Chapter 2
Trees and forests

2.1 Introduction to trees, graphs and forests

Trees, series and numerical methods

It was pointed out by A. Cayley in 1857 [28] that trees, in the sense of this chapter, have an intimate link with the action of differential operators on operands which, in their turn, might also have something of the nature of operators. This is now extended to numerical methods through the order conditions and the use of B-series. As we saw in Chapter 1, the form of B-series, which provides the link between differential equations and numerical approximations, is given by (1.1 c) (p. 2).

The present chapter aims to present a detailed background to the graph-theoretical and combinatorial aspects of this subject, for use in Chapter 3 and later chapters. In this introductory section, trees and forests will be introduced together with an appreciation of Cayley's fundamental work.

Graphs and trees

A graph in mathematics is a set of points (vertices) and a set of connections (edges) between some of the pairs of vertices. It is convenient to name or label each of the vertices and use their names in specifying the edges. If V is the set of vertices and E the set of edges, then the graph is referred to as (V, E).

Basic terminology and observations

- A path is a sequence of vertices in which each successive pair is an edge.
- A graph is connected if there is a path from any vertex to any other vertex.
- A loop is a path from a vertex to itself, in which the other vertices comprising the path are distinct.

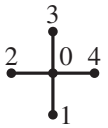
J. C. Butcher, *B-Series*, Springer Series in Computational Mathematics 55,
https://doi.org/10.1007/978-3-030-70956-3_2

- The order of a graph is the number of vertices.
- A tree is a connected graph with at least one vertex and with no loops.
- For a tree, the number of edges is one less than the number of vertices.
- The "empty tree" $\varnothing$, with $V = E = \emptyset$, is somtimes included, as an additional tree.
- The set of trees with a positive number of vertices will be denoted by T and the set of trees, with $\varnothing$ included by $\mathrm{T}^{\#}$. That is, $\mathrm{T}^{\#} = \mathrm{T} \cup \{\varnothing\}$.
- The set of trees t with a positive number of vertices, not exceeding p, will be denoted by T_p.

Examples of graphs

In these examples, the standard notation, V for the set of vertices and E for the set of edges, is used.

$$V = \{a,b,c\} \qquad E = \big\{\{a,b\},\{b,c\},\{c,a\}\big\} \tag{2.1 a}$$



$$V = \{0,1,2,3,4\} \qquad E = \big\{\{0,1\},\{0,2\},\{0,3\},\{0,4\}\big\} \tag{2.1 b}$$

$$V = \{\alpha,\beta,\gamma\} \qquad E = \big\{\{\beta,\gamma\}\big\} \tag{2.1 c}$$

The graph (2.1 a) is connected but contains a loop; hence it is not a tree. In contrast, (2.1 b) is connected and has no loops and is therefore a tree. Note that there are 5 vertices and 4 edges. The final example (2.1 c) has no loops but it is not connected and is therefore not a tree.

Exercise 19 For a graph with a positive number of vertices, show that any two of the following three statements imply the third: (i) the graph is connected, (ii) there are no loops, (iii) the number of vertices is one more than the number of edges.

Rooted and unrooted trees

In using trees in mathematics, it is sometimes natural to treat all vertices in an even-handed way. To specify the structure of such a tree, the sets V and E are all that is needed. However, it is often more useful to specify a particular vertex as constituting the "root" r of the tree. That is, a triple of the form (V,E,r) is needed to fully specify a "rooted tree". Throughout this volume, the single word "tree" will mean such a rooted tree. In contrast, a tree where no root is specified, is called a "free tree" or an "unrooted tree". Throughout this volume, "free tree" and "unrooted tree" will be used interchangeably. The set of free trees will be denoted by U.

There is a good reason for studying trees, both rooted and unrooted. Trees play a central role in the formulation of order conditions for Runge–Kutta and other numerical methods. In particular, elementary differentials, which are the building blocks of B-series, are indexed on the set of rooted trees. Furthermore, unrooted trees emerge as fundamental concepts in the theory of symplectic methods and their generalizations.

Trees and forests

A forest, as a collection of trees, possibly containing duplications, is a natural extension of the idea of trees. We will always include the empty forest in the set of forests and, for a non-empty forest, no account will be taken of the order in which the constituent trees are listed. Algebraically, the set of forests is a monoid (semi-group with identity) and is related to the set of trees via the B^+ operation. The identity, that is the empty set of trees, is denoted by 1.

Terminology

- A typical tree, such as t, t', t_1, t_2, ..., is a member of T.
- A typical unrooted tree, such as u, is a member of the set U.
- A typical forest, such as f, is a member of the set of all forests F.
- A weighted sum of trees, such as τ, is a member of the tree space **T**.
- A weighted sum of forests, such as ϕ, is a member of the forest space **F**.

Some examples

Members of F

(a) 1

(b) 

(c)

Members of **T**

(a) 

(b) 

(c) 

Members of **F**

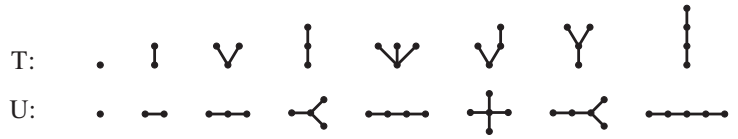
(a) 

(b) 

(c) 

Diagrammatic representations

In this book we will invariably use diagrams with the root at the lowest point and other vertices rising upwards. To represent free trees, we will use diagrams with no special point empahasized in any way.
Examples:



Cayley, trees and differential equations

The pioneering work of Cayley [28] (1857) directly connected trees with differential operators. In this discussion we adapt these ideas to the ordinary differential equation context. Consider the system

$$\frac{\mathrm{d}y^i}{\mathrm{d}x} = f^i = f^i(y^1, y^2, \ldots, y^N), \qquad i = 1, 2, \ldots, N. \tag{2.1 d}$$

Using elementary calculus and the summation convention,

$$\frac{\mathrm{d}^2 y^i}{\mathrm{d}x^2} = f^i_j f^j, \tag{2.1 e}$$

where subscript on f^i_j denotes partial differentiation $(\partial y_j)^{-1}$.

Take this further

$$\frac{\mathrm{d}^3 y^i}{\mathrm{d}x^3} = f^i_{jk} f^j f^k + f^i_j f^j_k f^k, \tag{2.1 f}$$

$$\frac{\mathrm{d}^4 y^i}{\mathrm{d}x^4} = f^i_{jk\ell} f^j f^k f^\ell + 3 f^i_{jk} f^j f^k_\ell f^\ell + f^i_j f^j_{k\ell} f^k f^\ell + f^i_j f^j_k f^k_\ell f^\ell. \tag{2.1 g}$$

If we translate expressions like f^i, f^i_j, f^i_{jk}, … into analogous statements about parent-child relationships involving people whose names are i, j, k, …, we see that the terms occurring in (2.1 d), (2.1 e), (2.1 f), (2.1 g) translate to kinship statements, such as

f^i translates to

"i has no children",

$f^i_j f^j$ translates to

"i has one child named j, who has no children",

$f^i_{jk} f^j f^k$ translates to

"i has two children named j and k", each of whom has no children,

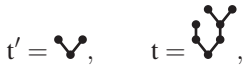
$f^i_j f^j_k f^k$ translates to

"i has a child j who has a child k" who has no children.

We can write these "kinship statements" as labelled trees shown, respectively, in the following diagrams

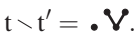
Subtrees, supertrees and prunings

For the two given trees,



(2.1 h)

t' is a subtree of t and t is a supertree of t', because t' can be constructed by deleting some of the vertices and edges from t or, alternatively, t can be constructed from t' by adding additional vertices and edges. The pruning associated with a tree and a possible supertree is the member of the forest space, written as $\mathrm{t} \smallsetminus \mathrm{t}'$, formed from the collection of possible ways t can be "pruned" to form t'. For example, for the trees given by (2.1 h),



In two further examples,

(2.1 i)

(2.1 j)

where the factor 2 in $\mathrm{t} \smallsetminus \mathrm{t}'$ (2.1 i) is a consequence of the fact that t' could equally well have been replaced by its mirror image. Note that $\mathrm{t} \smallsetminus \mathrm{t}'$ in (2.1 j) has two terms indicating that the pruning can be carried out in two different ways.

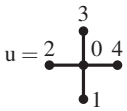
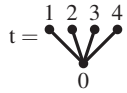
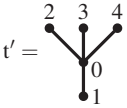
Chapter outline

Elementary properties of graphs, trees and forests are discussed in Sections 2.2 – 2.4. Functions of trees are introduced in Section 2.5 and partitions and evolution of trees in Section 2.6. The generalization known as the "stump" of a tree is introduced in Section 2.7.

Prunings of trees, and subtrees and supertrees, to be introduced in Section 2.8, are essential precursors to the composition rule to be introduced in Chapter 3. This is followed in 2.9 by a consideration of antipodes of trees and forests.

2.2 Rooted trees and unrooted (free) trees

To illustrate the relationship between rooted and unrooted trees, consider u, given by (2.1 b) (p. 40) and reproduced in (2.2 a). If $\mathrm{u} = (V,E)$ and $r \in V$, the triple (V,E,r) is a rooted tree. In this case, if $r = 0$ we obtain t and if $r = 1$ we obtain t′. Note that, because of symmetry, $r = 2,3,4$ we also obtain t′.

$$
\begin{aligned}
\mathrm{u} = &\ \text{  } && (V,E) = \big(\{0,1,2,3,4\},\big\{\{0,1\},\{0,2\},\{0,3\},\{0,4\}\big\}\big)\\
\mathrm{t} = &\ \text{  } && (V,E,r) = \big(\{0,1,2,3,4\},\big\{\{0,1\},\{0,2\},\{0,3\},\{0,4\}\big\},0\big)\\
\mathrm{t}' = &\ \text{  } && (V,E,r) = \big(\{0,1,2,3,4\},\big\{\{0,1\},\{0,2\},\{0,3\},\{0,4\}\big\},1\big)
\end{aligned}
\qquad (2.2\,\mathrm{a})
$$

Even though free trees, that is, trees without a root specified, have been introduced as a basic structure, we will now consider rooted trees as being the more fundamental constructs, and we will define unrooted trees in terms of rooted trees.

However, we will first introduce recursions to generate trees from the single tree with exactly one vertex.

Building blocks for trees

Let τ denote the unique tree with only a single vertex. We will show how to write an arbitrary $\mathrm{t} \in \mathrm{T}$ in terms of τ using additional operations.

Definition 2.2A The order of the tree t, denoted by $|\mathrm{t}|$, is the cardinality of the set of vertices. That is, if $\mathrm{t} = \{V,E,r\}$, then $|\mathrm{t}| = \mathrm{card}(V)$.

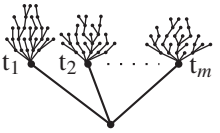
Using the B^+ operation

If trees $\mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_m$ are given, then

$$\mathrm{t} = [\mathrm{t}_1 \mathrm{t}_2 \ldots \mathrm{t}_m], \qquad (2.2\,\mathrm{b})$$

also written $B^+(\mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_m)$ because of the connection with Hopf algebras, will denote a tree formed by introducing a new vertex, which becomes the root of t, and m new edges from the root of t to each of the roots of t_i, $i = 1,2,\ldots,m$. Expressed diagrammatically, we have

t =



If a particular tree is repeated, this is shown using a power notation. For example, $[t_1^3 t_2^2]$ is identical to $[t_1 t_1 t_1 t_2 t_2]$. Similarly subscripts on a pair of brackets $[_m \cdots]_m$ will indicate repetition of this bracket pair. For example, $[_2 t_1 t_2 t_3 \cdots]_2$ has the same meaning as $[[t_1 t_2 t_3 \cdots]]$.

B^+ *induction*

Many results concerning trees can be proved by an induction principle which involves showing the result is true for $t = \tau$ and for $t = [f]$, where the forest f is assumed to consist of trees with order less than $|t|$. Thus, effectively, B^+ induction is induction on the value of $|t|$.

Balanced parentheses

By adjusting the terminology slightly we see that trees can be identified with balanced sequences of parentheses and forests by sequences of balanced parentheses sequences. To convert from the B^+ terminology it is only necessary to replace each occurrence of τ by $()$ and to replace each matching bracket pair $[\cdot]$ by the corresponding parenthesis pair $(\cdot)$.

For example,

$$\tau \mapsto ();$$
$$[\tau] \mapsto (());$$
$$[\tau^2] \mapsto (()());$$
$$[_2\tau]_2 \mapsto ((()));$$

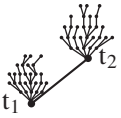
In the BNF notation, widely used in the theory of formal languages [75] (Naur, 1963), we can write

```
<tree> ::= (<forest>)
<forest> ::=     | <forest> <tree>
```

Using the beta-product

A second method of forming new trees from existing trees is to use the unsymmetrical product introduced in [14]. Given t_1 and t_2, the product $t_1 * t_2$ is formed by introducing a new edge between the two roots and defining the root of the product to be the same as the root of t_1. Hence, if $t = t_1 * t_2$ then we have the diagram

t =



This operation will be used frequently throughout this book and will be given the distinctive name of "beta-product".

Using iterated beta-products

Because the beta product is not associative, it has been customary to use parentheses to resolve ambiguities of precedence. For example, $(\tau * \tau) * \tau$ gives  and $\tau * (\tau * \tau)$ gives  . Ambiguity can also be avoided by the use of formal powers of the symbol $*$ so that $\mathrm{t}_1 *^2 \mathrm{t}_2\mathrm{t}_3 = \mathrm{t}_1 * *\mathrm{t}_2\mathrm{t}_3$ denotes $(\mathrm{t}_1 * \mathrm{t}_2) * \mathrm{t}_3$, whereas $\mathrm{t}_1 * \mathrm{t}_2 * \mathrm{t}_3$ will conventionally denote $\mathrm{t}_1 * (\mathrm{t}_2 * \mathrm{t}_3)$.

Table 1 Trees to order 4 with B^+ notation, beta products and Polish subscripts

$\lvert\mathrm{t}\rvert$	t	B^+	beta	Polish
1		τ	τ	0
2		$[\tau]$	$\tau * \tau$	10
3		$[\tau^2]$	$\tau *^2 \tau^2$	200
3		$[_2\tau]_2$	$\tau * \tau * \tau$	110
4		$[\tau^3]$	$\tau *^3 \tau^3$	3000
4		$[\tau[\tau]]$	$\tau *^2 \tau\tau * \tau = \tau *^2 \tau * \tau\tau$	$2010 = 2100$
4		$[_2\tau^2]_2$	$\tau * \tau *^2 \tau^2$	1200
4		$[_3\tau]_3$	$\tau * \tau * \tau * \tau$	1110

Using a Polish operator

Finally we propose a "Polish" type of tree construction. Denote by τ_n the operation, of forming the tree $[\mathrm{t}_1\mathrm{t}_2 \cdots \mathrm{t}_n]$, when acting on the sequence of operands t_1, t_2, …, t_n. This operation is written in Polish notation as

$$\tau_n \mathrm{t}_1 \mathrm{t}_2 \cdots \mathrm{t}_n.$$

From $[\tau] = \tau_1\tau$ we can for example write

$$[\tau[\tau]] = \tau_2\tau\tau_1\tau = \tau_2\tau_0\tau_1\tau_0,$$

where τ_0 is identified with τ. For compactness, this tree can be designated just by the subscripts; in this case 2010, or because the operation τ_2 is symmetric, as 2100. These Polish codes have been added to Table 1.

In the spirit of Polish notation, it is also convenient to introduce a prefix operator β which acts on a pair of trees to produce the beta-product of these trees. That is

$$\beta \mathrm{t}_1 \mathrm{t}_2 := \mathrm{t}_1 * \mathrm{t}_2.$$

Examples to order 4 are

$$\begin{aligned}
\tau &= \tau, \\
\beta\tau\tau &= [\tau], \\
\beta\beta\tau\tau\tau &= [\tau^2], \\
\beta\tau\beta\tau\tau &= [{}_2\tau]_2, \\
\beta\beta\beta\tau\tau\tau\tau &= [\tau^3], \\
\beta\beta\tau\beta\tau\tau\tau = \beta\beta\tau\tau\beta\tau\tau &= [\tau[\tau]], \\
\beta\tau\beta\beta\tau\tau\tau &= [{}_2\tau^2]_2, \\
\beta\tau\beta\tau\beta\tau\tau &= [{}_3\tau]_3.
\end{aligned}$$

Other uses of Polish notation are introduced in Chapter 3, Section 3.3 (p. 106).

Exercise 20 Write the following tree in a variety of notations such as (i) iterated use of the B^+ operation, (ii) iterated use of the beta-product, (iii) Polish notation using a sequence of factors τ, τ_1, τ_2, ...

t = 

An infix iterated beta operator

It is sometimes convenient to replace $*^n \mathrm{t}_1 \mathrm{t}_2 \cdots \mathrm{t}_n$ by $\circledast \mathrm{t}_1 \mathrm{t}_2 \cdots \mathrm{t}_n$. The operator $\circledast$ operates on whatever trees lie to its right, and parentheses are typically needed to indicate a departure from strict Polish form. For example,

$$(\tau_m \circledast \mathrm{t}_1 \mathrm{t}_2 \cdots \mathrm{t}_n)\mathrm{t}'_1 \mathrm{t}'_2 \cdots \mathrm{t}'_m = \tau_{m+n} \mathrm{t}_1 \mathrm{t}_2 \cdots \mathrm{t}_n \mathrm{t}'_1 \mathrm{t}'_2 \cdots \mathrm{t}'_m = [\mathrm{t}_1 \mathrm{t}_2 \cdots \mathrm{t}_n \mathrm{t}'_1 \mathrm{t}'_2 \cdots \mathrm{t}'_m].$$

Exercise 21 Write the tree $\tau_2 \circledast \left(\tau_3 \circledast \tau^3\right)^2$ in $B+$ and beta-prodict forms.

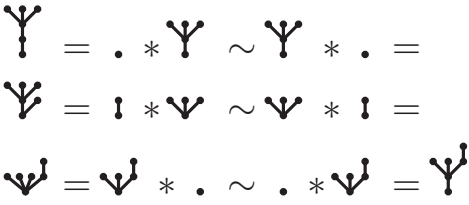
Equivalence classes of trees

If $\mathrm{t}_1 = \mathrm{t}' * \mathrm{t}''$, $\mathrm{t}_2 = \mathrm{t}'' * \mathrm{t}'$ for some t', t'' then we write $\mathrm{t}_1 \sim \mathrm{t}_2$. We extend this to an equivalence relation by defining $\mathrm{t}_1 \approx \mathrm{t}_n$ if there exist $\mathrm{t}_2, \ldots, \mathrm{t}_{n-1}$ such that

$$\mathrm{t}_1 \sim \mathrm{t}_2 \sim \mathrm{t}_3 \cdots \mathrm{t}_{n-1} \sim \mathrm{t}_n.$$

Definition 2.2B An unrooted tree u is an equivalence class of T. The set U is the partition of T into unrooted trees.

As an example, we show how four specific trees, with order 6, are connected using the $\sim$ relation.



Hence,

This defines the corresponding unrooted tree in terms of its equivalence class

S-trees and N-trees

In the theory of symplectic integration, as in Chapter 7 (p. 247), unrooted trees have an important role [84]. In this theory a distinction has to be made between two types of trees.

