f'(z)\frac{d^{n-1}}{d^{n-1}} \\
&= z^n \frac{d^n}{dz^n} k_{n-1}(z) + \sum_{i=1}^{n-1} \sum_{j=0}^{i} z^j \frac{d^j}{dz^j} h_{i,j}(z) + z^n f'(z)\frac{d^{n-1}}{d^{n-1}} && \text{by the induction hypothesis} \\
&= z^n \frac{d^n}{dz^n} k_{n-1}(z) + \sum_{i=1}^{n-1} \sum_{j=0}^{i} z^j \frac{d^j}{dz^j} h_{i,j}(z) + \sum_{i=0}^{n-1} z^i \frac{d^i}{dz^i} p_i(z) && \text{by the induction hypothesis} \\
&= \sum_{i=0}^{n-1} z^i \frac{d^i}{dz^i} \left(p_i(z) + \sum_{j=i}^{n-1} h_{j,i}(z) \right) + z^n \frac{d^n}{dz^n} k_{n-1}(z).
\end{aligned}
$$

Setting $g_i = p_i(z) + \sum_{j=i}^{n-1} h_{j,i}(z)$ for $i = 1, \ldots, n-1$ and $g_n(z) = k_{n-1}(z)$, we are done. $\qquad$ $\square$

4.7.2 Combinatorial solutions to differential equations

From the propositions in the preceding subsection we get the following result.

Theorem 4.7.5. *Let*

$$
e(F, z) = \sum_{i=0}^{N} z^i f_i(z)\frac{d^i}{dz^i} \phi_i(z, F(z))
$$

with $f_i(z) \in \mathbb{K} \to \mathbb{K}$, $\phi_i : \mathbb{K} \times \mathbb{K} \to \mathbb{K}$, both i-times differentiable. Then

$$
e(F, z) = \sum_{i=0}^{N} (1 + \theta)^i \psi_i(z, F(z)),
$$

for some $\psi_i : \mathbb{K} \times \mathbb{K} \to \mathbb{K}$, i-times differentiable with respect to z.

Proof. This follows by applying Proposition 4.7.3 to each summand and then applying Proposition 4.7.2 to each of the resulting summands. $\qquad$ $\square$

Let us see how we can view the latter differential equation combinatorially under some mild hypotheses. If $e(F, z) = \sum_{j=0}^{M} a_j(1 + \theta)^j F'(z)$ for some M and a_i then we have the equation:

$$\sum_{j=0}^{M} a_j(1 + \theta)^j F'(z) = \sum_{i=0}^{N} (1 + \theta)^i \psi_i(z, F(z)). \tag{4.7.6}$$

Consider the system of tree classes:

$$\mathcal{T}^{(i)} \cong \mathcal{Z} \times \psi_i(\mathcal{Z}, \mathcal{T}) \; \forall i = 1, \ldots, N$$

$$\mathcal{T} = \bigcup_{i=0}^{N} \mathcal{T}^{(i)}.$$

Let $A(k) = \sum_{j=0}^{M} a_j k^{j+1}$. If $A(k) \neq 0$ for all $k \in \mathbb{N}^+$ then set $B_k^{(i)} = \frac{k^i}{A(k)}$ and $B_t = B_k^{(i)}$ for all $t \in \mathcal{T}^{(i)}$. If $A(k)$ has a zero in the natural numbers we cannot proceed. Also the system of tree classes should be solvable. This requires that $\psi_i(z, 0)$ is non zero for some i.

By Corollary 4.5.4 it follows that for each i, $F_{\mathcal{T}^{(i)}, B}$ satisfies the differential equation.

$$\sum_{j=0}^{M} a_j(1 + \theta)^j F'_{\mathcal{T}^{(i)}, B}(z) = (1 + \theta)^i \psi_i(z, F_{\mathcal{T}, B}(z)).$$

Since $F_{\mathcal{T}, B} = \sum_{i=0}^{N} F_{\mathcal{T}^{(i)}, B}$, we have that $F_{\mathcal{T}, B}$ solves Equation 4.7.6.

Now we shall look at a few examples that highlight how Theorem 4.7.5 can be used to represent differential equations with combinatorics.

Example 4.7.6. First we have linear differential equations. Consider the differential equation:

$$\sum_{i=0}^{N} f_i(z) F^{(i)}(z) = f_{-1}(z), \tag{4.7.7}$$

where f_i are $2i$ times differentiable functions and have a zero at the origin of order at least $i - 1$ for $i \geq 1$.

We shall proceed to rewrite the above differentiable equation in the end form of Theorem 4.7.5. For $i \geq 1$, we know that $f_i(z) = a_i z^{i-1} + z^i g_i(z)$ for some $a_i \in \mathbb{K}$ and g_i, i times differentiable function. If we let $g_0 = f_0$ then we can rewrite Equation 4.7.7 as such:

$$\sum_{i=1}^{N} a_i z^{i-1} F^{(i)}(z) = -\sum_{i=0}^{N} z^i g_i(z) F^{(i)}(z) + f_{-1}(z).$$

From Proposition 4.7.3 we know that there exist $h_0, \ldots, h_N$ such that Equation 4.7.7 is equivalent to:

$$\sum_{i=1}^{N} a_i z^{i-1} F^{(i)}(z) = \sum_{i=0}^{N} z^i \frac{d^i}{dz^i}(h_i(z) F(z)) + f_{-1}(z).$$

Also from Proposition 4.7.2 if we let $b_i = \sum_{j=i}^{N} s(i, j) a_j$ and $\phi_i(z) = \sum_{j=i}^{N} s(i + 1, j + 1) h_j(z)$ we get that Equation 4.7.7 is equivalent to:

$$\sum_{i=1}^{N} b_i(1 + \theta)^{i-1} F'(z) = \sum_{i=0}^{N} (1 + \theta)^i (\phi_i(z) F(z)) + f_{-1}(z), \tag{4.7.8}$$

which is the desired form.

Now we define our tree classes. Let $\mathcal{T}, \mathcal{T}^{(-1)}, \mathcal{T}^{(0)}, \ldots$ be tree classes satisfying:

$$\mathcal{T}^{(-1)} \cong \mathcal{Z} \times \mathrm{SEQ}(\mathcal{Z}),$$

$$\mathcal{T}^{(m)} \cong \mathcal{Z} \times \mathcal{Z}^m \times \mathcal{T} \, \forall m \geq 0,$$

$$\mathcal{T} = \bigcup_{m \geq -1} \mathcal{T}^{(m)}.$$

Let $b(k) = \sum_{i=1}^{N} b_i k^i$ and $\phi_m^*(k) = \sum_{i=0}^{N} [z^m] \phi_i(z) k^i$. Assuming that $b(k)$ is nonzero for all positive integers k, we also define hook length operators:

$$B_k^{(-1)} = \frac{[z^{k-1}] f_{-1}(z)}{b(k)},$$

$$B_k^{(m)} = \frac{\phi_m^*(k)}{b(k)} \, \forall m \geq 0,$$

$$B_t = B_{|t|}^{(m)}, t \in \mathcal{T}^{(m)}.$$

By Corollary 4.5.4 $F = F_{\mathcal{T},B}$ solves Equation 4.7.8 and therefore solves Equation 4.7.7. We can write $F_{\mathcal{T},B}$ explicitly as a sum of compositions as follows:

$$F_{\mathcal{T},B}(z) = \sum_{(a_1, \ldots, a_n) \in \mathcal{P}} \frac{([x^{a_1 - 1}] f_{-1}(x)) \prod_{i=2}^{n} \phi_{a_i - 1}^*(A(i))}{\prod_{i=1}^{n} b(A(i))} z^{A(n)},$$

where $A(i) = \sum_{j=1}^{i} a_j$ and $\mathcal{P}$ is the class of compositions. ∎

We are not restricted to having the f_i have a zero at the origin of order at least $i - 1$ as can we simply multiply both sides of the equation by z until that is the case. The following example illustrates this.