Definition 2.2C A tree t with the property that $t \sim t' * t'$, for some t', is said to be an S-tree. An equivalence class of S-trees is also referred to as an S-tree. An N-tree is a rooted or unrooted tree which is not an S-tree.

The terminology "S-tree" is based on their superfluous role in the order theory for symplectic integrators. Hence, following the terminology in [84], S-trees will also be referred to as "superfluous trees". Similarly N-trees are referred to as "non-superfluous trees."

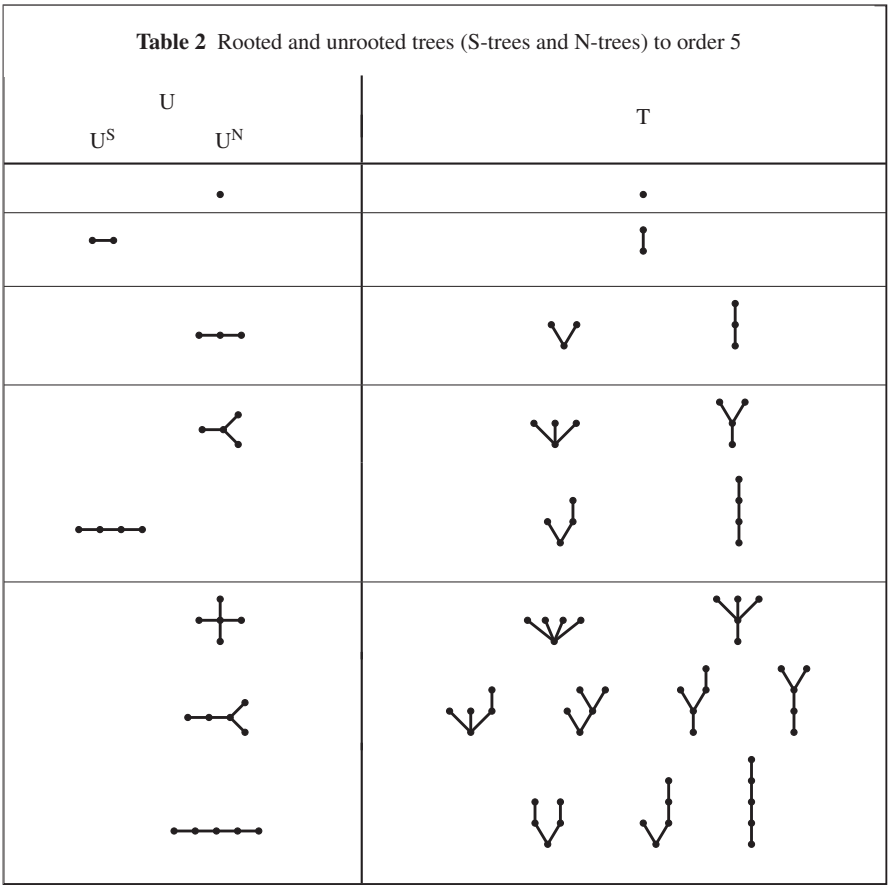
Unrooted trees to order 5

For orders 1 and 2 there is only a single tree and hence, the equivalence classes contain only these single members. For order 3, the two trees are similar to each other and constitute the only similarity class of this order. For order 4, the 4 trees break into two similarity classes and, for order 5, they break into 3 classes. We will adopt the

convention of drawing unrooted trees with all vertices spread out at approximately the same level so that there is no obvious root.

The unrooted trees, separated into S-trees and N-trees, together with the corresponding rooted trees, up to order 5, are shown in Table 2.

Table 2 Rooted and unrooted trees (S-trees and N-trees) to order 5



Exercise 22 For the following unrooted tree, find the four corresponding trees, in the style of Table 2. Show how these trees are connected by a sequence of $\sim$ links between the four rooted trees which make up the unrooted tree

Exercise 23 As for Exercise 22, find the five corresponding trees for the following unrooted tree, in the style of Table 2, and show the $\sim$ connections

Classifying trees by partitions

For a tree $\mathrm{t} = [\mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_m]$, with $|\mathrm{t}| = p$, the corresponding partition of t is the partition of $p-1$ given by

$$p - 1 = |\mathrm{t}_1| + |\mathrm{t}_2| + \cdots + |\mathrm{t}_m|.$$

The number of components in this partition, that is, the value of m, will be referred to as the "order of the partition".

Given the value of p, the set of all partitions of $p-1$ gives a convenient classification of the trees of the given order. For example, to identify the trees of order 5, and to list them in a systematic manner, first find the partitions of 4:

$$1+1+1+1, \quad 1+1+2, \quad 1+3, \quad 2+2, \quad 4$$

and then write down the required list of trees using this system.

The result of this listing procedure, up to $p = 6$, is shown in Table 3.

2.3 Forests and trees

A forest is a juxtaposition of trees. The set of forests F can be defined recursively by the generating statements

$$\begin{aligned} 1 &\in \mathrm{F}, \\ \mathrm{t}\,\mathrm{f} = \mathrm{f}\,\mathrm{t} &\in \mathrm{F}, \qquad \mathrm{t} \in \mathrm{T}, \quad \mathrm{f} \in \mathrm{F}, \\ \mathrm{f}_1\,\mathrm{f}_2 = \mathrm{f}_2\,\mathrm{f}_1 &\in \mathrm{F}, \qquad \mathrm{f}_1, \mathrm{f}_2 \in \mathrm{F}. \end{aligned}$$

The structure used in (2.2 b) can now be interpreted as a mapping from F to T

$$\mathrm{f} \mapsto [\mathrm{f}] \in \mathrm{T} \tag{2.3 a}$$

with the special case $[1] = \tau$.

We will sometimes use the terminology B^+ for the mapping in (2.3 a) together with the inverse mapping B^-. That is,

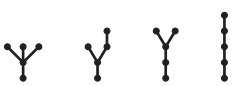
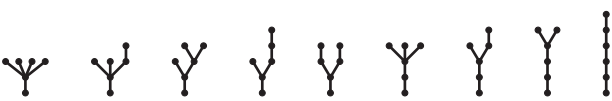
$$B^+(\mathrm{f}) = [\mathrm{f}] \in \mathrm{T}, \qquad B^-([\mathrm{f}]) = \mathrm{f} \in \mathrm{F}.$$

The order of a forest is an extension of Definition 2.2A (p. 44):

Definition 2.3A The order of $\mathrm{f} \in \mathrm{F}$ is defined by

$$\begin{aligned} |1| &= 0, \\ |\mathrm{t}_1\,\mathrm{t}_2 \cdots \mathrm{t}_m| &= |\mathrm{t}_1| + |\mathrm{t}_2| + \cdots + |\mathrm{t}_m|. \end{aligned}$$

Table 3 Trees up to order 6, classified by partitions

order	partition	trees
2	1	
3	1+1	
	2	
4	1+1+1	
	1+2	
	3	
5	1+1+1+1	
	1+1+2	
	1+3	
	2+2	
	4	
6	1+1+1+1+1	
	1+1+1+2	
	1+1+3	
	1+2+2	
	1+4	
	2+3	
	5	

Theorem 2.3B Amongst elementary consequences of Definitions 2.2A and 2.3A, we have

$$\begin{aligned} |\mathrm{t}\ \mathrm{f}| &= |\mathrm{t}| + |\mathrm{f}|, \\ |\mathrm{f}_1\ \mathrm{f}_2| &= |\mathrm{f}_1| + |\mathrm{f}_2|, \\ |\mathrm{t}_1 * \mathrm{t}_2| &= |\mathrm{t}_1| + |\mathrm{t}_2|, \\ |B^+(\mathrm{f})| &= 1 + |\mathrm{f}|. \end{aligned}$$

In contrast to the width of a forest, we will also use the "span" of a forest, according to the recursive definition:

Definition 2.3C The span of a forest f is defined by

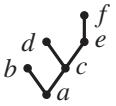
$$\begin{aligned} \big|1\big| &= 0, \\ \big|\mathrm{ft}\big| &= \big|\mathrm{f}\big| + 1, \quad \mathrm{f} \in \mathrm{F}, \quad \mathrm{t} \in \mathrm{T}. \end{aligned}$$

Miscellaneous tree terminology and metrics

The root of a tree t will be denoted by root(t). Every other vertex is a member of the set nonroot(t). Because a tree is connected and has no loops, there is a unique path from root(t) to any $v \in$ nonroot(t). The number of links in this path will be referred to as the "height" of v and the height of a tree is the maximum of the heights of its vertices. If the last link is the edge $\{v', v\}$ then v' is the "parent" of v and v is a "child" of v'. If v has no children it is a "leaf" of t. If v'' is present in the path from the root to v then v'' is an "ancestor" of v and v is a "descendant" of v''. The "dependants" of a vertex v are its descendants together with v itself. Every vertex, except the root, has a unique parent. The "width" of a tree is defined recursively by

$$\begin{aligned} \mathrm{width}(\tau) &= 1, \\ \mathrm{width}([\mathrm{t}_1 \mathrm{t}_2 \cdots \mathrm{t}_m]) &= \sum_{i=1}^{m} \mathrm{width}(\mathrm{t}_i). \end{aligned}$$

For the tree

$\mathrm{t} =$ 

the various relationships are given for the 6 vertices $a, b, \ldots, f$ as follows

v	height(v)	child(v)	parent(v)	descendant(v)	ancestor(v)
a	0	$\{b,c\}$	$\emptyset$	$\{b,c,d,e,f\}$	$\emptyset$
b	1	$\emptyset$	a	$\emptyset$	$\{a\}$
c	1	$\{d,e\}$	a	$\{d,e,f\}$	$\{a\}$
d	2	$\emptyset$	c	$\emptyset$	$\{a,c\}$
e	2	$\{f\}$	c	$\{f\}$	$\{a,c\}$
f	3	$\emptyset$	e	$\emptyset$	$\{a,c,e\}$

For this tree, root(t) $= a$, nonroot(t) $= \{b,c,d,e,f\}$, height(t) $=$ width(t) $= 3$.

2.4 Tree and forest spaces

The tree space

A formal linear combination of trees, including $\varnothing$, constitutes a member of the "tree space". The sum, and multiplication by a scalar, are defined by

$$\big(a_0\varnothing + \sum_{\mathrm{t}\in\mathrm{T}} a(\mathrm{t})\mathrm{t}\big) + \big(b_0\varnothing + \sum_{\mathrm{t}\in\mathrm{T}} b(\mathrm{t})\mathrm{t}\big) = (a_0+b_0)\varnothing + \sum_{\mathrm{t}\in\mathrm{T}} \big(a(\mathrm{t})+b(\mathrm{t})\big)\mathrm{t},$$
$$c\big(a_0\varnothing + \sum_{\mathrm{t}\in\mathrm{T}} a(\mathrm{t})\mathrm{t}\big) = ca_0\varnothing + \sum_{\mathrm{t}\in\mathrm{T}} ca(\mathrm{t})\mathrm{t}.$$

The forest space

A formal linear combination of forests will constitute a member of the "forest space" F. That is, $\mathsf{f} \in \mathsf{F}$, for an expression of the form

$$\mathsf{f} = \sum_{\mathrm{f}\in\mathrm{F}} a(\mathrm{f})\mathrm{f}, \qquad a : \mathrm{F} \to \mathbb{R}.$$

If $\phi : \mathbb{R} \to \mathbb{R}$ is given, with a Taylor expansion of the form

$$\phi(x) = a_0 + a_1 x + a_2 x^2 + \cdots,$$

then for a specific $\mathrm{t} \in \mathrm{T}$, $\mathsf{f} = \phi(\mathrm{t}) \in \mathsf{F}$ is defined by

$$\mathsf{f} = a_0 1 + a_1 \mathrm{t} + a_2 \mathrm{t}^2 + \cdots.$$

For example,

$$(1-\mathrm{t})^{-1} = \sum_{n=0}^{\infty} \mathrm{t}^n.$$

Theorem 2.4A The sum of all forests is given by the formal product

$$\prod_{\mathrm{t}\in\mathrm{T}}(1-\mathrm{t})^{-1}=\sum_{\mathrm{f}\in\mathrm{F}}\mathrm{f}. \tag{2.4 a}$$

The forest space as a ring

A ring is one of the fundamental algebraic structures. A unitary commutative ring is a set R with two binary operations, known as addition and multiplication, with the following properties

1. $(R,+)$ is an abelian group with identity 0.
2. $(R,\cdot)$ is associative and commutative with identity 1. That is, it is a commutative semi-group with identity (or a commutative monoid).
3. The left-distributive rule holds: $x(y+z)=xy+xz$

Without providing details, we assert that the forest space **F** is a unitary commutative ring.

Enumeration of trees and free trees

Introduce the three generating functions

$$\begin{aligned}\mathscr{A}(x)&=a_1x+a_2x^2+\cdots,\\ \mathscr{B}(x)&=b_1x+b_2x^2+\cdots,\\ \mathscr{C}(x)&=c_1x+c_2x^2+\cdots,\end{aligned} \tag{2.4 b}$$

where, for each $n=1,2,\ldots,$
a_n is the number of trees with order n,
b_n is the number of unrooted (free) trees with order n,
c_n is the number of non-superfluous unrooted trees with order n.

Low order terms in $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{C}$

By inspection of Table 2 (p. 49), we see that

$$\begin{array}{lllll} a_1=1, & a_2=1, & a_3=2, & a_4=4, & a_5=9,\\ b_1=1, & b_2=1, & b_3=1, & b_4=2, & b_5=3,\\ c_1=1, & c_2=0, & c_3=1, & c_4=1, & c_5=3.\end{array}$$

Hence we can insert the coefficients in (2.4 b):

$$\begin{aligned}\mathscr{A}(x)&=x+x^2+2x^3+4x^4+9x^5+\cdots,\\ \mathscr{B}(x)&=x+x^2+x^3+2x^4+3x^5+\cdots,\\ \mathscr{C}(x)&=x+x^3+x^4+3x^5+\cdots.\end{aligned}$$

Let ξ be the surjection from $\mathbf{F}$ to the space of power series in some indeterminant x such that $\xi(\mathrm{t}) \mapsto x^{|\mathrm{t}|}$, for any tree t. For example, $\xi(1-\mathrm{t}) = 1 - x^{|\mathrm{t}|}$. ξ is a homomorphism because, in additional to the required vector space properties,

$$\xi(\mathrm{ff}') = x^{|\mathrm{ff}'|} = x^{|\mathrm{f}|+|\mathrm{f}'|} = x^{|\mathrm{f}|}x^{|\mathrm{f}'|} = \xi(\mathrm{f})\xi(\mathrm{f}').$$

Theorem 2.4B The coefficients in $\mathscr{A}(x)$ are given by

$$a_1 + a_2x + a_3x^2 + \cdots = (1-x)^{-a_1}(1-x^2)^{-a_2}\ldots$$

Proof. Apply the operation ξ to each side of (2.4 a). The left-hand side maps to $\prod_{n=1}^{\infty}(1-x^n)^{-a_n}$ and, because the number of trees of order n is the same as the number of forests of order $n-1$ (note the equivalence $[\mathrm{f}] \leftrightarrow \mathrm{t}$), the right-hand side of (2.4 a) maps to $\sum_{n=1}^{\infty} a_n x^{n-1}$. □

Picking out the coefficients of the first few powers of x we obtain in turn

$$\begin{aligned} a_1 &= 1, \\ a_2 &= a_1 = 1, \\ a_3 &= \tfrac{1}{2}a_1(a_1+1) + a_2 = 2, \\ a_4 &= \tfrac{1}{6}a_1(a_1+1)(a_1+2) + a_1a_2 + a_3s = 4. \end{aligned}$$

The conclusions of Theorem 2.4B are used in Algorithm 1.

Algorithm 1 Find the number of trees of each order

```
Input:   p_max
Output:  Ntrees[1 : p_max]
    %
    %   Calculate the number of trees of each order 1 : p_max and place
    %   the results in Ntrees, using Theorem 2.4B
    %
 1  for i from 1 to p_max do
 2     Ntrees[i] ← 1
 3  end for
 4  for i from 2 to p_max − 1 do
 5     for j from p_max step −1 to 1 do
 6        a ← Ntrees[i]
 7        for k from 1 to floor((j − 1)/i) do
 8           Ntrees[j] ← Ntrees[j] + a ∗ Ntrees[j − i ∗ k]
 9           a ← a ∗ (Ntrees[i] + k)/(k + 1)
10        end for
11     end for
12  end for
```

Let $\widehat{a}_n = \sum_{i=1}^{n} a_i$ denote the number of trees with order not exceeding n. Values of a_n and $\widehat{a}_n$ up to $n = 10$ are given below using a Julia realization of Algorithm 1 (p. 55). Further information is available in Table 4 (p. 58)

n	1	2	3	4	5	6	7	8	9	10	11	12
a_n	1	1	2	4	9	20	48	115	286	719	1842	4766
$\widehat{a}_n$	1	2	4	8	17	37	85	200	486	1205	3047	7813

Exercise 24 Let $\widetilde{\mathrm{T}}$ denote the subset of T in which no vertex has exactly one descendant. That is, the first few members are

$$\widetilde{\mathrm{T}} = \Big\{ \bullet, \quad \Lambda, \quad \Psi, \quad \mathrm{Y}, \quad \ldots \Big\}.$$

If $\widetilde{a}_n$ is the number of members of $\widetilde{\mathrm{T}}$ with order n, show that

$$1 + x + \widetilde{a}_3 x^2 + \widetilde{a}_4 x^3 + \cdots = (1-x)^{-1} \prod_{n=3}^{\infty} (1-x^n)^{-\widetilde{a}_n}.$$

Lemma 2.4C

$$\mathscr{B}(x) - \mathscr{C}(x) = \mathscr{A}(x^2).$$

Proof. Because each superfluous tree is made up by joining two copies of a rooted tree, $b_{2n} - c_{2n} = a_n$, $b_{2n-1} - c_{2n-1} = 0$. □

Lemma 2.4D

$$\mathscr{B}(x) + \mathscr{C}(x) = 2\mathscr{A}(x) - \mathscr{A}(x)^2.$$

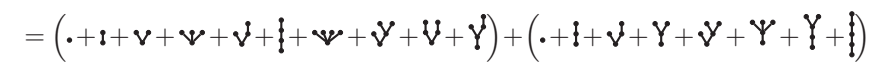
Proof. Let $\mathsf{T} \in \mathbf{T}$ be the sum of all rooted trees and consider

$$\mathsf{T}' = 2\mathsf{T} - \mathsf{T} * \mathsf{T}.$$

For any $\mathrm{u} \in \mathrm{U}^{\mathrm{N}}$, write the corresponding equivalence class of trees as a sequence t_1, $\mathrm{t}_2, \ldots, \mathrm{t}_n$, where $\mathrm{t}_{i-1} \sim \mathrm{t}_i$, $i = 1, 2, \ldots, n$. Hence, the total of the coefficients of these trees is $2n$, from the term $2\mathsf{T}$, minus $2n-2$ from the term $-\mathsf{T} * \mathsf{T}$, making a total of 2. If $\mathrm{u} \in \mathrm{U}^{\mathrm{S}}$, then the result is $2n - (2n-1) = 1$. Hence, $\xi(\mathsf{T}') = \mathscr{B}(x) + \mathscr{C}(x)$, which equals $\xi(2\mathsf{T} - \mathsf{T} * \mathrm{t})S = 2\mathscr{A}(x) - \mathscr{A}(x)^2$. □

To illustrate the proof of Lemma 2.4D, $2\mathsf{T} - \mathsf{T} * \mathsf{T}$ is evaluated, with only trees up to order 5 included. Note that trees of a higher order that appear in the manipulations are deleted.

$$2\Big(\bullet + \mathrm{I} + \Lambda + \mathrm{I} + \Psi + \mathrm{V} + \mathrm{Y} + \mathrm{I} + \Psi + \mathrm{V} + \mathrm{V} + \mathrm{V} + \mathrm{V} + \mathrm{Y} + \mathrm{Y} + \mathrm{I}\Big)$$
$$- \Big(\bullet + \mathrm{I} + \Lambda + \mathrm{I} + \Psi + \mathrm{V} + \mathrm{Y} + \mathrm{I}\Big) * \Big(\bullet + \mathrm{I} + \Lambda + \mathrm{I} + \Psi + \mathrm{V} + \mathrm{Y} + \mathrm{I}\Big)$$



A final result consists of two terms; the first has a representative of each equivalence class, with superfluous trees included, and the second term contains a different representative of each class but with superfluous trees omitted.

Finally the generating functions for members of U and U^S can be deduced.

Theorem 2.4E

$$\mathscr{B}(x) = \mathscr{A}(x) - \tfrac{1}{2}(\mathscr{A}(x)^2 - \mathscr{A}(x^2),$$
$$\mathscr{C}(x) = \mathscr{A}(x) - \tfrac{1}{2}(\mathscr{A}(x)^2 + \mathscr{A}(x^2).$$

Proof. Use Lemmas 2.4C and 2.4D. □

The calculation of the coefficients in $\mathscr{B}(x)$ and $\mathscr{C}(x)$ is shown in Algorithm 2. Results from Algorithms 1 (p. 55) and 2 are presented in Table 4. In this table, $a_n = Ntrees$, $b_n = Utrees$, $c_n = UNtrees$. Accumulated totals are denoted by $\widehat{a}$, $\widehat{b}$, $\widehat{c}$.

Algorithm 2 Find the number of unrooted and non-superfluous unrooted trees of each order

Input: *p_max* and *Ntrees* from Algorithm 1 (p. 55)
Output: *Utrees*, *UNtrees*

```
%
%   Calculate the number of unrooted trees of each order 1 : p_max and place
%   the results in Utrees, and the number of non-superfluous
%   unrooted trees and place these in UNtrees, using the result of Theorem 2.4E
%
for i from 1 to p_max do
    Xtrees[i] ← 0
    Ytrees[i] ← 0
end for
for i from 1 to p_max − 1 do
    for j from 1 to p_max − i do
        Xtrees[i + j] ← Xtrees[i + j] + Ntrees[i] ∗ Ntrees[j]
    end for
end for
for i from 1 to floor(p_max/2) do
    Ytrees[2 ∗ i] ← Ntrees[i]
end for
for i from 1 to p_max do
    Utrees[i] ← Ntrees[i] − (Xtrees[i] − Ytrees[i])/2
    UNtrees[i] ← Ntrees[i] − (Xtrees[i] + Ytrees[i])/2
end for
```

Table 4 Tree enumerations for rooted trees, unrooted trees, and non-superfluous unrooted trees, with accumulated totals.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a_n	1	1	2	4	9	20	48	115	286	719	1842	4766	12486	32973	87811
$\widehat{a}_n$	1	2	4	8	17	37	85	200	486	1205	3047	7813	20299	53272	141083
b_n	1	1	1	2	3	6	11	23	47	106	235	551	1301	3159	7741
$\widehat{b}_n$	1	2	3	5	8	14	25	48	95	201	436	987	2288	5447	13188
c_n	1	0	1	1	3	4	11	19	47	97	235	531	1301	3111	7741
$\widehat{c}_n$	1	1	2	3	6	10	21	40	87	184	419	950	2251	5362	13103

2.5 Functions of trees

A number of special functions on the set of trees have applications in Chapter 3. We will continue the present chapter by introducing these and, at the same time, establishing a standard ordering for the trees up to order 6.

For a tree t defined as the graph (V,E,r), consider a permutation π of the members of V which acts also on the vertex pairs comprising E and on the root r. That is, $v \in V$ maps to πv, an edge $\{v_1,v_2\}$ maps to $\{\pi v_1, \pi v_2\}$ and r maps to πr. In compact notation (V,E,r) maps to $(\pi V, \pi E, \pi r)$. Because π is a permutation, $\pi V = V$. If also $\pi E = E$ and $\pi r = r$, then π is said to be an automorphism of t. Define the group $A(\mathrm{t})$ as the group of automorphisms of t. Finally, we have the definition

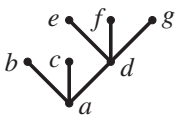
Definition 2.5A The symmetry of t, written as $\sigma(\mathrm{t})$, is the order of the group $A(\mathrm{t})$.

For example, consider the tree defined by

$$V = \{a,b,c,d,e,f,g\},$$
$$E = \{\{a,b\},\{a,c\},\{a,d\},\{d,e\},\{d,f\},\{d,g\}\},$$
$$r = a,$$

and represented by the diagram

t =



(2.5 a)

By inspection we see that the only automorphisms are the permutations for which a and d are invariant, the set $\{b,c\}$ is invariant and the set $\{e,f,g\}$ is invariant. Hence, $\sigma(t) = 2!3! = 12$.

Theorem 2.5B The function σ satisfies the recursion

$$\sigma(\tau) = 1,$$
$$\sigma(\mathrm{t}) = \prod_{i=1}^{m} k_i! \sigma(\mathrm{t}_i)^{k_i}, \qquad \mathrm{t} = [\mathrm{t}_1^{k_1} \mathrm{t}_2^{k_2} \cdots \mathrm{t}_m^{k_m}]. \tag{2.5 b}$$

Proof. Write the set of vertices of t given in (2.5 b) in the form

$$V = V_0 \cup \bigcup_{i=1}^{m} \bigcup_{j=1}^{k_m} V_{ij},$$

where V_0 contains the label assigned to the root and V_{ij} contains the labels assigned to copy number j of t_i. The permutations of the members of V which retain the connections, as represented by the set of all edges, consist of the compositions of two types of permutations. These are (1) the permutations within each of the k_i copies of t_i, for $i = 1, 2, \ldots, m$, and (2) for each $i = 1, 2, \ldots, m$, the permutations of the sets E_{ij}, $j = 1, 2, \ldots, k_i$, amongst the k_j copies of t_i. To contribute to the final result, (1) gives a factor of $\prod_{i=1}^{m} k_i!$ and (2) gives a factor of $\prod_{i=1}^{m} \sigma(\mathrm{t}_i)^{k_i}$. □

Another characterization of $\sigma(\mathrm{t})$ is found by attaching an integer S_i to each vertex i of t. Let the forest of descendant trees from i be $\mathrm{t}_{i1}^{k_{i1}} \mathrm{t}_{i2}^{k_{i2}} \cdots \mathrm{t}_{im_i}^{k_{im_i}}$, where t_{i1}, t_{i2}, …, t_{im_i} are distinct. Define

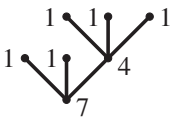
$$S_i = k_{i1}! k_{i2}! \cdots k_{im_i}!. \tag{2.5 c}$$

Corollary 2.5C With S_i defined by (2.5 c),

$$\sigma(\mathrm{t}) = \prod_i S_i.$$

Proof. The result follows by B^+ induction. □

The factorial (sometimes referred to as the density) is defined as the product of the number of dependants of each vertex. For t given by (2.5 a), the numbers of dependants is shown in (2.5 d), giving $\mathrm{t}! = 28$.

$\mathrm{t} =$ 

. (2.5 d)

It is convenient to give a formal definition in the form of a B+ recursion.

Definition 2.5D The factorial of t, written as t!, is

$$\tau! = 1,$$
$$\mathrm{t}! = ([\mathrm{t}_1\mathrm{t}_2\cdots\mathrm{t}_m])! = |\mathrm{t}|\prod_{i=1}^{m}(\mathrm{t}_i!).$$

A recursion based on the beta operation is sometimes useful.

Theorem 2.5E The factorial function satisfies the following recursion, where t and t′ are any trees,

$$\tau! = 1,$$
$$(\mathrm{t}*\mathrm{t}')! = \frac{\mathrm{t}!\,\mathrm{t}'!\,(|\mathrm{t}|+|\mathrm{t}'|)}{|\mathrm{t}|}.$$

We next introduce three combinatorial functions which have important applications in Chapter 3.

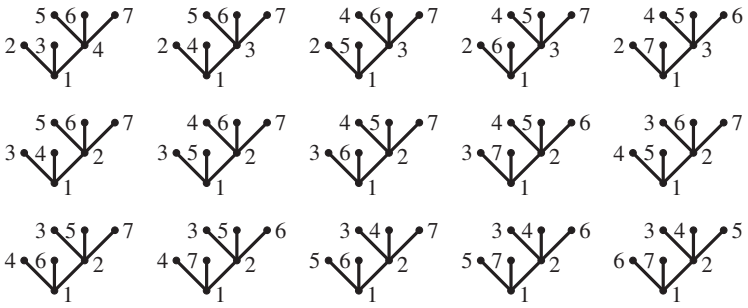
α For a given tree t with $|\mathrm{t}| = n$, $\alpha(\mathrm{t})$ is the number of distinct ways of writing t in the form (V,E,r), where V is a permutation of $\{1,2,\ldots,n\}$ and

every vertex is assigned a lower label than each of its descendants. (2.5 e)

β For a given tree t with $|\mathrm{t}| = n$, $\beta(\mathrm{t})$ is the number of distinct ways of writing t in the form (V,E,r), where V is a permutation of $\{1,2,\ldots,n\}$; there is no requirement to satisfy (2.5 e).

$\overline{\beta}$ For a given tree t with $|\mathrm{t}| = n$, $\overline{\beta}(\mathrm{t})$ is the number of distinct ways of writing t in the form (V,E,r), where V is a permutation of $\{1,2,\ldots,n\}$; where there is no requirement to satisfy (2.5 e) but the root is always labelled 1.

For t given by (2.5 a), $\alpha(\mathrm{t}) = 15$. The labelled trees contributing to this total are



The value of $\beta(\mathrm{t})$ is found to be 420 and $\overline{\beta}(\mathrm{t})$ to be 60. The general results are

Theorem 2.5F

$$\alpha(\mathrm{t}) = \frac{|t|!}{\sigma(\mathrm{t})\mathrm{t}!}, \tag{2.5 f}$$

$$\beta(\mathrm{t}) = \frac{|t|!}{\sigma(\mathrm{t})}, \tag{2.5 g}$$

$$\overline{\beta}(\mathrm{t}) = \frac{(|t|-1)!}{\sigma(\mathrm{t})}. \tag{2.5 h}$$

Proof. The value of $\beta(\mathrm{t})$ given by (2.5 g) is found by assigning labels to the vertices in all possible ways and then dividing by $\sigma(\mathrm{t})$ to compensate for duplicated numbering of the same tree. The factor t! in the denominator of (2.5 f) compensates for the fact that each vertex v must receive the lowest label amongst the dependants of v. Similarly, (2.5 h) is found by dividing (2.5 g) by $|\mathrm{t}|$ because of the allocation of 1 to the root. □

Standard numbering of trees

We will need to make use of individual rooted trees throughout this volume and it will be convenient to assign standardized numbers. Denote tree number n in the sequence of numbered trees as $\mathbf{t}_n$. We will now describe the procedure for constructing the sequence of all trees in the standard order.

Starting with $\mathbf{t}_1 = \tau$, we generate trees of orders $2, 3, \ldots$ recursively. For a given order $p > 1$ suppose all trees of lower order have been assigned sequence numbers. To generate the numbered trees of order p, write $\mathrm{t} = \mathbf{t}_\ell * \mathbf{t}_r$, where r runs through the sequence numbers of previously assigned trees of order less than p. For each choice of r, ℓ runs, in order, through all integers such that $|\mathbf{t}_\ell| + |\mathbf{t}_r| = p$.

If t constructed by this process has not been assigned a sequence number already, then it is given the next number available, say n, and we write $L(n) = \ell$ and $R(n) = r$. However, if it is found that $\mathrm{t} = \mathbf{t}_\ell * \mathbf{t}_r$ is identical with an existing numbered tree, then t is regarded as a duplicate and no new number needs to be assigned.

While the numbers are being assigned, it is also convenient to assign standard factors to each tree $\mathbf{t}_n$, in the form $\mathbf{t}_{L(n)}$ and $\mathbf{t}_{R(n)}$ say. This is done the first time each new tree arises and the same factors automatically apply to any duplicates.

Table 5 (p. 62) illustrates how this procedure works for trees up to order 4. The values of $\mathrm{t}' = \mathbf{t}_\ell$ and $\mathrm{t}'' = \mathbf{t}_r$ are shown for each tree together with its sequence number. However, in the case that this is a duplicate, no diagram for the tree is given and no entries are given for L or R.

In the numbering system based on these rules, the partition orders, for a given tree order, are automatically in decreasing order as was illustrated in Table 3 (p. 51). The standard numbering is shown to order 6 in Table 6 (p. 63), classified according to the partition order.

For example, the numbered trees up to order 4 are

Table 5 Illustration of tree numbering to order 4, showing the beta-product and unique factorization

t		t′ ∗ t″	L(t)	R(t)
$\mathbf{t}_1$				
$\mathbf{t}_2$		$\mathbf{t}_1 * \mathbf{t}_1$	1	1
$\mathbf{t}_3$		$\mathbf{t}_2 * \mathbf{t}_1$	2	1
$\mathbf{t}_4$		$\mathbf{t}_1 * \mathbf{t}_2$	1	2
$\mathbf{t}_5$		$\mathbf{t}_3 * \mathbf{t}_1$	3	1
$\mathbf{t}_6$		$\mathbf{t}_4 * \mathbf{t}_1$	4	1
$\mathbf{t}_6$		$\mathbf{t}_2 * \mathbf{t}_2$		
$\mathbf{t}_7$		$\mathbf{t}_1 * \mathbf{t}_3$	1	3
$\mathbf{t}_8$		$\mathbf{t}_1 * \mathbf{t}_4$	1	4

$\mathbf{t}_1 =$ $\mathbf{t}_2 =$ $\mathbf{t}_3 =$  $\mathbf{t}_4 =$ 

$\mathbf{t}_5 =$  $\mathbf{t}_6 =$  $\mathbf{t}_7 =$  $\mathbf{t}_8 =$ 

A Julia realization of Algorithm 3 (p. 64), using $p\_max = 5$, gives the expected results:

$$\begin{bmatrix} L[2] & L[3] & \cdots & L[17] \\ R[2] & R[3] & \cdots & R[17] \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 3 & 4 & 1 & 1 & 5 & 6 & 7 & 8 & 4 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 & 3 & 4 & 1 & 1 & 1 & 1 & 2 & 5 & 6 & 7 & 8 \end{bmatrix}$$

$$\begin{bmatrix} prod[1,1] & prod[1,2] & \cdots & prod[1,7] & prod[1,8] \\ prod[2,1] & prod[2,2] & \cdots & prod[2,7] & \\ \vdots & & & & \\ prod[7,1] & & & & \\ prod[8,1] & & & & \end{bmatrix} = \begin{bmatrix} 2 & 4 & 7 & 8 & 14 & 15 & 16 & 17 \\ 3 & 6 & 11 & 12 & & & & \\ 5 & 10 & & & & & & \\ 6 & 13 & & & & & & \\ 9 & & & & & & & \\ 10 & & & & & & & \\ 11 & & & & & & & \\ 12 & & & & & & & \end{bmatrix},$$

where the blank entries are not relevant to this algorithm if the maximum order is 5.

Table 6 Standard numbering of trees to order 6 classified by the order of their partitions

$\mathbf{t}_1 =$

$\mathbf{t}_2 =$ 

$\mathbf{t}_3 =$ 

$\mathbf{t}_4 =$ 

$\mathbf{t}_5 =$ 

$\mathbf{t}_6 =$ 

$\mathbf{t}_7 =$ 

$\mathbf{t}_8 =$ 

$\mathbf{t}_9 =$ 

$\mathbf{t}_{10} =$ 

$\mathbf{t}_{11} =$ 

$\mathbf{t}_{12} =$ 

$\mathbf{t}_{13} =$ 

$\mathbf{t}_{14} =$ 

$\mathbf{t}_{15} =$ 

$\mathbf{t}_{16} =$ 

$\mathbf{t}_{17} =$

$\mathbf{t}_{18} =$ 

$\mathbf{t}_{19} =$ 

$\mathbf{t}_{20} =$ 

$\mathbf{t}_{21} =$ 

$\mathbf{t}_{22} =$ 

$\mathbf{t}_{23} =$ 

$\mathbf{t}_{24} =$ 

$\mathbf{t}_{25} =$ 

$\mathbf{t}_{26} =$ 

$\mathbf{t}_{27} =$ 

$\mathbf{t}_{28} =$ 

$\mathbf{t}_{29} =$ 

$\mathbf{t}_{30} =$ 

$\mathbf{t}_{31} =$ 

$\mathbf{t}_{32} =$ 

$\mathbf{t}_{33} =$ 

$\mathbf{t}_{34} =$ 

$\mathbf{t}_{35} =$ 

$\mathbf{t}_{36} =$ 

$\mathbf{t}_{37} =$

Algorithm 3 Generate trees

```
Input:   p_max
Output:  order, first, last, L, R, prod
    %
    %   order[1:last[p_max]] is the vector of orders of each tree
    %   first[1 : p_max] is the vector of first serial numbers for each order
    %   last[1 : p_max] is the vector of last serial numbers for each order
    %   L[2 : p_max] is the vector of left factors for each tree
    %   R[2 : p_max] is the vector of right factors for each tree
    %   prod is the array of products of trees ℓ and r such that order[ℓ] + order[r] ≤ p_max
    %
 1  first[1] ← 1
 2  last[1] ← 1
 3  order[1] ← 1
 4  n ← 1
 5  for p from 2 to p_max do
 6     first[p] ← n + 1
 7     for r from 1 to last[p − 1] do
 8        for ℓ from first[p − order[r]] to last[p − order[r]] do
 9           if ℓ = 1 or r ≤ R[ℓ] then
10              n ← n + 1
11              prod[ℓ, r] ← n
12              L[n] ← ℓ
13              R[n] ← r
14              order[n] ← p
15           else
16              prod[l, r] ← prod[prod[L[ℓ], r], R[ℓ]]
17           end if
18        end for
19     end for
20     last[p] ← n
21  end for
```

The standard numbering of the 37 trees to order 6 is shown in Table 7 (p. 66). Also shown are $\sigma(\mathrm{t})$ and t! for each tree t.

Generation of various tree functions

We will temporarily introduce *Bminus* to denote some aspects of the structure of a tree t. Let $n = |\mathrm{t}|$, then $Bminus(\mathrm{t})$ will denote a vector with dimension $m := last[n-1]$, where $last[n]$ is the total number of trees with order up to n. Thus *last* is one of the results computed in Algorithm 3. The elements of $Bminus(\mathrm{t})$ are the exponents k_1, k_2, …, k_m such that

$$\mathrm{t} = \left[\begin{array}{cccc} \mathbf{t}_1^{k_1} & \mathbf{t}_2^{k_2} & \cdots & \mathbf{t}_m^{k_m} \end{array} \right].$$

We will present Algorithm 4 to generate values of *Bminus* together with the *factorial* and *symmetry* functions.