Example 4.7.7. Consider the differential equation:

$$F^{(n)}(z) = 1. \tag{4.7.9}$$

From antidifferential calculus we know that solution to Equation 4.7.9 have the form $F(z) = \frac{z^n}{n!} + \sum_{i=0}^{n-1} C_i z^i$.

Multiplying both sides of the equation by z^{n-1} to get:

$$z^{n-1} F^{(n)}(z) = z^{n-1}.$$

By Proposition 4.7.2 we know that this is equivalent to:

$$\sum_{i=0}^{n-1} s(n, i+1)(1+\theta)^i F'(z) = z^{n-1}. \tag{4.7.10}$$

Let $\mathcal{T} \cong \mathcal{Z}^n$ and $B_n = \frac{1}{\sum_{i=1}^{n} s(n,i) k^i}$. By Corollary 4.5.4 $F_{\mathcal{T},B} = F$ solves Equation 4.7.10 and thus solves Equation 4.7.9. Note that

$$F_{\mathcal{T},B} = B(\bullet^n) z^n = \frac{1}{\sum_{i=1}^{n} s(n,i) n^i} z^n = \frac{1}{n!} z^n,$$

75

which is the solution where $C_i = 0$ for all $i = 0, \ldots, n-1$. ∎

Now we shall investigate a more interesting class of differential equations that lead to classes of trees that are more than just lines.

Example 4.7.8. Consider the differential equation:

$$F'(z) = \sum_{i=0}^{N} z^i \frac{d^i}{dz^i}(P_i(F(z))), \tag{4.7.11}$$

where the P_i are i times differentiable functions.

From Proposition 4.7.2 if we let $Q_i(x) = \sum_{j=i}^{N} s(i+1, j+1)Q_j(x)$ we get that Equation 4.7.11 is equivalent to:

$$F'(z) = \sum_{i=0}^{N} (1 + \theta)^i (Q_i(F(z))), \tag{4.7.12}$$

which is the desired form.

Now we define our tree classes. Let $\mathcal{T}, \mathcal{T}^{(-0)}, \mathcal{T}^{(-1)}, \ldots$ be tree classes satisfying:

$$\mathcal{T}^{(m)} \cong \mathcal{Z} \times \mathcal{T}^m \; \forall m \geq 0,$$
$$\mathcal{T} = \bigcup_{m \geq 0} \mathcal{T}^{(m)}.$$

Let $Q_m^*(k) = \sum_{i=0}^{N} [z^m] Q_i(z) k^i$.

$$B_k^{(m)} = \frac{Q_m^*(k)}{k} \; \forall m \geq 0,$$
$$B_t = B_{|t|}^{(m)}, t \in \mathcal{T}^{(m)}.$$

By Corollary 4.5.4 $F = F_{\mathcal{T}, B}$ solves Equation 4.7.12 and therefore solves Equation 4.7.11. ∎

We have seen how some differential equations can be represented combinatorially. We can broaden the types of differential equations we can combinatorially resolve by applying the differential equation in Corollary 4.2.4 thereby including forests. We can also resolve systems of differential equations like Leroux and Viennot [39, 40].

Chapter V

Conclusion

The main purpose of this thesis is to unify the idea of the tree factorial and its generalizations amongst three communities. These communities are the tree hook length formula community, the B-series community and the quantum field theory community. The ideas of the three communities can be collected together in the following theorem.

Theorem 5.0.9. *Let $\mathcal{T}$ be a class of decorated trees satisfying: $\mathcal{T} \cong \mathcal{Z} \times \Phi(\mathcal{Z}, \mathcal{T}, \mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(l)})$, where each $\mathcal{T}^{(i)}$ is a class of decorated trees and let B be a general hook length operator where $s, t \in \mathcal{T}_n$ implies $B_s = B_t$. Define*

$$F(z) = \sum_{t \in \mathcal{T}} B(t)\zeta^{(|t|)},$$

$$F^{(i)}(z) = \sum_{t \in \mathcal{T}^{(i)}} B(t)\zeta^{(|t|)}$$

and ϕ to be the formal power series of Φ. Then

1. *$[z^0]F(z) = 0$ and*

$$B_k = \frac{[z^k]F(z)}{[z^{k-1}]\phi(z, F(z), F^{(1)}(z), \ldots, F^{(l)}(z))}, \forall k \geq 1$$

2. *F satisfies*

$$F(z) = L_B^* \phi(z, F(z), F^{(1)}(z), \ldots, F^{(l)}(z)),$$

   and

3. *F satisfies the dynamical system*

$$F'(z) = L_B(1 + \theta)\phi(z, F(z), F^{(1)}(z), \ldots, F^{(l)}(z)), \quad F(0) = 0.$$

   Here L_B^* and L_B are as defined in Sections 3.2 and 4.2 respectively.

The above theorem is a combination of Corollary 2.6.1, Corollary 3.2.2 and Corollary 4.5.4.
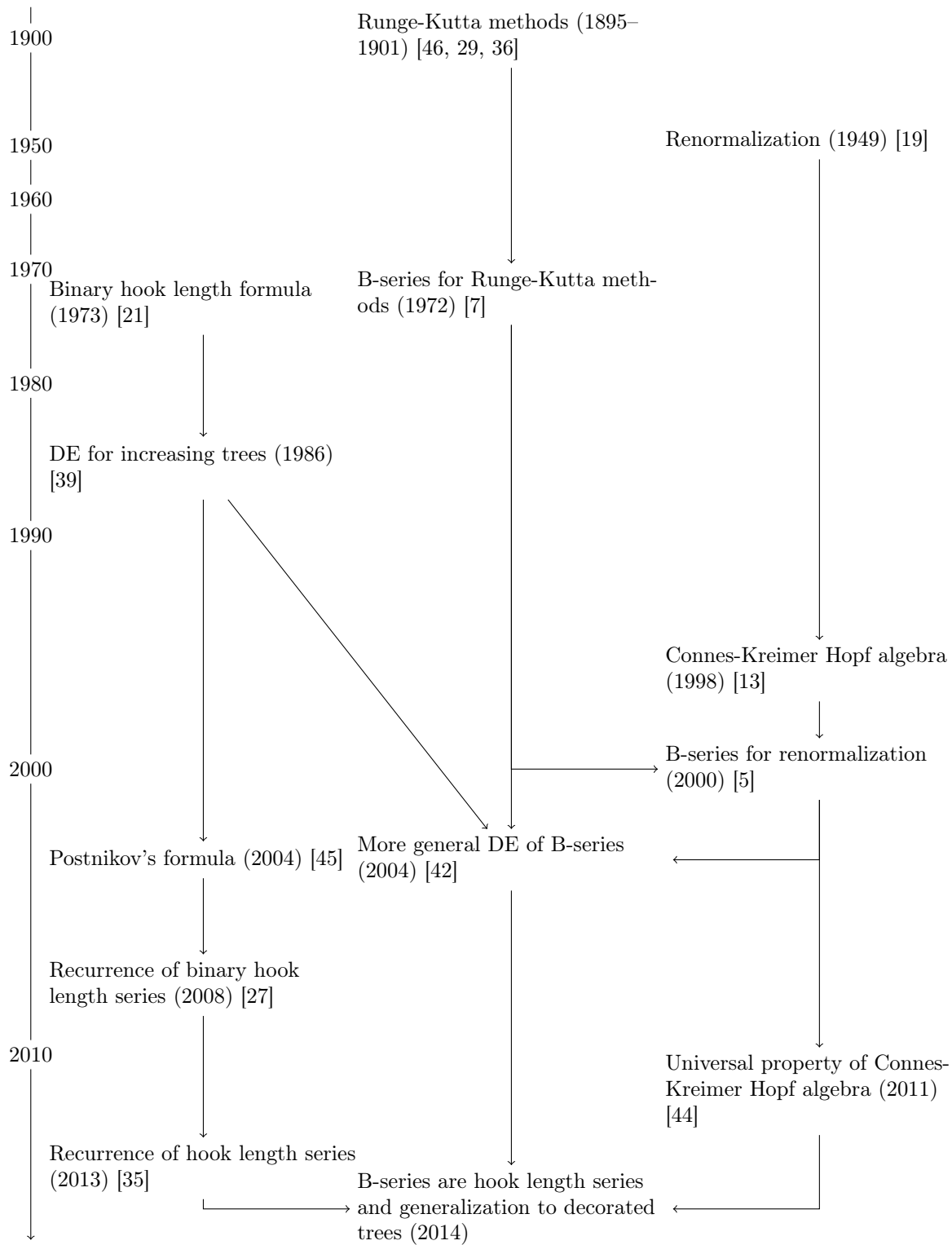
Figure 5.0.1: The history of hook length formulae, B-series and Feynman rules. The edges represent knowledge of previous work.

The hook length formula community was able to find the recurrence for simple trees in 2013 and also recurrences for a few more general classes (see [35]). The B-series community was able to find the differential equation for simple trees in 2004 [42]. The quantum field theory community was able to define hook length-like Feynman rules for decorated trees in 2011 [44] and use B-series for toy Feynman examples in 2000 [5]. Figure 5.0.1 illustrates the history of the three communities as we best understand it in a graphical manner.

From the unification of hook length formulae and B-series we were able to develop new methods using the differential equation to find hook length formulae. We were also able to extend the work of Butcher, Leroux, Viennot and Mazza of finding combinatorial solutions to differential equations.

This thesis investigated two frameworks for finding or proving hook length formulae. One was through a recurrence (Theorem 2.2.4) and the other was through a differential equation (Theorem 4.2.1). The advantage of the recurrence is that it is easier to compute the hook length operator needed to obtain a given hook length series. The advantages of the differential equation are that it was easier to obtain the hook length series from a given hook length operator and that the differential equation can give closed form solutions. We also extended these frameworks to decorated trees and some non-simple tree classes. Finally given that the differential equation framework is simpler to use than the recurrence, we developed methods to still use the differential equation if the hook length operator did not fit into Mazza's specification.

We saw in Section 2.3 that some hook length formulae imply isomorphisms among certain combinatorial classes. For some of the new hook length formulae such isomorphisms have been found: Formulae 6.16.3 and 6.17.1. However, a couple of new hook length formulae imply the existence of isomorphisms that remain to be found. One is Formula 6.11.6 which implies that there is a bijection between bicolored labelled unordered rooted trees with with one color increasing and n nodes and planar embedded trees with $n + 1$ nodes. Another is Formula 6.6.4, which implies that there is a isomorphism between a class of labelled Motzkin trees and the class of permutations. All of the isomorphisms in Section 2.3 use the recurrence construction of the tree class considered. However it is unknown whether there is a universal form for the isomorphisms arising from hook length formula. The new isomorphisms all use unlabelling which may be useful to find other isomorphisms from hook length formulae.

We also saw how Feynman rules in quantum physics can be related to hook length operators. This leads to new toy models in quantum physics that have combinatorial properties. This connection gave us a combinatorial specification for hook length series.

One open problem is to further explain why the hook length series of Formulae 6.11.7 and 6.14.2 are equal. Specifically, it would be nice to have a partition of the set of trees of size n that have an odd-degree node where each part of the partition add up to zero after applying the hook length operator. Another topic that is open for study is the general hook length operators of Section 2.6. While there are examples of more general hook length operators (like Kuba and Panholzer's hook length operators that depend on height [35]), there is no mention of hook length formulae of general hook length operators in the combinatorial community, except in this thesis.

We can also study hook length operators of other combinatorial objects. Hook length operators of partitions are very classical (see [51]) and a similar recurrence to Theorem 2.2.4 was found by Han [27]. Another object that hook length operators have been studied with is lattices. The hook length operator for lattices actually generalizes the hook length of partitions and trees [52, 3]. The existence of a recurrence for lattices akin to the recurrences for trees and partitions is open.

The final chapter of this thesis is a catalogue of known hook length formulae. As mentioned in the introduction, a catalogue of this extent has not been complied anywhere prior to this work. There are nineteen new hook length formulae found by the author. Of these new hook length formulae, Formulae 6.6.4, 6.7.8, 6.8.1, 6.9.4, 6.10.2, 6.16.4, 6.18.11 and 6.20.2 all have a hook length series that is one of $\frac{z}{1-z}$, $\frac{z}{1-z^r}$ or $\frac{1}{1-z}$ and Formulae 6.5.9 and 6.11.6 both use Postnikov's hook length operator, $B_k = 1 + \frac{1}{k}$.

Chapter VI

Catalogue of hook length formulae

This catalogue contains tables of known hook length formulae. Each table contains the known hook length formulae for a specific class of trees or forests and is headed by the combinatorial specification for that class. The tables are also sometimes followed by notes about specific formulae in that table.

Each table has four columns. The first column, **No.**, is a number to identify the formula in this catalogue. The second column, $\mathbf{B_k}$, is the hook length operator for the formula. For the hook length operator k denotes the variable and other letters, if present, are arbitrary constants. The third column, **F**, is the hook length series or the coefficient of z^n of the hook length series. If the entry is $F(z) = f(z)$ then $f(z)$ is the hook length series and if the entry is $F_n = a_n$ or $n!F_n = n!a_n$ then a_n is the coefficient of z^n of the hook length series. F_n is used for unlabelled classes and $n!F_n$ is used for labelled classes. The last column, **source**, is the earliest source where the hook length formula was found. This will either be an external citation or a reference to a theorem or example in this thesis. If the latter then it is a new hook length formula. Some of external sources did not express the hook length series as a function, but instead just gave the series expansion. We express the hook length as a function whenever possible, even if no other source gave the formula with a function.

For the last table, Section 6.21, the third column, $\mathbf{B(f)}$, is B when applied to a single forest, f.