Algorithm 4 Generate tree functions

Input: *order*, *first*, *last*, *L*, *R*, *prod*, *p_max* from Algorithm 3
Output: *Bminus*, *factorial*, *symmetry*

```
    %
    %   Bminus[n][1 : order[n] − 1], n = 1 : lastp_top, is the sequence of Bminus values
    %   factorial[1 : last[p_top] is the array of factorial values
    %   symmetry[1 : last[p_top] is the array of symmetry values
    %
 1  Bminus[1] ← []
 2  factorial[1] ← 1
 3  symmetry[1] ← 1
 4  for n from 2 to last[p_max] do
 5     factorial[n] ← factorial[L[n]] ∗ factorial[R[n]] ∗ order[n]/order[L[n]]
 6     Bminus[n] = pad(last[order[n]], Bminus[L[n]])
 7     Bminus[n][R[n]] ← Bminus[n][R[n]] + 1
 8     symmetry[n] = symmetry[L[n]] ∗ symmetry[R[n]] ∗ Bminus[n][R[n]]
 9  end for
```

A realization in Julia, up to order 5, gave the results

$$
\begin{aligned}
symmetry &= \{1,1,2,1,6,1,2,1,24,2,2,1,2,6,1,2,1\}\\
factorial &= \{1,2,3,6,4,8,12,24,5,10,15,30,20,20,40,60,120\}\\
Bminus &= \big\{\{\},\{1\},\{2,0\},\{0,1\},\{3,0,0,0\},\{1,1,0,0\},\{0,0,1,0\},\{0,0,0,1\},\\
&\qquad \{4,0,0,0,0,0,0,0\},\{2,1,0,0,0,0,0,0\},\{1,0,1,0,0,0,0,0\},\\
&\qquad \{1,0,0,1,0,0,0,0\},\{0,2,0,0,0,0,0,0\},\{0,0,0,0,1,0,0,0\},\\
&\qquad \{0,0,0,0,0,1,0,0\},\{0,0,0,0,0,0,1,0\},\{0,0,0,0,0,0,0,1\}\big\}
\end{aligned}
$$

2.6 Trees, partitions and evolutions

For a tree $\mathrm{t} = [\mathrm{t}_1\mathrm{t}_2\cdots\mathrm{t}_m]$, with $|\mathrm{t}| = n+1$, the set $\{|\mathrm{t}_1|,|\mathrm{t}_2|,\ldots,|\mathrm{t}_m|\}$ is a partition of n into m components, in the sense that

$$|\mathrm{t}_1| + |\mathrm{t}_2| + \cdots + |\mathrm{t}_m| = n.$$

In this section, we will consider partitions of sets and numbers and their relationship with trees. Partitions have a central role in the study of B-series in Chapter 3, with specific applications in Section 3.5 (p. 117).

Partitions of numbers and sets

Given a finite set S with cardinality n, it is convenient to distinguish between the partitions of S, denoted by $\mathscr{P}[S]$, and of n, denoted by $\mathscr{P}(n)$.

Table 7 Trees to order 6 showing $\sigma(t)$ and t!

$\|t\|$	t			$\sigma(t)$	t!
1	$\mathbf{t}_1$	τ		1	1
2	$\mathbf{t}_2$	$[\tau]$		1	2
3	$\mathbf{t}_3$	$[\tau^2]$		3	3
3	$\mathbf{t}_4$	$[_2\tau]_2$		1	6
4	$\mathbf{t}_5$	$[\tau^3]$		6	4
4	$\mathbf{t}_6$	$[\tau[\tau]]$		1	8
4	$\mathbf{t}_7$	$[_2\tau^2]_2$		2	12
4	$\mathbf{t}_8$	$[_3\tau]_3$		1	24
5	$\mathbf{t}_9$	$[\tau^4]$		24	5
5	$\mathbf{t}_{10}$	$[\tau^2[\tau]]$		2	10
5	$\mathbf{t}_{11}$	$[\tau[\tau^2]]$		2	15
5	$\mathbf{t}_{12}$	$[\tau[_2\tau]_2]$		1	30
5	$\mathbf{t}_{13}$	$[[\tau]^2]$		2	20
5	$\mathbf{t}_{14}$	$[_2\tau^3]_2$		6	20
5	$\mathbf{t}_{15}$	$[_2\tau[\tau]]_2$		1	40
5	$\mathbf{t}_{16}$	$[_3\tau^2]_3$		2	60
5	$\mathbf{t}_{17}$	$[_4\tau]_4$		1	120
6	$\mathbf{t}_{18}$	$[\tau^5]$		120	6
6	$\mathbf{t}_{19}$	$[\tau^3[\tau]]$		6	12
6	$\mathbf{t}_{20}$	$[\tau^2[\tau^2]]$		4	18
6	$\mathbf{t}_{21}$	$[\tau^2[_2\tau]_2]$		2	36
6	$\mathbf{t}_{22}$	$[\tau[\tau]^2]$		2	24
6	$\mathbf{t}_{23}$	$[\tau[\tau^3]]$		6	24
6	$\mathbf{t}_{24}$	$[\tau[\tau[\tau]]]$		1	48
6	$\mathbf{t}_{25}$	$[\tau[_2\tau^2]_2]$		2	72
6	$\mathbf{t}_{26}$	$[\tau[_3\tau]_3]$		1	144
6	$\mathbf{t}_{27}$	$[[\tau][\tau^2]]$		2	36
6	$\mathbf{t}_{28}$	$[[\tau][_2\tau]_2]$		1	72
6	$\mathbf{t}_{29}$	$[\tau[_2\tau^2]_2]$		24	30
6	$\mathbf{t}_{30}$	$[\tau[_3\tau]_3]$		2	60
6	$\mathbf{t}_{31}$	$[[\tau][\tau^2]]$		2	90
6	$\mathbf{t}_{32}$	$[[\tau][_2\tau]_2]$		1	180
6	$\mathbf{t}_{33}$	$[_2[\tau]^2]_2$		2	120
6	$\mathbf{t}_{34}$	$[_3\tau^3]_3$		6	120
6	$\mathbf{t}_{35}$	$[_3\tau[\tau]]_3$		1	240
6	$\mathbf{t}_{36}$	$[_4\tau^2]_4$		2	360
6	$\mathbf{t}_{37}$	$[_5\tau]_5$		1	720

Table 8 Systematic generation of trees to order 6

$\mathbf{t}_1$			
$\mathbf{t}_2 = \mathbf{t}_1 * \mathbf{t}_1$			
 $\mathbf{t}_3 = \mathbf{t}_2 * \mathbf{t}_1$	$\mathbf{t}_4 = \mathbf{t}_1 * \mathbf{t}_2$		
 $\mathbf{t}_5 = \mathbf{t}_3 * \mathbf{t}_1$ $\mathbf{t}_8 = \mathbf{t}_1 * \mathbf{t}_4$	 $\mathbf{t}_6 = \mathbf{t}_4 * \mathbf{t}_1$	$\mathbf{t}_6 = \mathbf{t}_2 * \mathbf{t}_2$	 $\mathbf{t}_7 = \mathbf{t}_1 * \mathbf{t}_3$
 $\mathbf{t}_9 = \mathbf{t}_5 * \mathbf{t}_1$ $\mathbf{t}_{10} = \mathbf{t}_3 * \mathbf{t}_2$  $\mathbf{t}_{14} = \mathbf{t}_1 * \mathbf{t}_5$	 $\mathbf{t}_{10} = \mathbf{t}_6 * \mathbf{t}_1$  $\mathbf{t}_{13} = \mathbf{t}_4 * \mathbf{t}_2$  $\mathbf{t}_{15} = \mathbf{t}_1 * \mathbf{t}_6$	 $\mathbf{t}_{11} = \mathbf{t}_7 * \mathbf{t}_1$ $\mathbf{t}_{11} = \mathbf{t}_2 * \mathbf{t}_3$  $\mathbf{t}_{16} = \mathbf{t}_1 * \mathbf{t}_7$	 $\mathbf{t}_{12} = \mathbf{t}_8 * \mathbf{t}_1$ $\mathbf{t}_{12} = \mathbf{t}_2 * \mathbf{t}_4$ $\mathbf{t}_{17} = \mathbf{t}_1 * \mathbf{t}_8$
 $\mathbf{t}_{18} = \mathbf{t}_9 * \mathbf{t}_1$  $\mathbf{t}_{22} = \mathbf{t}_{13} * \mathbf{t}_1$  $\mathbf{t}_{26} = \mathbf{t}_{17} * \mathbf{t}_1$  $\mathbf{t}_{28} = \mathbf{t}_8 * \mathbf{t}_2$ $\mathbf{t}_{28} = \mathbf{t}_4 * \mathbf{t}_4$ $\mathbf{t}_{26} = \mathbf{t}_2 * \mathbf{t}_8$  $\mathbf{t}_{32} = \mathbf{t}_1 * \mathbf{t}_{12}$  $\mathbf{t}_{36} = \mathbf{t}_1 * \mathbf{t}_{16}$	 $\mathbf{t}_{19} = \mathbf{t}_{10} * \mathbf{t}_1$  $\mathbf{t}_{23} = \mathbf{t}_{14} * \mathbf{t}_1$ $\mathbf{t}_{19} = \mathbf{t}_5 * \mathbf{t}_2$ $\mathbf{t}_{20} = \mathbf{t}_3 * \mathbf{t}_3$ $\mathbf{t}_{23} = \mathbf{t}_2 * \mathbf{t}_5$  $\mathbf{t}_{29} = \mathbf{t}_1 * \mathbf{t}_9$  $\mathbf{t}_{33} = \mathbf{t}_1 * \mathbf{t}_{13}$ $\mathbf{t}_{37} = \mathbf{t}_1 * \mathbf{t}_{17}$	 $\mathbf{t}_{20} = \mathbf{t}_{11} * \mathbf{t}_1$  $\mathbf{t}_{24} = \mathbf{t}_{15} * \mathbf{t}_1$ $\mathbf{t}_{22} = \mathbf{t}_6 * \mathbf{t}_2$ $\mathbf{t}_{28} = \mathbf{t}_4 * \mathbf{t}_3$ $\mathbf{t}_{24} = \mathbf{t}_2 * \mathbf{t}_6$  $\mathbf{t}_{30} = \mathbf{t}_1 * \mathbf{t}_{10}$  $\mathbf{t}_{34} = \mathbf{t}_1 * \mathbf{t}_{14}$	 $\mathbf{t}_{21} = \mathbf{t}_{12} * \mathbf{t}_1$  $\mathbf{t}_{25} = \mathbf{t}_{16} * \mathbf{t}_1$  $\mathbf{t}_{27} = \mathbf{t}_7 * \mathbf{t}_2$ $\mathbf{t}_{21} = \mathbf{t}_3 * \mathbf{t}_4$ $\mathbf{t}_{25} = \mathbf{t}_2 * \mathbf{t}_7$  $\mathbf{t}_{31} = \mathbf{t}_1 * \mathbf{t}_{11}$  $\mathbf{t}_{35} = \mathbf{t}_1 * \mathbf{t}_{15}$

Definition 2.6A A partition P of a set S is a set of subsets $P = \{S_1, S_2, \ldots, S_m\}$, such that for $i \neq j \in \{1, 2, \ldots, m\}$, $S_i \cap S_j = \emptyset$ and such that $\bigcup_{i=1}^{m} S_i = S$. A partition p of a positive integer n is a set of integers $p = \{p_1, p_2, \ldots, p_m\}$, where repetitions are permitted, such that $n = \sum_{i=1}^{m} p_i$. If the components are permuted, it is regarded as the same partition.

For example, if $S = \{1, 2, 3, 4\}$, then the members of $\mathscr{P}[S]$ are $\{\{1\}, \{2\}, \{3\}, \{4\}\}$, $\{\{1\}, \{2\}, \{3, 4\}\}$ and 13 further partitions. We will, for reasons of brevity, write these in a compressed notation, starting with: 1+2+3+4, 1+2+34 ... . The complete list is

$$\begin{aligned}
&\text{(line 1)} \quad 1{+}2{+}3{+}4,\\
&\text{(line 2)} \quad 1{+}2{+}34,\ 1{+}3{+}24,\ 1{+}4{+}23,\ 2{+}3{+}14,\ 2{+}4{+}13,\ 3{+}4{+}12,\\
&\text{(line 3)} \quad 1{+}234, 2{+}134, 3{+}124, 4{+}123,\\
&\text{(line 4)} \quad 12{+}34,\ 13{+}24,\ 14{+}23,\\
&\text{(line 5)} \quad 1234,
\end{aligned} \tag{2.6 a}$$

where the five lines correspond to the five members of $\mathscr{P}(4)$:

$$\begin{aligned}
&\text{(line 1)} \quad 1{+}1{+}1{+}1,\\
&\text{(line 2)} \quad 1{+}1{+}2,\\
&\text{(line 3)} \quad 1{+}3,\\
&\text{(line 4)} \quad 2{+}2,\\
&\text{(line 5)} \quad 4.
\end{aligned} \tag{2.6 b}$$

Definition 2.6B Let $P = \{S_1, S_2, \ldots, S_m\}$, be a partition of a set S, then $\phi(P)$ is the corresponding partition of $n = \text{card}(P)$, where

$$\phi(P) = \{\text{card}(S_1), \text{card}(S_2), \ldots, \text{card}(S_m)\}.$$

Furthermore, if p is a partition of the integer n then the p-weight is

$$\text{p-weight}(p) := \text{card}(\phi^{-1}(p)).$$

As examples of these ideas, $\phi(P)$ for any of the partitions in line $i = 1, 2, 3, 4, 5$ of (2.6 a) are given in the corresponding lines of (2.6 b). Furthermore, by comparing (2.6 a) and (2.6 b), we see that

$$\begin{aligned}
&\text{p-weight}(1{+}1{+}1{+}1) = 1, \quad \text{p-weight}(1{+}1{+}2) = 6, \quad \text{p-weight}(1{+}3) = 4,\\
&\text{p-weight}(2{+}2) = 3, \qquad \text{p-weight}(4) = 1.
\end{aligned}$$

In referring to a particular partition of n, it will often be convenient to specify the number of repetitions of each particular component. Thus we will write

$$1^{n_1}+2^{n_2}+\cdots+m^{n_m},$$

as an alternative way of writing

$$\overbrace{1+1+\cdots+1}^{n_1}+\overbrace{2+2+\cdots+2}^{n_2}+\cdots+\overbrace{m+m+\cdots+m}^{n_m}$$

to specify the number of occurrences of each part. In this notation, k^{n_j} may be omitted from the sum if $n_j = 0$, for $1 \le j \le m$. For example, the 7 partitions of 5 can be written in several alternative but equivalent ways, such as

$$\begin{aligned}
1+1+1+1+1 &= 1^5,\\
1+1+1+2 &= 1^3+2,\\
1+1+3 &= 1^2+2^0+3 = 1^2+3,\\
1+2+2 &= 1+2^2,\\
1+4 &= 1+2^0+3^0+4,\\
2+3 &= 1^0+2+3,\\
5 &= 1^0+2^0+3^0+4^0+5.
\end{aligned}$$

The value of p-weight

Using this terminology, it is possible to find a formula for p-weight.

Theorem 2.6C Let $p = 1^{n_1}+2^{n_2}+\cdots+m^{n_m} \in \mathscr{P}(n)$. Then

$$\text{p-weight}(p) = \frac{n!}{\prod_{i=1}^{m} n_i!(i!)^{n_i}}.$$

Proof. Given $P \in \mathscr{P}[S]$, with $\text{card}(S) = n$, let

$$P = \bigcup_{i=1}^{m}\bigcup_{j=1}^{n_i} S_{ij}, \qquad \text{card}(S_{ij}) = i,$$

where $\phi(P) = 1^{n_1}+2^{n_2}+\cdots+m^{n_m} \in \mathscr{P}(n)$, The number of ways of assigning n labels to the sets S_{ij} is given by the multinomial coefficient $n!/\prod_{i=1}^{m}(i!)^{n_i}$. However, for each $i = 1,2,\ldots,m$, the n_i sets S_{ij}, $j = 1,2,\ldots,n_i$ can be permuted without altering the set-partition. Hence, to compensate for the over-counting, it is necessary to divide by $i!$. □

Exercise 25 Find the partitions of $\{\mathbf{1},\mathbf{2},\mathbf{3},\mathbf{4},\mathbf{5}\}$ such that $\phi(P) = 1+2^2$.

Evolution of partitions of numbers and sets

Evolution of $\mathscr{P}[S]$ for an embedded sequence of sets

Let $S_1 \subset S_2 \subset \cdots$ be a sequence of sets, such that $\text{card}(S_n) = n$, $n = 1, 2, \ldots$ Equivalently, given a sequence of distinct symbols $s_1, s_2, \ldots$, define $S_n = \{s_1, s_2, \ldots, s_n\}$. In the examples given below, we will use the sequence starting with $\{1, 2, 3, 4, 5, \ldots\}$.

Definition 2.6D Given partitions $P \in \mathscr{P}[S_{n-1}]$, $P' \in \mathscr{P}[S_n]$, P is the parent of P' and P' is a child of P if (i) $\text{card}(P) = \text{card}(P') - 1$ and $P' = P + \{s_n\}$, or (ii) P_0, P_1, P_2 exist such that

$$P = P_0 \cup \{P_1\},$$
$$P' = P_0 \cup \{P_2\},$$
$$P_2 = P_1 \cup \{s_n\}.$$

We will explore the evolutionary structure of the sequence of partitions $\mathscr{P}[S_n], n = 1, 2, \ldots$, as we move from step to step and indicate with the notation $P \longrightarrow P'$ that P' is a child of P.

Given $P \in \mathscr{P}[S_{n-1}]$, the expression evolve$(P)$ will denote the formal sum of the children of P. Furthermore, evolve acting on a formal sum of members of $\mathscr{P}[S_{n-1}]$ will denote

$$\text{evolve}\left(\sum_{i=1}^{m} P_i\right) = \sum_{i=1}^{m} \text{evolve}(P_i).$$

For the sequence, $\{1\}, \{1, 2\}, \{1, 2, 3\}, \{1, 2, 3, 4\}$, this structure is shown in Figure 4.