6.1 Lines

$\mathcal{T} \cong Z \times (1 + \mathcal{T})$

No.	$\mathbf{B_k}$	F	Source
6.1.1	$B_k = \frac{\prod_{i=1}^{r}(a_i+k-1)}{k\prod_{i=1}^{s}(b_i+k-1)}$	$F(z) = {}_r\mathbf{F}_s(a_1,\ldots,a_r;b_1,\ldots,b_s;z) - 1$	Theorem 4.3.1
6.1.2	$B_k = \frac{(a+k-1)(b+k-1)}{k(c+k-1)}$	$F(z) = {}_2\mathbf{F}_1(a,b;c;z) - 1$	Theorem 4.3.1

6.2 Binary trees

$\mathcal{T} \cong \mathcal{Z} \times (1+\mathcal{T})^2$

No.	B_k	F	Source
6.2.1	$B_k = \frac{1}{k}$	$F(z) = \frac{z}{1-z}$	[21]
6.2.2	$B_k = 1 + \frac{1}{k}$	$F(z) = \frac{\mathbf{W}(2z)}{2z} - 1$	[45]
6.2.3	$B_k = \frac{(k+1)a+1-k}{2k}$	$F_n = \frac{1}{n+1}\binom{(n+1)a}{n}$	[18]
6.2.4	$B_k = \frac{1}{k2^{k-1}}$	$F(z) = e^z - 1$	[27]
6.2.5	$B_k = \frac{(a+k)^{k-1}}{k(2a+k-1)^{k-2}}$	$F(z) = \left(\frac{\mathbf{W}(2z)}{2z}\right)^a - 1$ *	[27]
6.2.6	$B_k = \frac{6}{k(k+2)}$	$F(z) = \frac{1}{(1-z)^2} - 1$	[27]
6.2.7	$B_k = \begin{cases} a & \text{if } k = 1 \\ \frac{\prod_{i=1}^{k-1}(a+i)}{2k\prod_{i=1}^{k-2}(2a+i)} & \text{if } k \geq 2 \end{cases}$	$F(z) = \frac{1}{(1-z)^a} - 1$	[27]
6.2.8	$B_k = \frac{k+3}{2k}$	$F(z) = \left(\frac{1-\sqrt{1-4z}}{2z}\right)^2 - 1$	[27]
6.2.9	$B_k = \begin{cases} a & \text{if } k = 1 \\ \frac{\prod_{i=1}^{k-1}(a+2k-i)}{2k\prod_{i=1}^{k-2}(2a+2k-2-i)} & \text{if } k \geq 2 \end{cases}$	$F(z) = \left(\frac{1-\sqrt{1-4z}}{2z}\right)^a - 1$	[27]
6.2.10	$B_k = \begin{cases} a(b+1) & \text{if } k = 1 \\ \frac{\prod_{i=1}^{k-1}\alpha_i(k,a,b)}{2k\prod_{i=1}^{k-2}\alpha_i(k-1,2a,b)} & \text{if } k \geq 2 \end{cases}$	$F_n = \frac{a(b+1)}{n!}\prod_{i=1}^{n-1}\alpha_i(n,a,b)$ †	[27]
6.2.11	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{2k} & \text{if } k > 1 \end{cases}$	$F(z) = \tan z + \sec z - 1$	[27]
6.2.12	$B_k = \begin{cases} a & \text{if } k = 1 \\ \frac{1}{2ak} & \text{if } k \text{ is even} \\ \frac{a}{k(1+a^2)} & \text{if } k \geq 3 \text{ is odd} \end{cases}$	$F(z) = a\tan z + \sec z - 1$	[27]
6.2.13	$B_k = \begin{cases} 1 & \text{if } k \text{ is odd} \\ -\frac{1}{k} & \text{if } k \text{ is even} \end{cases}$	$F(z) = \frac{z-z^2}{1+z^2}$	[27]
6.2.14	$B_k = \begin{cases} \frac{1}{m} & \text{if } k = 3m-2 \\ 0 & \text{if } k = 3m-1 \\ -\frac{1}{m} & \text{if } k = 3m \end{cases}$	$F(z) = \frac{z-z^3}{1+z^3}$	[27]
6.2.15	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ 2 & \text{if } k \geq 2 \end{cases}$	$F(z) = \frac{1-4z-\sqrt{1-8z(1-z)}}{4z}$	[27]
6.2.16	$B_k = \frac{1}{(2k+1)2^{2k-1}}$	$F(z) = \left(\sinh(\sqrt{z})/\sqrt{z}\right) - 1$	[28]
6.2.17	$B_k = \frac{((k-1)!)^2}{(2k-1)!}$	$F_n = \frac{2^n}{(n+1)!n!}$	Example 2.4.3

* The series expansion of this function is $F_n = \frac{a(a+n)^{n-1}}{n!}2^n$.

† $\alpha_i(n, a, b) = ab + a + (2n - i)b + i$.

6.3 Complete binary trees

$\mathcal{T} \cong \mathcal{Z} \times (1 + \mathcal{T}^2)$

No.	$\mathbf{B_k}$	F	Source
6.3.1	$B_k = \frac{1}{k}$	$F(z) = \tan z$	[21]
6.3.2	$B_k = \begin{cases} a & \text{if } k = 1 \\ \frac{1}{ak} & \text{if } k \geq 2 \end{cases}$	$F(z) = a\tan z$	[27] *
6.3.3	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{2}{k-1} & \text{if } k \geq 2 \end{cases}$	$F(z) = \frac{z}{1-z^2}$	[27] *
6.3.4	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{k2^{k-2}} & \text{if } k \geq 2 \end{cases}$	$F(z) = \sinh(z)$	[27] *
6.3.5	$B_k = \begin{cases} a & \text{if } k = 1 \\ \frac{1}{a} & k \geq 2 \end{cases}$	$F(z) = a\frac{1-\sqrt{1-4z^2}}{2z}$	[27] *
6.3.6	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{2}{1-k} & k \geq 2 \end{cases}$	$F(z) = \frac{z}{1+z^2}$	[27] *

* The odd-sized semicomplete binary trees of the next section are precisely the complete binary trees. The formulae in this table whose source is [27] were derived by the author of this thesis from the formulae for semicomplete binary trees.

6.4 Semicomplete binary trees

$\mathcal{T} \cong \mathcal{Z} \times (1 + \mathcal{T}_{\mathrm{odd}} \times (1 + \mathcal{T}))$

No.	B_k	F	Source
6.4.1	$B_k = \frac{1}{k}$	$F(z) = \tan z + \sec z - 1$	[55]
6.4.2	$B_k = \begin{cases} a & \text{if } k = 1 \\ \frac{1}{ak} & \text{if } \geq 2 \end{cases}$	$F(z) = a \tan z + \sec z - 1$	[27]
6.4.3	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{k2^{k-2}} & \text{if } \geq 2 \end{cases}$	$F(z) = e^z - 1$	[27]
6.4.4	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{\left\lfloor \frac{k}{2} \right\rfloor} & k \geq 2 \end{cases}$	$F(z) = \frac{z}{1-z}$	[27]
6.4.5	$B_k = \begin{cases} a & \text{if } k = 1 \\ \frac{1}{a} & k \geq 2 \end{cases}$	$F(z) = \frac{1 - \sqrt{1 - 4z^2}}{2z^2}(1 + az) - 1$	[27]
6.4.6	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ -\frac{1}{\left\lfloor \frac{k}{2} \right\rfloor} & k \geq 2 \end{cases}$	$F(z) = \frac{z - z^2}{1 + z^2}$	[27]
6.4.7	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ 0 & \text{if } k = 3m - 1 \\ -\frac{1}{m} & \text{if } k = 6m - 3 \\ \frac{1}{m} & \text{if } k = 6m - 2 \\ -\frac{1}{m} & \text{if } k = 6m \\ \frac{1}{m} & \text{if } k = 6m + 1 \end{cases}$	$F(z) = \frac{z - z^3}{1 + z^3}$	[27]