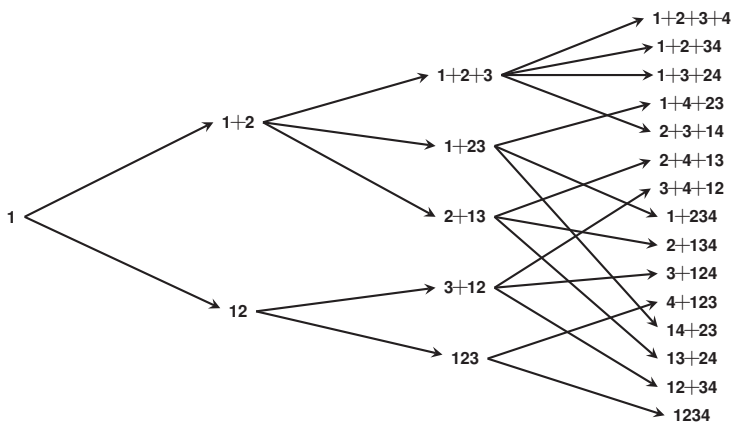


Figure 4 Evolution of the partitions from $\mathscr{P}[\{1\}]$, through $\mathscr{P}[\{1, 2\}]$, $\mathscr{P}[\{1, 2, 3\}]$ to $\mathscr{P}[\{1, 2, 3, 4\}]$

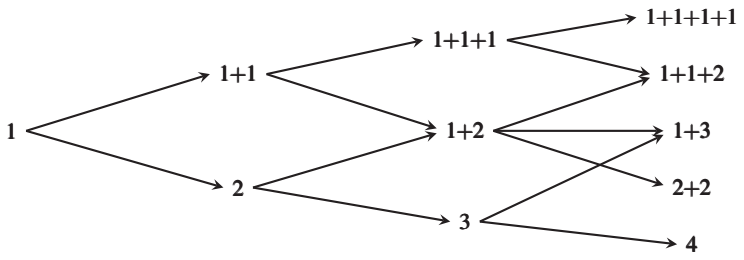


Figure 5 Evolution of the partitions from $\mathscr{P}(1)$, through $\mathscr{P}(2)$, $\mathscr{P}(3)$ to $\mathscr{P}(4)$

Evolution of $\mathscr{P}(n)$ ***for*** $n = 1,2,3,4$

Definition 2.6E Given partitions $p \in \mathscr{P}(n-1)$, $p' \in \mathscr{P}(n)$, p is the parent of p' and p' is a child of p if (i) either $p' = p \cup \{1\}$ or (ii) a set p_0, and an integer $m > 1$ exist such that

$$p = p_0 \cup \{m-1\},$$
$$p' = p_0 \cup \{m\}.$$

Given $p \in \mathscr{P}_{(n-1)}$, the expression evolve$(p)$ will denote the formal linear combination of the children of p, where the factor m is introduced in mp' if p' is a child of p in m different ways. Furthermore, evolve acting on a formal sum of members of $\mathscr{P}(n-1)$ will denote

$$\text{evolve}\Big(\sum_{i=1}^{m} C_i p_i\Big) = \sum_{i=1}^{m} C_i \,\text{evolve}(p_i).$$

For example,

$$\text{evolve}(1^3) = (1^4) + 3(1^2+2),$$
$$\text{evolve}(1+2) = (1^2+2) + (1+3) + (2^2),$$
$$\text{evolve}(3) = (1+3) + (4),$$

compared with evolve(P) for some corresponding sets, as read off from Figure 4,

$$\text{evolve}(1{+}2{+}3) = (1{+}2{+}3{+}4){+}(1{+}2{+}34){+}(1{+}3{+}24){+}(1{+}4{+}23),$$
$$\text{evolve}(1{+}23) = (1{+}234){+}(1{+}4{+}23){+}(14{+}23),$$
$$\text{evolve}(123) = (4{+}123){+}(1234).$$

Corresponding to Figure 4 the evolutionary scheme for partitions of numbers is shown in Figure 2.6, where $p \longrightarrow p'$ indicates that p' is a child of p.

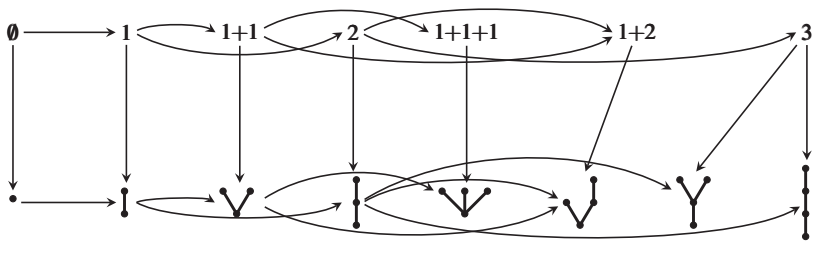


Figure 6 Evolution of the partitions from $\mathscr{P}(0)$, through $\mathscr{P}(1)$, $\mathscr{P}(2)$ to $\mathscr{P}(3)$ together with associated trees from $\mathbf{t}_1$ to $\mathbf{t}_8$

As partitions evolve, related trees evolve at the same time. This will be illustrated in Figure 6, showing how the evolution from $\mathscr{P}(0)$ to $\mathscr{P}(3)$ is accompanied by trees with orders 1 to 4.

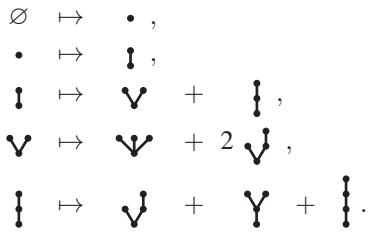
Evolution of labelled trees

We will look at the production of the $\alpha(\mathrm{t})$ copies of t as the result of an evolutionary process in which, in each step, one new vertex is added in every possible way to an existing tree. Denote each of the steps as the result of an operator evolve acting on an existing linear combination of trees. Examples of the action on low order trees are

$$\begin{aligned}
\mathrm{evolve}(\varnothing) &= \mathbf{t}_1,\\
\mathrm{evolve}(\mathbf{t}_1) &= \mathbf{t}_2,\\
\mathrm{evolve}(\mathbf{t}_2) &= \mathbf{t}_3 + \mathbf{t}_4,\\
\mathrm{evolve}(\mathbf{t}_3) &= \mathbf{t}_5 + 2\mathbf{t}_6,\\
\mathrm{evolve}(\mathbf{t}_4) &= \mathbf{t}_6 + \mathbf{t}_7 + \mathbf{t}_8,
\end{aligned}$$

where the action on $\varnothing$ is conventional.

These correspond to the diagrams



Write evolve^n as the composition of n applications of evolve and we have

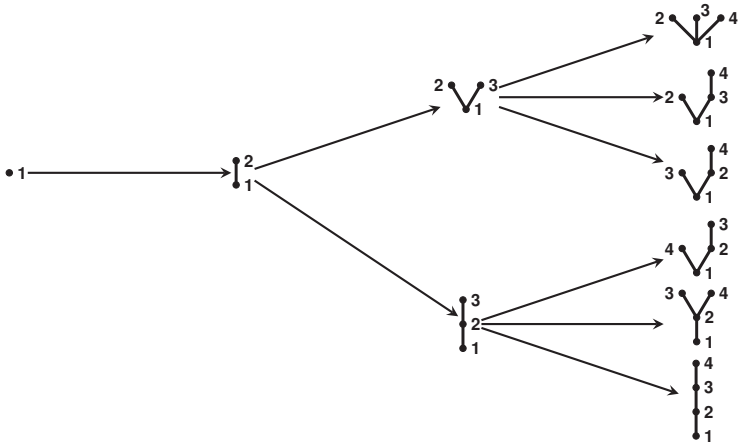


Figure 7 Evolution of labelled trees up to order 4

Theorem 2.6F

$$\text{evolve}^n(\varnothing) = \sum_{|\mathrm{t}|=n} \alpha(\mathrm{t})\mathrm{t}$$

Up to order 5 we have

$$\begin{aligned}
\text{evolve}(\varnothing) &= \mathbf{t}_1,\\
\text{evolve}^2(\varnothing) &= \mathbf{t}_2,\\
\text{evolve}^3(\varnothing) &= \mathbf{t}_3 + \mathbf{t}_4,\\
\text{evolve}^4(\varnothing) &= \mathbf{t}_5 + 3\mathbf{t}_6 + \mathbf{t}_7 + \mathbf{t}_8,\\
\text{evolve}^5(\varnothing) &= \mathbf{t}_9 + 6\mathbf{t}_{10} + 4\mathbf{t}_{11} + 4\mathbf{t}_{12} + 3\mathbf{t}_{13} + \mathbf{t}_{14} + 3\mathbf{t}_{15} + \mathbf{t}_{16} + \mathbf{t}_{17}.
\end{aligned}$$

It will now be shown diagrammatically how trees evolve by the addition of a further labelled vertex in each step. For order n, the labelled vertex n is attached to existing vertices in S_{n-1} in all possible ways. This gives the evolution diagram shown in Figure 7.

Strip off the labels and consolidate the identical unlabelled trees that remain. The result is shown in Figure 8 (p. 74).

Recursions for evolve

Theorem 2.6G evolve(t) satisfies the recursion

$$\text{evolve}(\mathrm{t}) = \begin{cases} \mathbf{t}_2, & \mathrm{t} = \mathbf{t}_1,\\ \text{evolve}(\mathrm{t}') * \mathrm{t}'' + \mathrm{t}' * \text{evolve}(\mathrm{t}''), & \mathrm{t} = \mathrm{t}' * \mathrm{t}''. \end{cases}$$

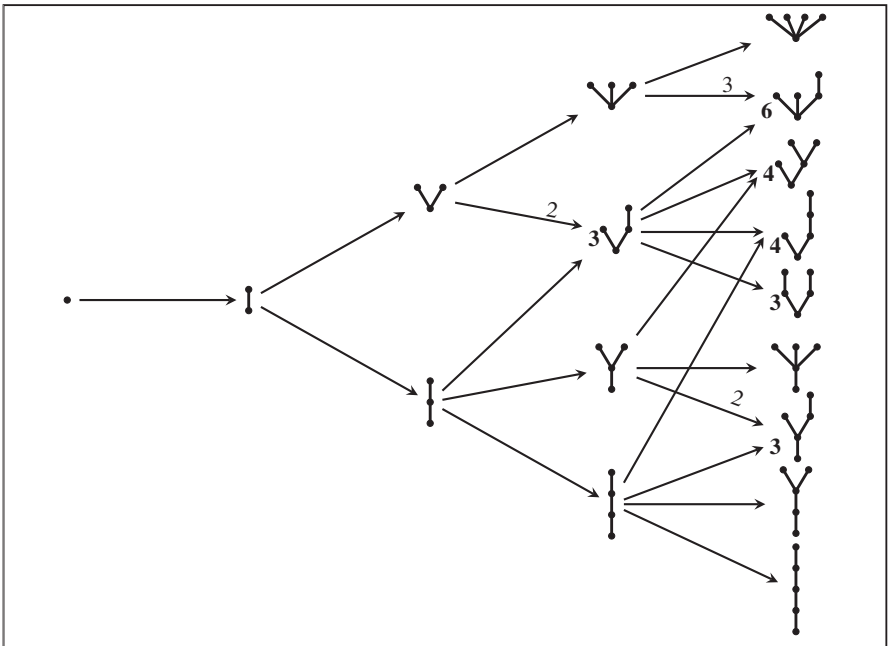


Figure 8 Evolution of trees up to order 5

Proof. The result for evolve(τ) is clear. It remains to show that evolve($t' * t''$) = evolve(t') $* t'' + t' *$ evolve(t''). The first term is the result of adding an additional child to each vertex in t' in turn and the second term is the result of adding the additional child to the vertices of t''. □

Theorem 2.6H evolve(t) satisfies the recursion

$$\text{evolve}(t) = \begin{cases} [\mathbf{t}_1], & t = \mathbf{t}_1, \\ t * \tau + \sum_{i=1}^{m} [t_1, \ldots, \text{evolve}(t_i), \ldots, t_m], & t = [t_1, t_2, \ldots, t_m]. \end{cases}$$

Proof. To prove the second option, note that an additional child is added in turn to the root, resulting in the term $t * \tau$ or, for each i, to one one of the vertices in t_i, resulting in the term $[t_1, \ldots, \text{evolve}(t_i), \ldots, t_m]$. □

The contributions of R. H. Merson

In 1957 a remarkable paper [72] appeared. This described the evolved combinations of trees, as illustrated here in Figure 8. As shown in Merson's work, and further discussed in Chapter 3 of the present volume, these expressions give the Taylor

series for the flow of a differential equation and, when compared with the series for a numerical approximation, enable the order conditions for Runge–Kutta method to be written down.

Labelling systems for forests

Using tuple***s***

Let tuple denote the set of all n-tuples of positive integers, for $n = 1,2,3,\dots$. That is

$$\text{tuple} = \big\{(1),(2),(3),\dots,(1,1),(1,2),\dots,(2,1),\dots,(1,1,1),(1,1,2),\dots\big\}.$$

If $x \in$ tuple is an n-tuple, denote by x^- the $(n-1)$-tuple formed from x by omitting the final integer from x. Conventionally, if $n = 1$, write $x^- = (\,) \notin$ tuple.

If (V,E) is connected then it is a labelled tree. The connected components combine together to comprise a forest. If $(V,E) = (\emptyset,\emptyset)$ then this is the empty tree $\varnothing$.

Definition 2.6I For $V \in$ tuple, a finite subset, define the graph (V,E) generated by V as the labelled graph for which

$$E = \big\{(x^-,x), x, x^- \in V\big\}.$$

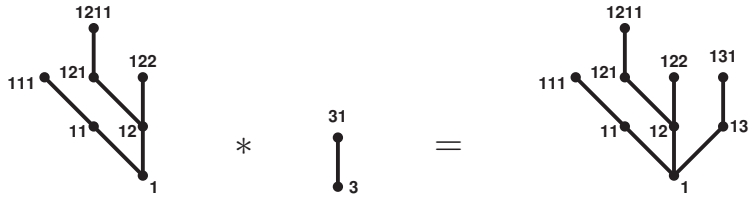
Lemma 2.6J Every forest corresponds to a graph generated by some finite subset of tuple.

Proof. It will be enough to restrict the result to trees and we prove this case by beta induction. For τ we can choose $V = \{(1)\}$. Assume the result has been proved for t and t$'$, and that these correspond to V and V', respectively. Without loss of generality, assume that the roots are (1) and (m), where m is chosen so that no member of V is of the form $(1,m)$. We note that every member of V' begins with the sequence $(m,\dots)$. Form V'' by prepending 1 to each member of V' so that the members of V'' are of the form $(1,m,\dots)$. The tree t $*$ t$'$ corresponds to $V \cup V''$. □

Convenient shortened form for tuples

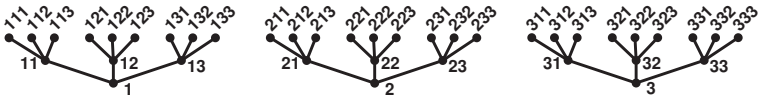
For convenience a tuple such as $(1,2,1,1)$ will sometimes be written as **1211**, if no confusion is possible.

We illustrate the formation of the beta product of two trees labelled using tuples, in a diagram:



The universal forest

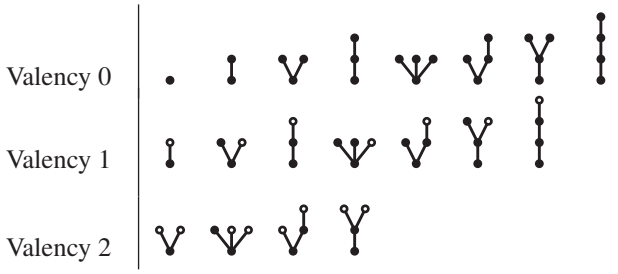
We can construct a connected graph with a denumerable set of vertices $V =$ tuple, rather than a finite subset. If we restrict the members of tuple to be based on $\{1,2,3\}$, rather than all the positive integers, we obtain a graph shown partially in this diagram.



2.7 Trees and stumps

We now consider a generalization of trees, known as "stumps". These can be regarded as modified trees with some leaves removed, but the edges from these leaves to their parents retained.

In the examples given here, a white disc represents the absence of a vertex. The number of white discs is the "valency". Note that trees are stumps with zero valency.

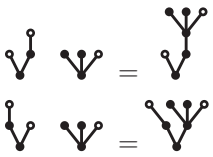


Right multiplication by one or more additional stumps implies grafting to open valency positions.

Products of stumps

Given two stumps s, s′, the product ss′ has a non-trivial product if s′ is not the trivial stump ∘ and s has valency at least 1; that is, it is not a tree, the product is formed by grafting the root of s′ to the rightmost open valency in s.

Two examples of grafting illustrate the significance of stump orientations



If the s_1 is a tree, no contraction takes place.

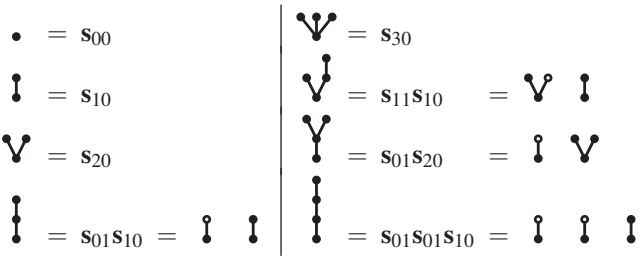
Atomic stumps

An atomic stump is a graph of the form

Note that no more than two generations can be present.

If m of the children of the root are represented by black discs and n are represented by white discs, then the atomic stump is denoted by $\mathbf{s}_{mn}$. The reason for the designation "atomic" is that every tree can be written as the product of atoms.

This is illustrated up to trees of order 4:

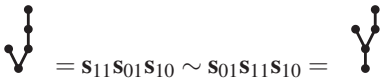


Isomeric trees

In the factorization of trees into products of atoms the factors are written in a specific order with each factor operating on later factors. However, if we interpret the atoms just as symbols that can commute with each other, we get a new equivalence relation:

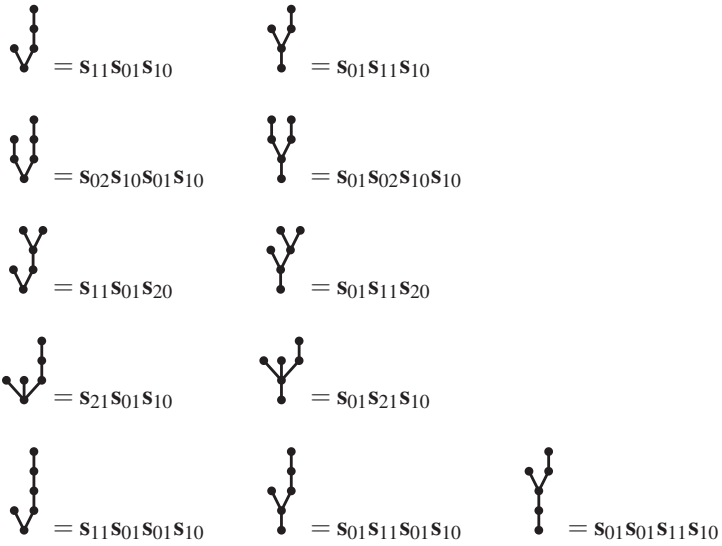
Definition 2.7A Two trees are isomeric if their atomic factors are the same.

Nothing interesting happens up to order 4 but, for order 5, we find that



It is a simple exercise to find all isomeries of any particular order but, as far as the author knows, this has not been done above order 6.

For orders 5 and 6, the isomers are



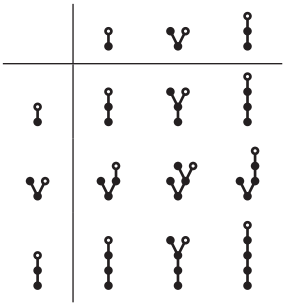
Exercise 26 Find two isomers of $\mathbf{s}_{21}\mathbf{s}_{01}\mathbf{s}_{01}\mathbf{s}_{10}$.

Isomers and Runge–Kutta order conditions

In Chapter 5, Section 5.2 (p. 183), structures related to stumps and isomeric trees will play a special role in distinguishing between the order conditions for Runge–Kutta methods intended for scalar problems, as distinct from more general high-dimensional systems.

The algebra of uni-valent stumps

Let S denote the semi-group under multiplication of stumps with valency 1. The multiplication table for this semi-group, restricted to stumps of orders 1 and 2, is shown on the left diagram below. Associativity is easy to verify.