6.5 Fibonacci trees

$\mathcal{T} \cong \mathcal{Z} \times (1 + \mathcal{T}) \times (1 + \mathcal{Z})$

No.	B_k	F	Source
6.5.1	$B_k = \frac{1}{k}$	$F(z) = e^{z + z^2/2} - 1$	[52]
6.5.2	$B_k = \frac{1}{k^2}$	$F(z) = e^z - 1$	[52]
6.5.3	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{2} & \text{if } k \geq 2 \end{cases}$	$F(z) = \frac{z}{1-z}$	[27]
6.5.4	$B_k = \frac{(k+a-1)(k+a-2)}{k((a+1)k-2)}$	$F(z) = \frac{1}{(1-z)^a} - 1$	[27]
6.5.5	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{4(2k-1)(2k-3)}{(k+1)(5k-6)} & \text{if } k \geq 2 \end{cases}$	$F(z) = \frac{1-\sqrt{1-4z}}{2z} - 1$	[27]
6.5.6	$B_k = \frac{(a+2k-4)(a+2k-3)(a+2k-2)(a+2k-1)}{k(a+k-2)(a+k)((a+4)k-6)}$	$F(z) = \left(\frac{1-\sqrt{1-4z}}{2z}\right)^a - 1$	[27]
6.5.7	$B_k = \begin{cases} a & \text{if } k = 1 \\ \frac{2m-1}{(m+1)a} & \text{if } k = 2m \\ \frac{2a(2m-1)}{a^2(m+1)+2(2m-1)} & \text{if } k = 2m+1 \end{cases}$	$F(z) = \frac{1-\sqrt{1-4z^2}}{2z^2}(1+az) - 1$	[27]
6.5.8	$B_k = \begin{cases} -1 & \text{if } k \equiv 0 \mod 3 \\ 1 & \text{if } k \equiv 1 \mod 3 \\ 0 & \text{if } k \equiv 2 \mod 3 \end{cases}$	$F(z) = \frac{z - z^3}{1 + z^3}$	[27]
6.5.9	$B_k = 1 + \frac{1}{k}$	$F(z) = \frac{1-2z}{((1-2z)^2(z+1))^{4/3}} - 1$	Example 4.5.7

6.6 Motzkin trees

$\mathcal{T} \cong \mathcal{Z} \times (1 + \mathcal{T} + \mathcal{T}^2)$

No.	B_k	F	Source
6.6.1	$B_k = \frac{1}{k}$	$F(z) = \frac{\sqrt{3}}{2}\tan\left(\frac{\sqrt{3}}{2}z + \frac{\pi}{6}\right) - \frac{1}{2}$	[1]
6.6.2	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ 2\left(1 + \frac{1}{k}\right) & \text{if } k > 1 \end{cases}$	$F(z) = \frac{1}{2}\left(\frac{\mathbf{W}(2z)}{2z} - 1\right)$	[35]
6.6.3	$B_k = \begin{cases} \frac{1}{2} & \text{if } k = 1 \\ \frac{1}{k2^{k-2}} & \text{if } k > 1 \end{cases}$	$F(z) = \frac{1}{2}\left(e^z - 1\right)$	[35]
6.6.4	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{k-1} & \text{if } k > 1 \end{cases}$	$F(z) = \frac{z}{1-z}$	Example 4.6.1

6.7 r-ary trees

$\mathcal{T} \cong \mathcal{Z} \times (1 + \mathcal{T})^r$

No.	B_k	F	Source
6.7.1	$B_k = \frac{1}{k}$	$F(z) = -1 + (1 - (r-1)z)^{1/1-r}$	[52]
6.7.2	$B_k = \frac{((r-1)k+1)a+1-k}{rk}$	$F_n = \frac{1}{(r-1)n+1}\binom{((r-1)n+1)a}{n}$	[18]
6.7.3	$B_k = a + \frac{1}{k}$	$F_n = \frac{a+1}{n!} \prod_{i=1}^{n-1} \beta_i(n, r, a)$ *	[18]
6.7.4	$B_k = r - 1 + \frac{1}{k}$	$F_n = \frac{r^n((r-1)n+1)^{n-1}}{n!}$	[18]
6.7.5	$B_k = 1 + \frac{a}{k}$	$F_n = \frac{a+1}{n!} \prod_{i=1}^{n-1} \gamma_i(n, r, a)$ †	[48]
6.7.6	$B_k = \frac{1}{kr^{k-1}}$	$F(z) = e^z - 1$	[56]
6.7.7	$B_k = \begin{cases} ab & \text{if } k = 1 \\ \prod_{i=1}^{k-1} \delta_i(n, r, a, b) & \text{if } k \geq 2 \end{cases}$ ‡	$F_n = \frac{ab}{n!} \prod_{i=1}^{n-1} \epsilon_i(n, r, a, b)$ §	[10]
6.7.8	$B_k = 1/\binom{k+r-2}{r-1}$	$F(z) = \frac{z}{1-z}$	Example 2.4.1

* $\beta_i(n, r, a) = ((r-1)n + i + 1)a + (r-1)(n-i) + 1$.

† $\gamma_i(n, r, a) = (ri - i + 1)(a + 1) + r(n - i)$.

‡ $\delta_i(k, r, a, b) = \begin{cases} \frac{ab+(b-1)rk-i(b-r)}{abr+r(b-1)(k-1)-i(b-r)} & \text{if } i < k - 1 \\ \frac{ab+(b-1)rk-(k-1)(b-r)}{rk} & \text{if } i = k - 1 \end{cases}$.

§ $\epsilon_i(n, r, a, b) = ab + (b-1)rn - i(b-r)$.

6.8 Complete r-ary trees

$\mathcal{T} \cong \mathcal{Z} \times (1 + \mathcal{T}^r)$

No.	B_k	F	Source
6.8.1	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{(r-1)!r^{r-1}}{\prod_{i=0}^{r-2}(k+ir-1)} & \text{if } k > 1 \end{cases}$	$F(z) = \frac{z}{1-z^r}$	Example 2.4.2

6.9 Plane trees

$\mathcal{T} \cong \mathcal{Z} \times \mathrm{SEQ}(\mathcal{T})$

No.	B_k	F	Source
6.9.1	$B_k = \frac{1}{k}$	$F(z) = 1 - \sqrt{1 - 2z}$	[39]
6.9.2	$B_k = \frac{(-1)^k}{k}$	$F(z) = 1 - e^z$	[56]
6.9.3	$B_k = \left(1 - \frac{1}{k}\right)^{k-1}$	$F(z) = 1 - e^{-W(z)}$ *	[10]
6.9.4	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ 2^{2-k} & \text{if } k > 1 \end{cases}$	$F(z) = \frac{z}{1-z}$	Example 4.6.5
6.9.5	$B_k = \begin{cases} 1 & \text{if } k \leq 2 \\ \frac{1}{2} & \text{if } k \equiv 1, 2 \mod 4, k > 2 \\ -\frac{1}{2} & \text{if } k \equiv 0, 3 \mod 4 \end{cases}$	$F(z) = \frac{z + z^2}{1 + z^2}$	Example 4.6.6

* The series expansion of this function is $F_n = \frac{(n-1)^{(n-1)}}{n!}$.