	τ_1	$\tau_2\tau$	$\tau_1\tau_1$
τ_1	$\tau_1\tau_1$	$\tau_1\tau_2\tau$	$\tau_1\tau_1\tau_1$
$\tau_2\tau$	$\tau_2\tau\tau_1$	$\tau_2\tau\tau_2\tau$	$\tau_2\tau\tau_1\tau_1$
$\tau_1\tau_1$	$\tau_1\tau_1\tau_1$	$\tau_1\tau_1\tau_2\tau$	$\tau_1\tau_1\tau_1\tau_1$

Exercise 27 Extend the right table to include stumps to order 3.

Sums over S

By taking free sums of members of S, we can construct an algebra of linear operators. Because of the important applications to the analysis of Rosenbrock and related methods, and to applications to exponential integrators, we will focus on members of this algebra of the form

$$a_0 1 + a_1\tau_1 + \cdots + a_n\tau_1^2 + \cdots . \tag{2.7 a}$$

Let $\phi(z) = a_0 + a_1 z + a_2 z^2 + \cdots$, then we will conventionally write (2.7 a) as $\phi(\tau_1)$.

Important examples in applications are

$$\phi(z) = (1-\gamma z)^{-1},$$
$$\phi(z) = \frac{1+\frac{1}{2}z}{1-\frac{1}{2}z},$$
$$\phi(z) = \frac{\exp(z)-1}{z}.$$

Taking this further, we can give a meaning to expressions such as

$$\phi(\tau_1)\mathrm{t} = a_0\mathrm{t} + a_1[\mathrm{t}] + \cdots + a_n[{}_n\mathrm{t}]_n + \cdots \tag{2.7 b}$$

This topic will be considered again, in the context of B-series and numerical applications, in Chapter 4, Section 4.8 (p. 174).

Sub-atomic stumps

The special stumps $\mathbf{s}_{0,n} = \tau_n$, with order 1, have a special role in Chapter 5, Section 5.2 (p. 183).

It is possible to write these in terms of an incomplete use of the beta product. That is, t$*$ is a stump in which an additional valency is assigned to the root of t. This would mean that

$$(\mathrm{t}*)\mathrm{t}' = \mathrm{t} * \mathrm{t}'.$$

Write $*^n$ for an iterated use of this notation and we have

$$\tau *^n = \mathbf{s}_{0,n} = \tau_n. \tag{2.7 c}$$

Definition 2.7B The objects occurring in (2.7 c) are sub-atomic stumps.

Note the identity

$$\mathbf{s}_{mn} = \tau *^{m+n} \tau^m,$$

which allows any tree written as a product of atomic stumps to be further factorized into sub-atomic stumps.

2.8 Subtrees, supertrees and prunings

A subtree, of a given tree, is a second tree embedded within the given tree, and sharing the same root. The vertices, and their connecting edges, cut off from the original tree to expose the subtree, comprise the offcut. Since the offcut is a collection of trees, they will be interpreted as a forest. If t′ is a subtree of t, then t is said to be a "supertree" of t′. The words "pruning" and "offcut" will be used interchangeably.

For the two given trees,

$\mathrm{t}' =$, $\quad \mathrm{t} =$  , (2.8 a)

t′ is a subtree of t, and t is a supertree of t′, because t′ can be constructed by deleting some of the vertices and edges from t or, alternatively, t can be constructed from t′ by adding additional vertices and edges.

The member of the forest space, written as t ⟍ t′, is formed from the collection of possible ways t can be "pruned" to form t′. For example, for the trees given by (2.8 a),

$$\mathrm{t} \setminus \mathrm{t}' = 2 \bullet$$  ,

where the factor 2 is a consequence of the fact that t could equally well have been replaced by its mirror image.

These simple ideas can lead to ambiguities and the possibility of duplications. For example, the tree $\mathrm{t} = [[\tau][\tau^2]]$, regarded as a labelled tree, can have the offcut removed in three ways to reveal the subtree $\mathrm{t}' = [_2\tau]_2$. This can be illustrated by using the orientations of the edges of t' to distinguish them, as shown in the diagram

cut =

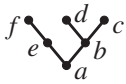
case	t showing cuts	t′ showing orientation	offcut
(i)			
(ii)			
(iii)			

In this diagram, cases (i) and (ii) give the same t' and the same prunings. Hence, a complete list of these for a given t would need to contain a duplicate of this case. Further, case (iii) needs to be included separately even though they both have the same subtree t'. We can deal with cases like this by reverting to the use of labelled trees. That is, we can use the (V,E,r) characterization of trees.

For the present example, we would identify t with the triple

$$\mathrm{t} = (V,E,r) := (\{a,b,c,d,e,f\},\{(a,b),(b,c),(b,d),(a,e),(e,f)\},a),$$

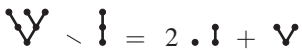
as in the diagram



and the three cases of $\mathrm{t}' = (V_1,E_1,r)$ and $\mathrm{f} = (V_2,E_2)$ are

(i) $(V_1,E_1,r) = (\{a,b,c\},\{(a,b),(b,c)\},a)$, $(V_2,E_2) = (\{d,e,f\},\{(e,f)\})$,
(ii) $(V_1,E_1,r) = (\{a,b,d\},\{(a,b),(b,d)\},a)$, $(V_2,E_2) = (\{c,e,f\},\{(e,f)\})$,
(iii) $(V_1,E_1,r) = (\{a,e,f\},\{(a,e),(e,f)\},a)$, $(V_2,E_2) = (\{b,c,d\},\{(b,c),(b,d)\})$.

In specifying a subgraph (V_1,E_1) of (V,E), it will be enough to specify only V_1 because the vertex pairs which comprise E_1 are the same as for E except for the deletion of any pair containing a member of $V \setminus V_1$; hence the members of E_1 do not need to be stated. In this example, we can write



Main definition

Definition 2.8A Consider a labelled tree $\mathrm{t} = (V, E, r)$. A subgraph defined by V_1, with the property that parent$(v) \in V_1$ whenever $v \in V_1$, is a subtree t' of t. If $V_1 \subset V$ then t' is a proper subtree. The corresponding offcut is the forest formed from $V \setminus V_1$.

In using subtrees in later discussions, we will not give explicit labels for the elements of V, but we will take the existence of labels into account.

Notations

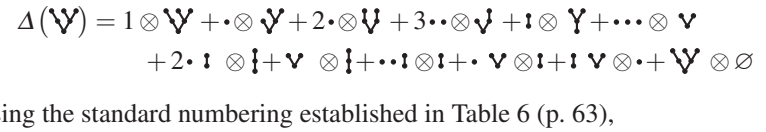
Notationally, $\mathrm{t}' \leq \mathrm{t}$ will denote "t' is a subtree of t" and $\mathrm{t}' < \mathrm{t}$ will denote "t' is a proper subtree of t"; Also $\varnothing < \mathrm{t}' \leq \mathrm{t}$ and $\varnothing < \mathrm{t}' < \mathrm{t}$ will have their obvious meanings. The offcuts corresponding to $V \setminus V_1$ will be written $\mathrm{t} \setminus \mathrm{t}'$.

Using the Sweedler notation [89] (Underwood, 2011), adapted from co-algebra theory, a "pruning" consisting of an offcut-subtree pair will be written $(\mathrm{t} \setminus \mathrm{t}') \otimes \mathrm{t}'$.

Definition 2.8B The function $\Delta(\mathrm{t})$ denotes the combination, with integer weights to indicate repetitions, of all prunings of t. That is,

$$\Delta(\mathrm{t}) = \mathrm{t} \otimes \varnothing + \sum_{\mathrm{t}' \leq \mathrm{t}} (\mathrm{t} \setminus \mathrm{t}') \otimes \mathrm{t}'.$$

For example,



or, using the standard numbering established in Table 6 (p. 63),

$$\begin{aligned}\Delta(\mathbf{t}_{27}) = {} & 1 \otimes \mathbf{t}_{27} + \mathbf{t}_1 \otimes \mathbf{t}_{11} + 2\mathbf{t}_1 \otimes \mathbf{t}_{13} + 3\mathbf{t}_1^2 \otimes \mathbf{t}_6 + \mathbf{t}_2 \otimes \mathbf{t}_7 + \mathbf{t}_1^3 \otimes \mathbf{t}_3 \\ & + 2\mathbf{t}_1\mathbf{t}_2 \otimes \mathbf{t}_4 + \mathbf{t}_3 \otimes \mathbf{t}_4 + \mathbf{t}_1^2\mathbf{t}_2 \otimes \mathbf{t}_2 + \mathbf{t}_1\mathbf{t}_3 \otimes \mathbf{t}_2 + \mathbf{t}_2\mathbf{t}_3 \otimes \mathbf{t}_1 + \mathbf{t}_{27} \otimes \varnothing.\end{aligned}$$

For later reference, $\Delta(\mathrm{t})$, for $|\mathrm{t}| \leq 5$, is given in Table 9.

Subtrees in Polish notation

If $\varnothing < \mathrm{t}' \leq \mathrm{t}$, write

$$\mathrm{t}' = \tau_{m_1}\tau_{m_2}\cdots\tau_{m_k}, \quad k = |\mathrm{t}'|.$$

then we can write a representation of t using the operator $\circledast$, introduced in Section 2.2 (p. 47). This result is given without proof.

Table 9 $\Delta(\mathrm{t})$, for $|\mathrm{t}| \le 5$

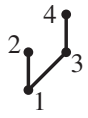
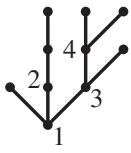
$\lvert\mathrm{t}\rvert$	t	$\Delta(\mathrm{t})$
0	$\varnothing$	$1\otimes\varnothing$
1	$\mathbf{t}_1$	$1\otimes\mathbf{t}_1+\mathbf{t}_1\otimes\varnothing$
2	$\mathbf{t}_2$	$1\otimes\mathbf{t}_2+\mathbf{t}_1\otimes\mathbf{t}_1+\mathbf{t}_2\otimes\varnothing$
3	$\mathbf{t}_3$	$1\otimes\mathbf{t}_3+2\mathbf{t}_1\otimes\mathbf{t}_2+\mathbf{t}_1^2\otimes\mathbf{t}_1+\mathbf{t}_3\otimes\varnothing$
3	$\mathbf{t}_4$	$1\otimes\mathbf{t}_4+\mathbf{t}_1\otimes\mathbf{t}_2+\mathbf{t}_2\otimes\mathbf{t}_1+\mathbf{t}_4\otimes\varnothing$
4	$\mathbf{t}_5$	$1\otimes\mathbf{t}_5+3\mathbf{t}_1\otimes\mathbf{t}_3+3\mathbf{t}_1^2\otimes\mathbf{t}_2+\mathbf{t}_1^3\otimes\mathbf{t}_1+\mathbf{t}_5\otimes\varnothing$
4	$\mathbf{t}_6$	$1\otimes\mathbf{t}_6+\mathbf{t}_1\otimes\mathbf{t}_3+\mathbf{t}_1\otimes\mathbf{t}_4+\mathbf{t}_1^2\otimes\mathbf{t}_2+\mathbf{t}_2\otimes\mathbf{t}_2+\mathbf{t}_1\mathbf{t}_2\otimes\mathbf{t}_1+\mathbf{t}_6\otimes\varnothing$
4	$\mathbf{t}_7$	$1\otimes\mathbf{t}_7+2\mathbf{t}_1\otimes\mathbf{t}_4+\mathbf{t}_1^2\otimes\mathbf{t}_2+\mathbf{t}_3\otimes\mathbf{t}_1+\mathbf{t}_7\otimes\varnothing$
4	$\mathbf{t}_8$	$1\otimes\mathbf{t}_8+\mathbf{t}_1\otimes\mathbf{t}_4+\mathbf{t}_2\otimes\mathbf{t}_2+\mathbf{t}_4\otimes\mathbf{t}_1+\mathbf{t}_8\otimes\varnothing$
5	$\mathbf{t}_9$	$1\otimes\mathbf{t}_9+4\mathbf{t}_1\otimes\mathbf{t}_5+6\mathbf{t}_1^2\otimes\mathbf{t}_3+4\mathbf{t}_1^3\otimes\mathbf{t}_2+\mathbf{t}_1^4\otimes\mathbf{t}_1+\mathbf{t}_9\otimes\varnothing$
5	$\mathbf{t}_{10}$	$1\otimes\mathbf{t}_{10}+2\mathbf{t}_1\otimes\mathbf{t}_6+\mathbf{t}_1\otimes\mathbf{t}_5+2\mathbf{t}_1^2\otimes\mathbf{t}_3+\mathbf{t}_1^2\otimes\mathbf{t}_4+\mathbf{t}_2\otimes\mathbf{t}_3+\mathbf{t}_1^3\otimes\mathbf{t}_2+2\mathbf{t}_1\mathbf{t}_2\otimes\mathbf{t}_2+\mathbf{t}_1^2\mathbf{t}_2\otimes\mathbf{t}_1+\mathbf{t}_{10}\otimes\varnothing$
5	$\mathbf{t}_{11}$	$1\otimes\mathbf{t}_{11}+2\mathbf{t}_1\otimes\mathbf{t}_6+\mathbf{t}_1\otimes\mathbf{t}_7+\mathbf{t}_1^2\otimes\mathbf{t}_3+2\mathbf{t}_1^2\otimes\mathbf{t}_4+\mathbf{t}_1^3\otimes\mathbf{t}_2+\mathbf{t}_3\otimes\mathbf{t}_2+\mathbf{t}_1\mathbf{t}_3\otimes\mathbf{t}_1+\mathbf{t}_{11}\otimes\varnothing$
5	$\mathbf{t}_{12}$	$1\otimes\mathbf{t}_{12}+\mathbf{t}_1\otimes\mathbf{t}_6+\mathbf{t}_1\otimes\mathbf{t}_8+\mathbf{t}_1^2\otimes\mathbf{t}_4+\mathbf{t}_2\otimes\mathbf{t}_3+\mathbf{t}_1\mathbf{t}_2\otimes\mathbf{t}_2+\mathbf{t}_4\otimes\mathbf{t}_2+\mathbf{t}_1\mathbf{t}_4\otimes\mathbf{t}_1+\mathbf{t}_{12}\otimes\varnothing$
5	$\mathbf{t}_{13}$	$1\otimes\mathbf{t}_{13}+2\mathbf{t}_1\otimes\mathbf{t}_6+\mathbf{t}_1^2\otimes\mathbf{t}_3+2\mathbf{t}_2\otimes\mathbf{t}_4+2\mathbf{t}_1\mathbf{t}_2\otimes\mathbf{t}_2+\mathbf{t}_2^2\otimes\mathbf{t}_1+\mathbf{t}_{13}\otimes\varnothing$
5	$\mathbf{t}_{14}$	$1\otimes\mathbf{t}_{14}+3\mathbf{t}_1\otimes\mathbf{t}_7+3\mathbf{t}_1^2\otimes\mathbf{t}_4+\mathbf{t}_1^3\otimes\mathbf{t}_2+\mathbf{t}_5\otimes\mathbf{t}_1+\mathbf{t}_{14}\otimes\varnothing$
5	$\mathbf{t}_{15}$	$1\otimes\mathbf{t}_{15}+\mathbf{t}_1\otimes\mathbf{t}_7+\mathbf{t}_1\otimes\mathbf{t}_8+\mathbf{t}_1^2\otimes\mathbf{t}_4+\mathbf{t}_2\otimes\mathbf{t}_4+\mathbf{t}_1\mathbf{t}_2\otimes\mathbf{t}_2+\mathbf{t}_6\otimes\mathbf{t}_1+\mathbf{t}_{15}\otimes\varnothing$
5	$\mathbf{t}_{16}$	$1\otimes\mathbf{t}_{16}+2\mathbf{t}_1\otimes\mathbf{t}_8+\mathbf{t}_1^2\otimes\mathbf{t}_4+\mathbf{t}_3\otimes\mathbf{t}_2+\mathbf{t}_7\otimes\mathbf{t}_1+\mathbf{t}_{16}\otimes\varnothing$
5	$\mathbf{t}_{17}$	$1\otimes\mathbf{t}_{17}+\mathbf{t}_1\otimes\mathbf{t}_8+\mathbf{t}_2\otimes\mathbf{t}_4+\mathbf{t}_4\otimes\mathbf{t}_2+\mathbf{t}_8\otimes\mathbf{t}_1+\mathbf{t}_{17}\otimes\varnothing$

Theorem 2.8C Any tree $\mathrm{t}\ge\mathrm{t}'$ can be written in the form

$$\mathrm{t}=(\tau_{m_1}\circledast\mathrm{f}_1)(\tau_{m_2}\circledast\mathrm{f}_2)\cdots(\tau_{m_k}\circledast\mathrm{f}_n),$$

where $\mathrm{f}_1,\mathrm{f}_2,\ldots,\mathrm{f}_k\in\mathrm{F}$.

For example, consider

$\mathrm{t}'=$  $\le$  $=\mathrm{t}.$ (2.8 b)

We have

$$\mathrm{t}=(\tau_2*\tau)(\tau*\tau_1\tau)(\tau_1*\tau)(\tau*\tau^2),$$

$$\mathrm{t}'=\tau_2\tau\tau_1\tau, \tag{2.8 c}$$

$$\mathrm{f}_1\mathrm{f}_2\mathrm{f}_3\mathrm{f}_4=\tau^4\tau_1\tau=\bullet\bullet\bullet\bullet\!\!\mid. \tag{2.8 d}$$

Note that in (2.8 b), the labelled vertices in t' and t, correspond to the the four factors τ_2, τ, τ_1, τ in (2.8 c). To convert t' to t, additional trees are attached as follows:
to label 1: an additional tree τ,
to label 2: an additional tree $\tau_1\tau$,
to label 3: an additional tree τ,
to label 4: two additional trees, each of them τ.
Combining these additional trees together, we obtain (2.8 d) as the forest

$$\mathrm{t} \smallsetminus \mathrm{t}' = (\tau)(\tau_1\tau)(\tau^2)(\tau) = \tau\tau_1\tau\tau^2\tau = \tau^4\tau_1\tau.$$

Starting with the example case we have already considered, the possible choices of f_1, f_2, f_3, f_4, to convert t' to t, are

$$\begin{array}{llll}
\mathrm{f}_1 = \bullet, & \mathrm{f}_2 = \bullet\!\!\bullet, & \mathrm{f}_3 = \bullet, & \mathrm{f}_4 = \bullet\bullet, \\
\mathrm{f}_1 = \bullet, & \mathrm{f}_2 = \bullet\!\!\bullet, & \mathrm{f}_3 = \bullet\!\curlyvee, & \mathrm{f}_4 = 1, \\
\mathrm{f}_1 = \bullet, & \mathrm{f}_2 = \bullet\curlyvee, & \mathrm{f}_3 = 1, & \mathrm{f}_4 = \bullet, \\
\mathrm{f}_1 = \curlyvee, & \mathrm{f}_2 = 1, & \mathrm{f}_3 = 1, & \mathrm{f}_4 = \bullet, \\
\mathrm{f}_1 = \bullet\!\!\bullet, & \mathrm{f}_2 = 1, & \mathrm{f}_3 = \bullet, & \mathrm{f}_4 = \bullet\bullet, \\
\mathrm{f}_1 = \bullet\!\!\bullet, & \mathrm{f}_2 = 1, & \mathrm{f}_3 = \bullet\!\curlyvee, & \mathrm{f}_4 = 1.
\end{array}$$

Using the numbered tree notation, we have

$$\mathrm{t} \smallsetminus \mathrm{t}' = \mathbf{t}_1^4\mathbf{t}_2 + \mathbf{t}_1\mathbf{t}_2\mathbf{t}_3 + \mathbf{t}_1^3\mathbf{t}_3 + \mathbf{t}_1\mathbf{t}_{11} + \mathbf{t}_1\mathbf{t}_2\mathbf{t}_4 + \mathbf{t}_3\mathbf{t}_4.$$

Subtrees, supertrees and prunings have their principal applications in Chapter 3,Section 3.9 (p. 133).