6.10 Fat plane trees

$\mathcal{T} \cong \mathcal{Z} \times \mathrm{SEQ}(\mathcal{T})^2$

No.	B_k	F	Source
6.10.1	$B_k = \frac{1}{k}$	$F(z) = 1 - (1 - 3z)^{1/3}$	Example 4.4.1
6.10.2	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{2^{k-3}(k+2)} & \text{if } k > 1 \end{cases}$	$F(z) = \frac{z}{1-z}$	Example 4.6.7

6.11 Labelled unordered trees

$\mathcal{T} \cong \mathcal{Z} \times \mathrm{SET}(\mathcal{T})$

No.	B_k	F	Source
6.11.1	$B_k = \frac{1}{k}$	$F(z) = \log(\frac{1}{1-z})$	[43]
6.11.2	$B_k = \frac{1}{k^2}$	$F(z) = 2\log\left(\frac{2}{2-z}\right)$	[42]
6.11.3	$B_k = \frac{1}{k\mathbb{B}_{k-1}}$ *	$F(z) = e^z - 1$	[10]
6.11.4	$B_k = \frac{1}{k(2k-1)}$ †	$F(z) = 2\log\left(\sec\left(\sqrt{z}\right)\right)$	[34]
6.11.5	$B_k = \frac{1}{k\binom{k+a-2}{a-1}}$	$F(z) = a\log\left(\frac{a}{a-z}\right)$	Example 2.4.4
6.11.6	$B_k = 1 + \frac{1}{k}$	$F(z) = 2\log\left(\frac{1}{1-\frac{1-\sqrt{1-4z}}{2}}\right)$	Example 4.4.2
6.11.7	$B_k = \frac{2}{k} - 1$	$F(z) = \log\left(z + \sqrt{z^2 + 1}\right)$	Example 4.4.3

* $\mathbb{B}_n$ is the nth Bell number. See [20, §II.3]

† Hook length operator was $\frac{1}{2k(2k-1)}$ in [34].

6.12 Labelled unordered binary trees

$\mathcal{T} \cong \mathcal{Z} \times \mathrm{SET}_{\leq 2}(\mathcal{T})$

No.	B_k	F	Source
6.12.1	$B_k = \frac{1}{k}$	$F(z) = \tan(z) + \sec(z) - 1$	[1]

6.13 Labelled unordered complete binary trees

$\mathcal{T} \cong \mathcal{Z} \times (1 + \mathrm{SET}_2(\mathcal{T}))$

No.	B_k	F	Source
6.13.1	$B_k = \frac{1}{k}$	$F(z) = \sqrt{2}\tan\left(\frac{z}{\sqrt{2}}\right)$	[1]

6.14 Labelled unordered even trees

$\mathcal{T} \cong \mathcal{Z} \times \mathrm{SET}_{\mathrm{even}}(\mathcal{T})$

No.	$\mathbf{B_k}$	$\mathbf{F}$	Source
6.14.1	$B_k = \frac{1}{k}$	$F(z) = \log\left(\tan(z) + \sec(z)\right)$	[1]
6.14.2	$B_k = \frac{2}{k} - 1$	$F(z) = \log\left(z + \sqrt{z^2 + 1}\right)$ *	Example 4.4.4

* This formula has the same hook length operator and hook length series as Formula 6.11.7.

6.15 Labelled unordered odd trees

$\mathcal{T} \cong \mathcal{Z} \times (1 + \mathrm{SET}_{\mathrm{odd}}(\mathcal{T}))$

No.	$\mathbf{B_k}$	$\mathbf{F}$	Source
6.15.1	$B_k = \frac{1}{k}$	$F(z) = 2\operatorname{arctanh}\left(1 - \sqrt{2}\tanh\left(\operatorname{arccoth}\left(\sqrt{2}\right) - \frac{z}{\sqrt{2}}\right)\right)$	[1]

6.16 Cyclic trees

$\mathcal{T} \cong \mathcal{Z} \times (1 + \mathrm{CYC}(\mathcal{T}))$

No.	$\mathbf{B_k}$	$\mathbf{F}$	Source
6.16.1	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ 1 - \frac{1}{k} & \text{if } k > 1 \end{cases}$	$F(z) = 1 - e^{-\mathbf{W}(z)}$	[35]
6.16.2	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{1}{k} - 1 & \text{if } k > 1 \end{cases}$	$F(z) = 1 + z - \sqrt{z^2 + 1}$	Example 4.6.2
6.16.3	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ k - 1 & \text{if } k > 1 \end{cases}$	$F(z) = 1 - 1/\sum_{n \geq 0} n! z^n$	Example 4.6.3
6.16.4	$B_k = \begin{cases} 1 & \text{if } k = 1 \\ \frac{k-1}{2^{k-1} - 1} & \text{if } k > 1 \end{cases}$	$F(z) = \frac{z}{1-z}$	Example 4.6.8

6.17 Schröder trees

$\mathcal{T} \cong \mathrm{SEQ}(\mathrm{SET}_{\geq 1}(\mathcal{T}))$

No.	$\mathbf{B_k}$	$\mathbf{F}$	Source
6.17.1 *	$B_k = \frac{1}{k}$	$F(z) = \mathbf{W}\left(\frac{1}{2}\exp\left(\frac{z-1}{2}\right)\right) + \frac{z-1}{2}$	Example 4.4.5

6.18 Plane forests

$\mathcal{T} \cong \mathrm{SEQ}(\mathcal{Z} \times \mathcal{T})$

No.	B_k	F	Source
6.18.1	$B_k = 1 + \frac{a}{k}$	$F_n = \frac{a+1}{n!} \prod_{i=1}^{n-1} \left((2i+1)(a+1) + (n-i)\right)$	[48]
6.18.2	$B_k = a + \frac{1}{k}$	$F_n = \frac{a+1}{n!} \prod_{i=1}^{n-1} \left((2n+1)(a+1) - (a+2)i\right)$	[18]
6.18.3	$B_k = \frac{1}{k}$	$F(z) = \frac{1}{\sqrt{1-2z}}$	[18]
6.18.4	$B_k = \frac{(-1)^k}{k}$	$F(z) = e^{-z}$	[10]
6.18.5	$B_k = \mathfrak{B}_k{}^{*}$	$F(z) = \frac{1-e^{-z}}{z}$	[10]
6.18.6	$B_k = \prod_{i=1}^{k-1} \zeta_i(k,a,b)^{\dagger}$	$F_n = \frac{ab}{n!} \prod_{i=1}^{n-1} \left((2n+a)b - (b+1)i\right)$	[10]
6.18.7	$B_k = \frac{\prod_{i=1}^{k-1}(2k-a-2i)}{k \prod_{i=2}^{k-1}(2k+a-2i)}$	$F_n = \frac{a}{n!} \prod_{i=1}^{n-1}(2n+a-2i)$	[10]
6.18.8	$B_k = \frac{(2k-a)^{k-1}}{k(2k-2+a)^{k-2}}$	$F_n = \frac{a}{n!}(2n+a)^{n-1}$	[10]
6.18.9	$B_k = \left(1 - \frac{1}{k}\right)^{k-1}$	$F(z) = \frac{\mathbf{W}(z)}{z}$	[10]
6.18.10	$B_k = \frac{(2k-a-1)!(k+a-1)!}{k(k-a)!(2k+a-3)!}$	$F_n = \frac{a}{n+a}\binom{2n+a}{n}$	[10]
6.18.11	$B_k = \begin{cases} 1 & \text{if } k=1 \\ 0 & \text{if } k>1 \end{cases}$	$F(z) = \frac{1}{1-z}$	Example 2.4.5

* $\mathfrak{B}_n$ is the nth Bernoulli number. See [20, §IV.6].