Recursions

The two tree-building mechanisms that have been introduced:

$$\mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_n \mapsto [\mathrm{t}_1\mathrm{t}_2\cdots\mathrm{t}_n],$$
$$\mathrm{t}_1, \mathrm{t}_2 \mapsto \mathrm{t}_1 * \mathrm{t}_2,$$

have been used to define and evaluate a number of functions recursively. We will now apply this approach to Δ.

Recursion for Δ using the beta-product

For a given tree t, $\Delta(\mathrm{t})$ is a combination of terms of the form $\mathrm{f} \otimes \mathrm{t}'$ with the proviso that for the term for which $\mathrm{t}' = \varnothing$, $\mathrm{f} = \mathrm{t}$. We need to extend the meaning of the beta-product to the individual terms of these types and extend the meaning further by treating the product as a bi-linear operator.

Theorem 2.8D Given trees t_1, t_2,

$$\Delta(t_1 * t_2) = \Delta(t_1) * \Delta(t_2),$$

where the product on the right is treated bi-linearly and specific term by term products are evaluated according to the rules

$$\begin{aligned}
&(f_1' \otimes t_1') * (f_2' \otimes t_2') = (f_1' f_2') \otimes (t_1' * t_2'),\\
&(t_1' \otimes \varnothing) * (f_2' \otimes t_2') = 0,\\
&(f_1' \otimes t_1') * (t_2' \otimes \varnothing) = (f_1' t_2') \otimes t_1',\\
&(t_1' \otimes \varnothing) * (t_2' \otimes \varnothing) = (t_1' * t_2') \otimes \varnothing.
\end{aligned}$$

Proof. If $t = t_1 \otimes t_2$, then the prunings of t consist of terms of the form

$$(t_1 \smallsetminus t_1')(t_2 \smallsetminus t_2') \otimes (t_1' * t_2'),$$

corresponding to $t_1' \leq t_1$, $t_2' \leq t_2$, where $t_1' > \varnothing$, together with $(t_1 * t_2) \otimes \varnothing$, corresponding to $t = \varnothing$. Now calculate

$$\Delta(t_1 * t_2) = \sum_{\varnothing < t_1' \leq t_1} \sum_{t_2' \leq t_2} (t_1 \smallsetminus t_1')(t_2 \smallsetminus t_2') \otimes (t_1' * t_2') + (t_1 * t_2) \otimes \varnothing$$

and compare with

$$\begin{aligned}
\Delta(t_1) * \Delta(t_2) &= \Big(\sum_{\varnothing < t_1' \leq t_1} (t_1 \smallsetminus t_1') \otimes t_1 + t_1 \otimes \varnothing \Big) \otimes \Big(\sum_{t_2' \leq t_1} (t_2 \smallsetminus t_2') \otimes t_2 \Big)\\
&= \sum_{\varnothing < t_1' \leq t_1} \sum_{t_2' \leq t_2} (t_1 \smallsetminus t_1')(t_2 \smallsetminus t_2') \otimes (t_1' * t_2') + (t_1 * t_2) \otimes \varnothing\\
&= \Delta(t_1 * t_2).
\end{aligned}$$

□

Examples of the beta-product recursion for Δ

To see how this works we will verify the formula

$$\Delta(\mathbf{t}_1) * \Delta(\mathbf{t}_2) = \Delta(\mathbf{t}_4).$$

We have

$$\begin{aligned}
\Delta(\mathbf{t}_1) * \Delta(\mathbf{t}_2) &= \big(1 \otimes \mathbf{t}_1 + \mathbf{t}_1 \otimes \varnothing\big) * \big(1 \otimes \mathbf{t}_2 + \mathbf{t}_1 \otimes \mathbf{t}_1 + \mathbf{t}_2 \otimes \varnothing\big)\\
&= (1 \otimes \mathbf{t}_1) * (1 \otimes \mathbf{t}_2) + (1 \otimes \mathbf{t}_1) * (\mathbf{t}_1 \otimes \mathbf{t}_1) + (1 \otimes \mathbf{t}_1) * (\mathbf{t}_2 \otimes \varnothing)\\
&\quad + (\mathbf{t}_1 \otimes \varnothing) * (1 \otimes \mathbf{t}_2) + (\mathbf{t}_1 \otimes \varnothing) * (\mathbf{t}_1 \otimes \mathbf{t}_1) + (\mathbf{t}_1 \otimes \varnothing) * (\mathbf{t}_2 \otimes \varnothing)\\
&= 1 \otimes (\mathbf{t}_1 * \mathbf{t}_2) + \mathbf{t}_1 \otimes (\mathbf{t}_1 * \mathbf{t}_1) + \mathbf{t}_2 \otimes (\mathbf{t}_1 * \varnothing) + (0) + (0) + (\mathbf{t}_1 * \mathbf{t}_2) \otimes \varnothing\\
&= 1 \otimes \mathbf{t}_4 + \mathbf{t}_1 \otimes \mathbf{t}_2 + \mathbf{t}_2 \otimes \mathbf{t}_1 + \mathbf{t}_4 \otimes \varnothing\\
&= \Delta(\mathbf{t}_4).
\end{aligned}$$

As a second example, we will verify the formula

$$\Delta(\mathbf{t}_2) * \Delta(\mathbf{t}_1) = \Delta(\mathbf{t}_3).$$

We have

$$\begin{aligned}
\Delta(\mathbf{t}_2)*\Delta(\mathbf{t}_1) &= \big(1\otimes\mathbf{t}_2+\mathbf{t}_1\otimes\mathbf{t}_1+\mathbf{t}_2\otimes\varnothing\big)*\big(1\otimes\mathbf{t}_1+\mathbf{t}_1\otimes\varnothing\big)\\
&= (1\otimes\mathbf{t}_2)*(1\otimes\mathbf{t}_1)+(1\otimes\mathbf{t}_2)*(\mathbf{t}_1\otimes\varnothing)+(\mathbf{t}_1\otimes\mathbf{t}_1)*(1\otimes\mathbf{t}_1)\\
&\quad +(\mathbf{t}_1\otimes\mathbf{t}_1)*(\mathbf{t}_1\otimes\varnothing)+(\mathbf{t}_2\otimes\varnothing)*(1\otimes\mathbf{t}_1)+(\mathbf{t}_2\otimes\varnothing)*(\mathbf{t}_1\otimes\varnothing)\\
&= 1\otimes(\mathbf{t}_2*\mathbf{t}_1)+\mathbf{t}_1\otimes(\mathbf{t}_2*\varnothing)+\mathbf{t}_1\otimes(\mathbf{t}_1*\mathbf{t}_1)+\mathbf{t}_1^2\otimes(\mathbf{t}_1*\varnothing)+0+(\mathbf{t}_2*\mathbf{t}_1)\otimes\varnothing\\
&= 1\otimes\mathbf{t}_3+2\mathbf{t}_1\otimes\mathbf{t}_2+\mathbf{t}_1^2\otimes\mathbf{t}_1+\mathbf{t}_3\otimes\varnothing\\
&= \Delta(\mathbf{t}_3).
\end{aligned}$$

Recursion for Δ using the B^+ operation

Theorem 2.8E Let $\mathrm{t} = [\mathrm{t}_1\mathrm{t}_2\cdots\mathrm{t}_m]$, then

$$\Delta(\mathrm{t}) = \Big[\prod_{i=1}^{m}\Delta(\mathrm{t}_i)\Big] + \mathrm{t}\otimes\varnothing,$$

where the expansion of the product $\prod_{i=1}^{m}$ and the term-by-term evaluation of $[\cdot]$ are according to the rules

$$\begin{aligned}
(\mathrm{f}_1\otimes\mathrm{f}_2)(\mathrm{f}_3\otimes\mathrm{f}_4) &= (\mathrm{f}_1\mathrm{f}_3)\otimes(\mathrm{t}_2*\mathrm{t}_4),\\
(\mathrm{t}_1\otimes\varnothing)(\mathrm{f}_2\otimes\mathrm{f}_3) &= (\mathrm{f}_1\mathrm{f}_2)\otimes\mathrm{f}_3,\\
(\mathrm{f}_2\otimes\mathrm{f}_3)(\mathrm{t}_1\otimes\varnothing) &= (\mathrm{f}_1\mathrm{f}_2)\otimes\mathrm{f}_3,\\
(\mathrm{f}_1\otimes\varnothing)(\mathrm{f}_2\otimes\varnothing) &= (\mathrm{f}_1\mathrm{f}_2)\otimes\varnothing,\\
[(\mathrm{f}_1\otimes\mathrm{f}_2)] &= \mathrm{f}_1[\mathrm{f}_2],\\
[(\mathrm{f}_1\otimes\varnothing)] &= \mathrm{f}_1.
\end{aligned}$$

Proof. The proof is by induction on the integer m. We will verify the result for the two cases

(i) $\mathrm{t} = [\mathrm{t}_1]$,
(ii) $\mathrm{t} = [(\mathrm{t}_1\mathrm{t}_2\cdots\mathrm{t}_{m-1})(\mathrm{t}_m)]$.

For case (i), we have, using Theorem 2.8D,

$$\begin{aligned}
\Delta([\mathrm{t}_1]) &= \Delta(\tau*\mathrm{t}_1)\\
&= \Delta(\tau)*\Delta(\mathrm{t}_1)\\
&= (1\otimes\tau+\tau\otimes\varnothing)*\Delta(\mathrm{t}_1)\\
&= [\Delta(\mathrm{t}_1)]+[\mathrm{t}_1]\otimes\varnothing.
\end{aligned}$$

For case (ii),

$$\begin{aligned}\Delta(\mathrm{t}) &= \Big(\Delta(\tau) * \Delta\Big(\prod_{i=1}^{m-1}(\mathrm{t}_i)\Big)\Big) * \Delta\mathrm{t}_m \\ &= \Big(\prod_{i=1}^{m-1}\Delta([\mathrm{t}_i]) + ([\mathrm{t}_1\mathrm{t}_2\cdots\mathrm{t}_{m-1}]\otimes\varnothing\Big) * \Delta(\mathrm{t}_m) \\ &= \prod_{i=1}^{m}\Delta([\mathrm{t}_i]) + [\mathrm{t}_1\mathrm{t}_2\cdots\mathrm{t}_m]\otimes\varnothing. \qquad \square\end{aligned}$$

Two examples

The first example is to evaluate $\Delta(\mathbf{t}_4) = \Delta([\mathbf{t}_2])$ and we see that $m = 1$. We have

$$\sum_{j=1}^{3}\widetilde{\mathrm{f}}_j\otimes\widehat{\mathrm{f}}_j = 1\otimes\mathbf{t}_2 + \mathbf{t}_1\otimes\mathbf{t}_1 + \mathbf{t}_2\otimes\varnothing,$$

and in accordance with Theorem 2.8E,

$$\Delta(\mathbf{t}_4) = 1\otimes\mathbf{t}_4 + \mathbf{t}_1\otimes\mathbf{t}_2 + \mathbf{t}_2\otimes\mathbf{t}_1 + \mathbf{t}_4\otimes\varnothing.$$

The second example is to evaluate $\Delta(\mathbf{t}_3) = \Delta([\mathbf{t}_1^2])$ and we have

$$\sum_{j=1}^{3}\widetilde{\mathrm{f}}_j\otimes\widehat{\mathrm{f}}_j = (1\otimes\mathbf{t}_1 + \mathbf{t}_1\otimes\varnothing)^2 = 1\otimes\mathbf{t}_1^2 + 2\mathbf{t}_1\otimes\mathbf{t}_1 + \mathbf{t}_1^2\otimes\varnothing,$$

leading to

$$\Delta(\mathbf{t}_3) = 1\otimes\mathbf{t}_3 + 2\mathbf{t}_1\otimes\mathbf{t}_2 + \mathbf{t}_1^2\otimes\mathbf{t}_1 + \mathbf{t}_3\otimes\varnothing.$$

Exercise 28 Let $\mathrm{t}_n = [\tau^n]$, $n = 0, 1, 2, \ldots$ Show that

$$\Delta(\mathrm{t}_n) = \sum_{i=0}^{n}\binom{n}{i}\tau^{n-i}\otimes\mathrm{t}_i + \mathrm{t}_n\otimes\varnothing,$$

where, conventionally, $\tau^0 = 1$.

Exercise 29 Let $\mathrm{t}_0 = \tau$, $\mathrm{t}_n = [{}_n\tau]_n$, $n = 1, 2, \ldots$ Find the value of $\Delta(\mathrm{t}_n)$.

Reversing the roles of subtrees and supertrees

The relation $\mathrm{t}' \le \mathrm{t}$ has been introduced as defining the subtrees of t but it can also be used to define the supertrees of t'. Looked at this way, $\mathrm{t} \smallsetminus \mathrm{t}'$ becomes the forest of trees that need to be grafted to t' to construct t.

Corresponding to the term $(\mathrm{t}\backslash\mathrm{t}')\otimes\mathrm{t}'$ in $\Delta(\mathrm{t})$, we will use $\mathrm{t}\widetilde{\otimes}(\mathrm{t}\backslash\mathrm{t}')$ as a term in $\widetilde{\Delta}(\mathrm{t}')$ to represent this same relationship. However, this needs to be modified by inserting scale factors related to the symmetry of the various trees.

Definition 2.8F

$$\widetilde{\Delta}(\mathrm{t}')=\sum_{\mathrm{t}\geq\mathrm{t}'}\frac{\sigma(\mathrm{t}')}{\sigma(\mathrm{t})}\mathrm{t}\widetilde{\otimes}(\mathrm{t}\backslash\mathrm{t}').$$

Because $\widetilde{\Delta}(\mathrm{t}')$ is an infinite series, it will not be possible to give complete information in the form of examples. However, four cases will be given and these will be expanded as far as terms with $|\mathrm{t}|\leq 5$.

$$\begin{aligned}
\widetilde{\Delta}(\mathbf{t}_1)&=\mathbf{t}_1\widetilde{\otimes}1+\mathbf{t}_2\widetilde{\otimes}\mathbf{t}_1+\tfrac{1}{2}\mathbf{t}_3\widetilde{\otimes}\mathbf{t}_1^2+\mathbf{t}_4\widetilde{\otimes}\mathbf{t}_2+\tfrac{1}{6}\mathbf{t}_5\widetilde{\otimes}\mathbf{t}_1^3+\mathbf{t}_6\widetilde{\otimes}\mathbf{t}_1\mathbf{t}_2+\mathbf{t}_7\widetilde{\otimes}\mathbf{t}_3+\mathbf{t}_8\widetilde{\otimes}\mathbf{t}_4\\
&\quad+\tfrac{1}{24}\mathbf{t}_9\widetilde{\otimes}\mathbf{t}_1^4+\tfrac{1}{2}\mathbf{t}_{10}\widetilde{\otimes}\mathbf{t}_1^2\mathbf{t}_2+\tfrac{1}{2}\mathbf{t}_{11}\widetilde{\otimes}\mathbf{t}_1\mathbf{t}_2+\mathbf{t}_{12}\widetilde{\otimes}\mathbf{t}_1\mathbf{t}_4+\tfrac{1}{2}\mathbf{t}_{13}\widetilde{\otimes}\mathbf{t}_2^2+\tfrac{1}{6}\mathbf{t}_{14}\widetilde{\otimes}\mathbf{t}_5\\
&\quad+\mathbf{t}_{15}\widetilde{\otimes}\mathbf{t}_6+\tfrac{1}{2}\mathbf{t}_{16}\widetilde{\otimes}\mathbf{t}_7+\mathbf{t}_{17}\widetilde{\otimes}\mathbf{t}_8,\\
\widetilde{\Delta}(\mathbf{t}_2)&=\mathbf{t}_2\widetilde{\otimes}1+\mathbf{t}_3\widetilde{\otimes}\mathbf{t}_1+\mathbf{t}_4\widetilde{\otimes}\mathbf{t}_1+\tfrac{1}{2}\mathbf{t}_5\widetilde{\otimes}\mathbf{t}_1^2+\mathbf{t}_6\widetilde{\otimes}(\mathbf{t}_1^2+\mathbf{t}_2)+\tfrac{1}{2}\mathbf{t}_7\widetilde{\otimes}\mathbf{t}_1^2+\mathbf{t}_8\widetilde{\otimes}\mathbf{t}_2\\
&\quad+\tfrac{1}{6}\mathbf{t}_9\widetilde{\otimes}\mathbf{t}_1^3+\mathbf{t}_{10}\widetilde{\otimes}(\tfrac{1}{2}\mathbf{t}_1^3+\mathbf{t}_1\mathbf{t}_2)+\tfrac{1}{2}(\mathbf{t}_{11}\widetilde{\otimes}(\mathbf{t}_1^3+\mathbf{t}_3)+\mathbf{t}_{12}\widetilde{\otimes}(\mathbf{t}_1\mathbf{t}_2+\mathbf{t}_4)\\
&\quad+\mathbf{t}_{13}\widetilde{\otimes}\mathbf{t}_1\mathbf{t}_2+\tfrac{1}{6}\mathbf{t}_{14}\widetilde{\otimes}\mathbf{t}_1^3+\mathbf{t}_{15}\widetilde{\otimes}\mathbf{t}_1\mathbf{t}_2+\tfrac{1}{2}\mathbf{t}_{16}\widetilde{\otimes}\mathbf{t}_3+\mathbf{t}_{17}\widetilde{\otimes}\mathbf{t}_4,\\
\widetilde{\Delta}(\mathbf{t}_3)&=\mathbf{t}_3\widetilde{\otimes}1+\mathbf{t}_5\widetilde{\otimes}\mathbf{t}_1+2\mathbf{t}_6\widetilde{\otimes}\mathbf{t}_1+\tfrac{1}{2}\mathbf{t}_9\widetilde{\otimes}\mathbf{t}_1^2+\mathbf{t}_{10}\widetilde{\otimes}(2\mathbf{t}_1^2+\mathbf{t}_2)+\mathbf{t}_{11}\widetilde{\otimes}\mathbf{t}_1^2+2\mathbf{t}_{12}\widetilde{\otimes}\mathbf{t}_2\\
&\quad+\mathbf{t}_{13}\widetilde{\otimes}\mathbf{t}_1^2,\\
\widetilde{\Delta}(\mathbf{t}_4)&=\mathbf{t}_4\widetilde{\otimes}1+\mathbf{t}_6\widetilde{\otimes}\mathbf{t}_1+\mathbf{t}_7\widetilde{\otimes}\mathbf{t}_1+\mathbf{t}_8\widetilde{\otimes}\mathbf{t}_1+\tfrac{1}{2}\mathbf{t}_{10}\widetilde{\otimes}\mathbf{t}_1^2+\mathbf{t}_{11}\widetilde{\otimes}\mathbf{t}_1^2+\mathbf{t}_{12}\widetilde{\otimes}\mathbf{t}_1^2\\
&\quad+\mathbf{t}_{13}\widetilde{\otimes}\mathbf{t}_2+\tfrac{1}{2}\mathbf{t}_{14}\widetilde{\otimes}\mathbf{t}_1^2+\mathbf{t}_{15}\widetilde{\otimes}(\mathbf{t}_1^2+\mathbf{t}_2)+\tfrac{1}{2}\mathbf{t}_{16}\widetilde{\otimes}\mathbf{t}_1^2+\mathbf{t}_{17}\widetilde{\otimes}\mathbf{t}_2.
\end{aligned}$$

2.9 Antipodes of trees and forests

The name "antipode" is adopted from Hopf-algebra terminology. In the present context, the antipode is a particular mapping antipode : $\mathsf{F}\to\mathsf{F}$ with the properties of an involution. That is antipode $\circ$ antipode $=$ id.