$\dagger \ \zeta_i(k,a,b) = \begin{cases} \frac{(2k-a)b-(b+1)i}{(2k-2+a)b-(b+1)i} & \text{if } i < k-1 \\ \frac{(2k-a)b-(b+1)(k-1)}{k} & \text{if } i = k-1. \end{cases}$

6.19 Labelled unordered forests

$\mathcal{T} \cong \mathrm{SET}(\mathcal{Z} \times \mathcal{T})$

No.	B_k	F	Source
6.19.1	$B_k = 1 + \frac{a}{k}$	$n!F_n = (a+1)\prod_{i=1}^{n-1}\left((i+1)(a+1)+(n-i)\right)$	[48]
6.19.2	$B_k = 1 + \frac{1}{k}$	$F(z) = \frac{1-2z-\sqrt{1-4z}}{2z^2}$	[48]
6.19.3	$B_k = \frac{2(2k-2)!!}{k(2k-3)!!}$	$F(z) = \frac{1}{\sqrt{1-4z}}$	[10]
6.19.4	$B_k = \frac{\prod_{i=1}^{k-1}(bk-(b-1)i)}{k\prod_{i=1}^{n-2}(k-1+a)b-(b-1)i}$	$n!F_n = ab\prod_{i=1}^{n-1}((n+a)b-(b-1)i)$	[10]
6.19.5	$B_k = \left(\frac{k}{k+a-1}\right)^{k-2}$	$F(z) = \left(\mathbf{W}(a)/z\right)^a$	[10]
6.19.6	$B_k = \frac{1}{k\binom{k+a-2}{a-1}}$	$F(z) = \frac{1}{(1-z/a)^a}$	[10]
6.19.7	$B_k = \frac{1}{k}$	$F(z) = \frac{1}{1-z}$	[10]
6.19.8	$B_k = \frac{1}{k^2}$	$F(z) = \frac{4}{(2-z)^2}$	[10]
6.19.9	$B_k = \frac{1}{k(2k-1)}$ *	$F(z) = 1 + \left(\tan\left(\sqrt{z}\right)\right)^2$	[34]

* Hook length operator was $\frac{1}{2k(2k-1)}$ in [34].

6.20 Cyclic forests

$\mathcal{T} \cong 1 + \mathrm{CYC}(\mathcal{Z} \times \mathcal{T})$

No.	B_k	F	Source
6.20.1	$B_k = \begin{cases} 1 & \text{if } n = 1 \\ 1 - \frac{1}{k} & \text{if } n > 1 \end{cases}$	$F(z) = \mathbf{W}(z) + 1$	[35]
6.20.2	$B_k = \sum_{i=1}^{k}\frac{(-1)^{i-1}}{i!}\binom{k-1}{i-1}$	$F(z) = \frac{1}{1-z}$	Example 2.4.6

6.21 Single forest

$f \in \mathcal{F}$

No.	$\mathbf{B_k}$	$\mathbf{B(f)}$	Source
6.21.1	$B_k = 1$	$B(f) = 1$	Folklore
6.21.2	$B_k = a$	$B(f) = a^{\|f\|}$	Folklore
6.21.3	$B_k = C_k D_k$	$B(f) = C(f)D(f)$	Folklore
6.21.4	$B_k = \begin{cases} a & \text{if } k = 1 \\ 1 & \text{if } k > 1 \end{cases}$	$B(f) = a^{\|\mathscr{B}(f)\|}$	Folklore
6.21.5	$B_k = \frac{1}{k}$	$B(f) = \frac{\text{inc}(f)}{\|f\|!}$ *	[31]
6.21.6	$B_k = \frac{[k]_a}{k}$ †	$B(f) = \frac{1}{\|f\|!} \sum_{f' \in l_p(f)} a^{\text{inv}(f')}$ ‡	[3]
6.21.7	$B_k = \frac{[k]_a}{k}$ †	$B(f) = \frac{1}{\|f\|!} \sum_{f' \in l_p(f)} a^{\text{maj}(f')}$ ‡,§	[3]
6.21.8	$B_k = \frac{1}{k}$	$B(f) = \frac{1}{\|f\|} \sum_{v \in \mathscr{B}(f)} \sigma(t \setminus v)$	[33]
6.21.9	$B_k = 1 + \frac{a}{k}$	$B(f) = \frac{1}{\|f\|!} \sum_{f' \in l_p(f)} (1+a)^{\text{inv}(f')}$ ‡	[23]
6.21.10	$B_k = \frac{[2k]_a}{k}$ †	$B(f) = \frac{1}{\|f\|!} \sum_{f' \in l_-(f)} a^{\text{inv}(f')+n_1(f')+n_2(f')}$ §	[11]
6.21.11	$B_k = \frac{(1+q^{k-1})[k]_a}{k}$ †	$B(f) = \frac{2}{\|f\|!} \sum_{f' \in l_e(f)} a^{\text{inv}_e(f')+n_2(f')}$ ¶,‖	[11]
6.21.12	$B_k = \frac{(1+ba^k)[k]_a}{k}$ †	$B(f) = \frac{1}{\|f\|!} \sum_{f' \in l_-(f)} b^{n_1(f')} a^{\text{inv}(f')+n_1(f')+n_2(f')}$ ¶	[11]

* $\text{inc}(f)$ is the number of increasing labellings of a plane embedding of f.

† $[n]_q = (1 + q + \cdots q^{n-1})$.

‡ $l_p(f)$ is the set of labellings of a plane embedding of f and
$\text{inv}(f) = \| \{(u,v) : u, v \in V(f), u \text{ is a descendant of } v, u \text{ is labelled with a greater label than } v\} \|$.

§ $\text{D}(f) = \{v \in V(f) : v \text{ is labelled with a greater label than its parent}\}$ and $\text{maj}(f) = \sum_{v \in \text{D}(f)} |f_v|$.

¶ A signed labelling of a combinatorial object, c, is a labelling from a set $S \subset \{-|c|, \ldots, |c|\}$ such that
$\|S\| = |c|$ and $\{|s| : s \in S\} = \{1, \ldots, |c|\}$, $l_-(f)$ is the set of signed labellings of a plane embedding
of f, $n_1 = \| \{v \in V(f) : \text{the label of } v \text{ is negative}\} \|$ and
$n_2 = \| \{(u,v) : u, v \in V(f), u \text{ is a descendant of } v, \text{ the sum of the labels of } u \text{ and } v \text{ is negative}\} \|$.

‖ $l_e(f)$ is the set of signed labellings of a plane embedding of f where each labelling has an even number
of negative labels.

Bibliography

[1] F. Bergeron, P. Flajolet and B. Salvo. Varieties of increasing trees. *Lecture Notes in Computer Science* **581**, 24–48, 1992.

[2] F. Bergeron, G. Labelle and P. Leroux. *Combinatorial Species and Tree-like Structures.* Cambridge University Press, 1998.

[3] A. Bjorner and M. L. Wachs. q-hook length formulas for forests. *Journal of Combinatorial Theory* **52(2)**, 165–187, 1989.

[4] M. Bousquet-Mélou and M. Mishna. Walks with small steps in the quarter plane. M. E. Lladser, R. S. Maier, M. Mishna and A. Rechnitzer (Editors), *Algorithmic Probability and Combinatorics, Contemporary Mathematics*, volume 520, pages 1–39. American Mathematics Society, 2010.

[5] C. Brouder. Runge-Kutta methods and renormalization. *The European Physics Journal C* **12**, 521–534, 2000.

[6] J. Butcher. Coefficients for the study of Runge-Kutta integration processes. *J. Austral. Math. Soc.* **3**, 185–201, 1963.

[7] J. Butcher. An algebraic theory of integration methods. *Mathematics of Computation* **26**, 79–106, 1972.

[8] J. Butcher. *The Numerical Analysis of Ordinary Differential Equations.* Wiley, 1987.

[9] W. Y. C. Chen. A general bijective algorithm for increasing trees. *Systems Science and Mathematical Sciences* **12(3)**, 193–203, 1999.

[10] W. Y. C. Chen, O. X. Q. Gao and P. L. Guo. Hook length formulas for trees by Han's expansion. *Electronic Journal of Combinatorics* **16**, 2009. R62.

[11] W. Y. C. Chen, O. X. Q. Gao and P. L. Guo. q-hook length formulas for signed labeled forests. *Advances in Applied Mathematics* **51(5)**, 563–582, 2013.

[12] W. Y. C. Chen and L. Yang. On Postnikov's hook length formula for binary trees. *European Journal of Combinatorics* **29**, 1563–1565, 2008.

[13] A. Connes and D. Kreimer. Hopf algebras, renormalization and noncommutative geometry. *Comm. Math. Phys.* **199**, 203–242, 1998.

[14] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics* **5**, 329–359, 1996.

[15] P. J. Davis. 6. Gamma function and related functions. M. Abtamowitz and I. A. Stegun (Editors), *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover Publications, 1965.

[16] R. Donaghey. Alternating permutations and binary increasing trees. *Journal of Combinatorial Theory, Series A* **18**, 141–148, 1975.

[17] J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *J. Compt. Appl. Math.* **6**, 19–26, 1980.

[18] R. Du and F. Liu. (k, m)-Catalan numbers and hook length polynomials for plane trees. *European Journal of Combinatorics* **28(4)**, 1312–1321, 2007.

[19] F. J. Dyson. The S matrix in quantum electrodynamics. *Phys. Rev.* **75**, 1736–1755, 1949.

[20] P. Flajolet and R. Sedgewick. *Analytic Combinatorics.* Cambridge University Press, 2009.

[21] D. Foata and M. P. Schützenberger. Nombres d'Euler et permutations alternantes. J. N. Srivastava, F. Harar, C. R. Rao, G. C. Rota and S. S. Shrikhande (Editors), *A Survey of Combinatorial Theory*, pages 173–187. 1973.

[22] J. Frame, G. Robinson and R. Thrall. The hook graphs of the symmetric group. *Canad. J. Math.* **6**, 316–325, 1954.

[23] I. Gessel and S. Seo. A refinement of Cayley's formula for trees. *Electronic Journal of Combinatorics* **11(2)**, R27, 2006.

[24] R. L. Graham, D. E. Knuth and O. Patashnik. *Concrete Matematics: A Foundation for Computer Science.* Addison-Wesley, 1989.

[25] E. Hairer, S. P. Nørsett and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems.* Springer-Verlag, 1993.

[26] E. Hairer and G. Wanner. On the Butcher group and general multi-value methods. *Computing* **13(1)**, 1–15, 1974.

[27] G. Han. Discovering hook length formulas by an expansion technique. *Electronic Journal of Combinatorics* **15(1)**, R133, 2008.

[28] G. Han. New hook length formulas for binary trees. *Combinatorica* **30(2)**, 253–256, 2010.

[29] K. Heun. Neue Methoden zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen. *Z. Math. Phys.* **45**, 23–48, 1900.

[30] C. Itzykson and J. Zuber. *Quantum Field Theory*. Dover, 2005.

[31] D. Knuth. *The Art of Computer Programming*, volume 3. Addison-Wesley, 1973.

[32] D. Knuth. *The Art of Computer Programming*, volume 3. Addison-Wesley, 1997.

[33] D. Kreimer. Chen's iterated integral represents the operator product expansion. *Advances in Theoretical and Mathematical Physics* **3(3)**, 1999.

[34] M. Kuba and A. Panholzer. Bilabelled increasing trees and hook-length formulas. *European Journal of Combinatorics* **33(2)**, 248–258, 2012.

[35] M. Kuba and A. Panholzer. A unifying approach for proving hook-length formulas for weighted tree families. *Graphs and Combinatorics* **29(6)**, 1839–1865, 2013.

[36] W. Kutta. Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. *Z. Math. Phys* **46**, 435–453, 1901.

[37] M. La Croix. A combinatorial proof of a result of Gessel and Greene. *Discrete Mathematics* **18**, 2251–2256, 2006.

[38] P. Leroux and B. Miloudi. Généralisations de la formule d'Otter. *Annales des Sciences Mathématiques du Québec* **16(1)**, 53–80, 1992.

[39] P. Leroux and G. X. Viennot. Combinatorial resolution of systems of differential equations I: Ordinary differential equations. *Lecture Notes in Mathematics* **1234**, 210–245, 1986.

[40] P. Leroux and G. X. Viennot. Combinatorial resolution of systems of differential equations IV: Separation of variables. *Discrete Mathematics* **72**, 237–250, 1988.

[41] Maplesoft. Maple 18. http://www.maplesoft.com/products/maple/.

[42] C. Mazza. Simply generated trees, B-series and Wigner processess. *Random Structures and Algorithms* **25(3)**, 293–310, 2004.

[43] A. Meir and J. W. Moon. On the altitude of nodes in random trees. *Canadian Journal of Mathematics* **30**, 997–1015, 1978.

[44] E. Panzer. *Hopf-algebraic renormalization of Kreimer's toy model*. Master's thesis, Humboldt-Universität Zu Berlin, 2011.

[45] A. Postnikov. Permutohedra, associahedra, and beyond. *International Mathematics Research Notices* **6**, 1026–1106, 2009.

[46] C. Runge. Über die numerishe Auflösung von Differentialgleichungen. *Math. Ann.* **46**, 167–178, 1895.

[47] B. Sagan. Probabilistic proofs of hook length formulas for complete m-ary trees. *Séminaire Lotharingien Mathematics* **309(8)**, 2584–2588, 2001.

[48] S. Seo. A combinatotial proof of Postnikov's identity and a generalized enumeration of labeled trees. *Electronic Journal of Combinatorics* **11**, N3, 2005.

[49] L. J. Slater. *Generalized Hypergeometic Functions.* Cambridge University Press, 1966.

[50] N. Sloane. The On-Line Encyclopedia of Integer Sequences. http://oeis.org.

[51] R. P. Stanley. Ordered structures and partitions. *Memoirs of the Amer. Math. Society* 1972.

[52] R. P. Stanley. The Fibonacci lattice. *Fibonacci Quart.* **13**, 215–232, 1975.

[53] R. P. Stanley. *Enumerative Combinatorics*, volume 2. Cambridge, 1999.

[54] Y. Sun and H. Zhang. Two kinds of hook length formulas for complete m-ary trees. *Discrete Mathematics* **309(28)**, 2584–2588, 2009.

[55] G. Viennot. Interprétations combinatoires des nombres d'Euler et de Genocchi. *Séminaire de Théorie des Nombres.* 1981.

[56] L. L. M. Yang. Generalizations of Han's hook length identities, 2008. arXiv:math.CO/0805.0109.

[57] K. Yeats. *Growth estimates for Dyson-Schwinger equations.* Ph.D. thesis, Boston University, 2008.

Definitions