We will construct antipode in three steps. First, antipode : $\mathrm{T}\to\mathsf{F}$ will be defined and this will then be extended to antipode : $\mathrm{F}\to\mathsf{F}$ and then to antipode : $\mathsf{F}\to\mathsf{F}$.

The antipode of a tree

Definition 2.9A A partition of a tree $\mathrm{t}=(V,E,r)$ is the forest formed from the set of trees induced by a partition of V. The set of all partitions is denoted by $P(\mathrm{t})$. For $\mathrm{f}=\mathrm{t}_1\mathrm{t}_2\cdots\mathrm{t}_n\in P(\mathrm{t})$, the corresponding signed product is

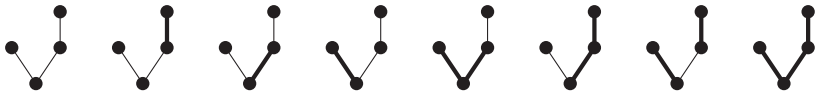
$$(-1)^n\prod_{i=1}^{n}\mathrm{t}_i. \tag{2.9a}$$

Table 10 antipode(t), for $|t| \leq 5$

$\lvert t\rvert$	t	antipode(t)
1	$\mathbf{t}_1$	$-\mathbf{t}_1$
2	$\mathbf{t}_2$	$\mathbf{t}_1^2 - \mathbf{t}_2$
3	$\mathbf{t}_3$	$-\mathbf{t}_1^3 + 2\mathbf{t}_1\mathbf{t}_2 - \mathbf{t}_3$
3	$\mathbf{t}_4$	$-\mathbf{t}_1^3 + 2\mathbf{t}_1\mathbf{t}_2 - \mathbf{t}_4$
4	$\mathbf{t}_5$	$\mathbf{t}_1^4 - 3\mathbf{t}_1^2\mathbf{t}_2 + 3\mathbf{t}_1\mathbf{t}_3 - \mathbf{t}_5$
4	$\mathbf{t}_6$	$\mathbf{t}_1^4 - 3\mathbf{t}_1^2\mathbf{t}_2 + \mathbf{t}_1\mathbf{t}_3 + \mathbf{t}_1\mathbf{t}_4 + \mathbf{t}_2^2 - \mathbf{t}_6$
4	$\mathbf{t}_7$	$\mathbf{t}_1^4 - 3\mathbf{t}_1^2\mathbf{t}_2 + \mathbf{t}_1\mathbf{t}_3 + 2\mathbf{t}_1\mathbf{t}_4 - \mathbf{t}_7$
4	$\mathbf{t}_8$	$\mathbf{t}_1^4 - 3\mathbf{t}_1^2\mathbf{t}_2 + 2\mathbf{t}_1\mathbf{t}_4 + \mathbf{t}_2^2 - \mathbf{t}_8$
5	$\mathbf{t}_9$	$-\mathbf{t}_1^5 + 4\mathbf{t}_1^3\mathbf{t}_2 - 6\mathbf{t}_1^2\mathbf{t}_3 + 4\mathbf{t}_1\mathbf{t}_5 - \mathbf{t}_9$
5	$\mathbf{t}_{10}$	$-\mathbf{t}_1^5 + 4\mathbf{t}_1^3\mathbf{t}_2 - 3\mathbf{t}_1^2\mathbf{t}_3 - \mathbf{t}_1^2\mathbf{t}_4 - 2\mathbf{t}_1\mathbf{t}_2^2 + \mathbf{t}_1\mathbf{t}_5 + 2\mathbf{t}_1\mathbf{t}_6 + \mathbf{t}_2\mathbf{t}_3 - \mathbf{t}_{10}$
5	$\mathbf{t}_{11}$	$-\mathbf{t}_1^5 + 4\mathbf{t}_1^3\mathbf{t}_2 - 2\mathbf{t}_1^2\mathbf{t}_3 - 2\mathbf{t}_1^2\mathbf{t}_4 - 2\mathbf{t}_1\mathbf{t}_2^2 + 2\mathbf{t}_1\mathbf{t}_6 + \mathbf{t}_1\mathbf{t}_7 + \mathbf{t}_2\mathbf{t}_3 - \mathbf{t}_{11}$
5	$\mathbf{t}_{12}$	$-\mathbf{t}_1^5 + 4\mathbf{t}_1^3\mathbf{t}_2 - \mathbf{t}_1^2\mathbf{t}_3 - 2\mathbf{t}_1^2\mathbf{t}_4 - 3\mathbf{t}_1\mathbf{t}_2^2 + 2\mathbf{t}_1\mathbf{t}_5 + \mathbf{t}_1\mathbf{t}_7 + \mathbf{t}_2\mathbf{t}_3 + \mathbf{t}_2\mathbf{t}_4 - \mathbf{t}_{12}$
5	$\mathbf{t}_{13}$	$-\mathbf{t}_1^5 + 4\mathbf{t}_1^3\mathbf{t}_2 - \mathbf{t}_1^2\mathbf{t}_3 - 2\mathbf{t}_1^2\mathbf{t}_4 - 3\mathbf{t}_1\mathbf{t}_2^2 + 2\mathbf{t}_1\mathbf{t}_6 + 2\mathbf{t}_2\mathbf{t}_4 - \mathbf{t}_{13}$
5	$\mathbf{t}_{14}$	$-\mathbf{t}_1^5 + 4\mathbf{t}_1^3\mathbf{t}_2 - 3\mathbf{t}_1^2\mathbf{t}_3 - 3\mathbf{t}_1^2\mathbf{t}_4 + \mathbf{t}_1\mathbf{t}_5 + 3\mathbf{t}_1\mathbf{t}_7 - \mathbf{t}_{14}$
5	$\mathbf{t}_{15}$	$-\mathbf{t}_1^5 + 4\mathbf{t}_1^3\mathbf{t}_2 - \mathbf{t}_1^2\mathbf{t}_3 - 2\mathbf{t}_1^2\mathbf{t}_4 - 3\mathbf{t}_1\mathbf{t}_2^2 + \mathbf{t}_1\mathbf{t}_6 + \mathbf{t}_1\mathbf{t}_8 + \mathbf{t}_2\mathbf{t}_3 + \mathbf{t}_2\mathbf{t}_4 - \mathbf{t}_{15}$
5	$\mathbf{t}_{16}$	$-\mathbf{t}_1^5 + 4\mathbf{t}_1^3\mathbf{t}_2 - \mathbf{t}_1^2\mathbf{t}_3 - 3\mathbf{t}_1^2\mathbf{t}_4 - 2\mathbf{t}_1\mathbf{t}_2^2 + \mathbf{t}_1\mathbf{t}_7 + 2\mathbf{t}_1\mathbf{t}_8 + \mathbf{t}_2\mathbf{t}_3 - \mathbf{t}_{16}$
5	$\mathbf{t}_{17}$	$-\mathbf{t}_1^5 + 4\mathbf{t}_1^3\mathbf{t}_2 - 3\mathbf{t}_1^2\mathbf{t}_4 - 3\mathbf{t}_1\mathbf{t}_2^2 + 2\mathbf{t}_1\mathbf{t}_8 + 2\mathbf{t}_2\mathbf{t}_4 - \mathbf{t}_{17}$

Definition 2.9B The antipode of a tree t is the sum of all the signed partitions of t, written antipode(t).

For example, the partitions of $\mathbf{t}_6$ are found from the following diagrams, where thin lines show where edges have been removed to reveal each partition:



This leads to

$$\text{antipode}(\mathbf{t}_6) = \mathbf{t}_1^4 - 3\mathbf{t}_1^2\mathbf{t}_2 + \mathbf{t}_1\mathbf{t}_3 + \mathbf{t}_1\mathbf{t}_4 + \mathbf{t}_2^2 - \mathbf{t}_6.$$

For later reference, antipode(t), for $|t| \leq 5$, is given in Table 10.

The antipode of forests

Definition 2.9C The antipode of a forest is the formal product of the antipodes of the trees which comprise the forest. That is,

$$\text{antipode}\left(\prod_{i=1}^{n} \mathrm{t}_i\right) = \prod_{i=1}^{n} \text{antipode}(\mathrm{t}_i).$$

The antipode of a linear combination of forests is the corresponding linear combination of the antipodes of the individual forests. That is

$$\text{antipode}\left(\sum_{i=1}^{n} C_i \mathrm{f}_i\right) = \sum_{i=1}^{n} C_i \,\text{antipode}(\mathrm{f}_i).$$

Theorem 2.9D The antipode of a forest can also be defined in the same way as the antipode of a tree, based on the partitions of the given forest.

Theorem 2.9E The antipode of $\mathrm{t} \in \mathrm{T}$ satisfies the recursion:

$$\text{antipode}(\tau) = -\tau, \tag{2.9 b}$$

$$\begin{aligned} \text{antipode}(\mathrm{t}) &= -\sum_{\mathrm{t}' \le \mathrm{t}} \text{antipode}(\mathrm{t} \smallsetminus \mathrm{t}')\mathrm{t}' \\ &= -\sum_{\mathrm{t}' < \mathrm{t}} \text{antipode}(\mathrm{t} \smallsetminus \mathrm{t}')\mathrm{t}' - \mathrm{t}. \end{aligned} \tag{2.9 c}$$

Proof. The case (2.9 b) follows because there is only a single partition for τ. For (2.9 c) collect all partitions for which t' is the component containing the root. For each of the partitions of $\mathrm{t} \smallsetminus \mathrm{t}'$, $(-\mathrm{t}')$ is the additional factor to construct the signed product of $(\mathrm{t} \smallsetminus \mathrm{t}')\mathrm{t}'$. □

Exercise 30 Show that

$$\text{antipode}([\tau^n]) = (-1)^{n+1}\tau^{n+1} - \sum_{i=1}^{n-1} (-1)^{n-i} \binom{n}{i} [\tau^i]\tau^{n-i} - [\tau^n].$$

Exercise 31 Find $\text{antipode}([_3\tau]_3)$.

The involution and other antipode properties

Lemma 2.9F For any $\mathrm{t} \in \mathrm{T}$,

$$\sum_{\mathrm{t}' < \mathrm{t}} \mathrm{antipode}(\mathrm{t} \smallsetminus \mathrm{t}')\mathrm{t}' = \sum_{\mathrm{t}' < \mathrm{t}} (\mathrm{t} \smallsetminus \mathrm{t}')\,\mathrm{antipode}(\mathrm{t}'). \tag{2.9 d}$$

Proof. The proof is by induction on $n = |\mathrm{t}|$. For $n = 1$, $\mathrm{t} = \tau$ and the two sides of (2.9 d) both vanish. For $n > 1$, use (2.9 c) from Theorem 2.9E in the form

$$\mathrm{antipode}(\mathrm{t}') = -\sum_{\mathrm{t}'' \leq \mathrm{t}} \mathrm{antipode}(\mathrm{t}' \smallsetminus \mathrm{t}'')\mathrm{t}''.$$

Substitute this formula into the right-hand side of (2.9 d) and replace the variable t' by t'' on the left hand side. The result is

$$\sum_{\mathrm{t}'' < \mathrm{t}} \mathrm{antipode}(\mathrm{t} \smallsetminus \mathrm{t}'')\mathrm{t}'' = -\sum_{\mathrm{t}' < \mathrm{t}} (\mathrm{t} \smallsetminus \mathrm{t}') \sum_{\mathrm{t}'' \leq \mathrm{t}'} \mathrm{antipode}(\mathrm{t}' \smallsetminus \mathrm{t}'')\mathrm{t}''$$

and we will prove that for $\mathrm{t}'' < \mathrm{t}$,

$$\mathrm{antipode}(\mathrm{t} \smallsetminus \mathrm{t}'') = -\sum_{\mathrm{t}' \colon \mathrm{t}'' \leq \mathrm{t}' < \mathrm{t}} (\mathrm{t} \smallsetminus \mathrm{t}')\,\mathrm{antipode}(\mathrm{t}' \smallsetminus \mathrm{t}''),$$

which can be written

$$\sum_{\mathrm{t}' \colon \mathrm{t}'' \leq \mathrm{t}' \leq \mathrm{t}} (\mathrm{t} \smallsetminus \mathrm{t}'') \smallsetminus (\mathrm{t}' \smallsetminus \mathrm{t}'')\,\mathrm{antipode}(\mathrm{t}' \smallsetminus \mathrm{t}'') = 0. \tag{2.9 e}$$

Write $\mathrm{t} \smallsetminus \mathrm{t}'' = \prod_{i=1}^{n} \mathrm{t}_i$ and $\mathrm{t}' \smallsetminus \mathrm{t}'' = \prod_{i=1}^{n} \mathrm{t}'_i$, where $\mathrm{t}'_i \leq \mathrm{t}_i$, so that (2.9 e) becomes

$$\prod_{i=1}^{n} \Big(\mathrm{t}_i + \sum_{\mathrm{t}'_i \leq \mathrm{t}_i} (\mathrm{t}_i \smallsetminus \mathrm{t}'_i)\,\mathrm{antipode}(\mathrm{t}'_i) \Big) = 0,$$

and this follows from (2.9 d) with t replaced by t_i, where we note that $|\mathrm{t}_i| < n$. □

Theorem 2.9G The antipode is an involution. That is

$$\mathrm{antipode} \circ \mathrm{antipode} = \mathrm{id}.$$

Proof. The case $\mathrm{antipode}(\mathrm{antipode}(\mathrm{t})) = \mathrm{t}$ follows from Lemma 2.9F. This extends to a linear combination of forests. □

Recursions for antipodes

Beta product recursion

The signed product (2.9 a), used in Definition 2.9A, can be written in the form

$$\prod_{i=1}^{n}(-\mathrm{t}_i) = -\prod_{i=2}^{n}(-\mathrm{t}_i)\mathrm{t}_1,$$

where the unique component of the partition which contains the root is designated as number 1.

Hence, the sum of the signed products can be written as

$$\theta_1(\mathrm{t})\mathbf{t}_1 + \theta_2(\mathrm{t})\mathbf{t}_2 + \cdots,$$

where θ_1, θ_2, …, are formed from the factors in the various signed partitions which do contain the root.

For example,

$$\begin{aligned}
\text{antipode}(\mathbf{t}_1) &= (-1)\mathbf{t}_1 &&= \theta(\mathbf{t}_1)^{\intercal}\widehat{\mathrm{T}},\\
\text{antipode}(\mathbf{t}_2) &= (\mathbf{t}_1)\mathbf{t}_1 + (-1)\mathbf{t}_2 &&= \theta(\mathbf{t}_2)^{\intercal}\widehat{\mathrm{T}},\\
\text{antipode}(\mathbf{t}_3) &= (-\mathbf{t}_1^2)\mathbf{t}_1 + (2\mathbf{t}_1)\mathbf{t}_2 + (-1)\mathbf{t}_3 &&= \theta(\mathbf{t}_3)^{\intercal}\widehat{\mathrm{T}},\\
\text{antipode}(\mathbf{t}_3) &= (\mathbf{t}_2 - \mathbf{t}_1^2)\mathbf{t}_1 + (\mathbf{t}_1)\mathbf{t}_2 + (-1)\mathbf{t}_4 &&= \theta(\mathbf{t}_4)^{\intercal}\widehat{\mathrm{T}},
\end{aligned}$$

where $\widehat{\mathrm{T}}$ is the list of all trees, written as an infinite-dimensional vector

$$\widehat{\mathrm{T}} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \mathbf{t}_3 & \cdots \end{bmatrix}^{\intercal}$$

and, for each t, $\theta(\mathrm{t})^{\intercal}$ is a terminating row vector,

$$\begin{aligned}
\theta(\mathbf{t}_1)^{\intercal} &= \begin{bmatrix} -1 \end{bmatrix},\\
\theta(\mathbf{t}_2)^{\intercal} &= \begin{bmatrix} \mathbf{t}_1 & -1 \end{bmatrix},\\
\theta(\mathbf{t}_3)^{\intercal} &= \begin{bmatrix} -\mathbf{t}_1^2 & 2\mathbf{t}_1 & -1 \end{bmatrix},\\
\theta(\mathbf{t}_4)^{\intercal} &= \begin{bmatrix} \mathbf{t}_2 - \mathbf{t}_1^2 & \mathbf{t}_1 & 0 & -1 \end{bmatrix},
\end{aligned}$$

and in the evaluation of $\theta(\mathrm{t})^{\intercal}\widehat{\mathrm{T}}$, only non-zero terms are included.

As an alternative to this formulation, we introduce a variant of Sweedler's notation [89] (Underwood, 2011), where we define

$$\nabla(\mathrm{t}) = \theta(\mathrm{t})^{\intercal} \otimes \widehat{\mathrm{T}},$$

with "$\otimes$" inserted between the two factors indicates how the inner product is to be interpreted.

As examples we have

$$\begin{aligned}
\nabla(\mathbf{t}_1) &= -1 \otimes \mathbf{t}_1,\\
\nabla(\mathbf{t}_2) &= \mathbf{t}_1 \otimes \mathbf{t}_1 - 1 \otimes \mathbf{t}_2,\\
\nabla(\mathbf{t}_3) &= -\mathbf{t}_1^2 \otimes \mathbf{t}_1 + 2\mathbf{t}_1 \otimes \mathbf{t}_2 - 1 \otimes \mathbf{t}_3,\\
\nabla(\mathbf{t}_4) &= (\mathbf{t}_2 - \mathbf{t}_1^2) \otimes \mathbf{t}_1 + \mathbf{t}_1 \otimes \mathbf{t}_2 - 1 \otimes \mathbf{t}_4.
\end{aligned}$$

We will use the notation expand$(\nabla(\mathrm{t}))$ to denote $\theta(\mathrm{t})^{\intercal}\widehat{\mathrm{T}}$, which is just $\nabla(\mathrm{t})$, with every occurrence of $\otimes$ replaced by multiplication in the forest ring. Recursion under the beta product satisfies the following properties:

Theorem 2.9H The values of $\nabla(\mathrm{t})$ and antipode(t) satisfy the recursions

$$\nabla(\tau) = -1 \otimes \tau, \tag{2.9 f}$$

$$\text{antipode}(\tau) = -\tau, \tag{2.9 g}$$

$$\nabla(\mathrm{t} * \mathrm{t}') = \text{antipode}(\mathrm{t}')\nabla(\mathrm{t}) - \nabla(\mathrm{t}) * \nabla(\mathrm{t}'), \tag{2.9 h}$$

$$\text{antipode}(\mathrm{t} * \mathrm{t}') = \text{expand}(\nabla(\mathrm{t} * \mathrm{t}')). \tag{2.9 i}$$

Proof. The results of (2.9 f), (2.9 g), (2.9 i) follow from the above discussions. To prove (2.9 h), we see that antipode$(\mathrm{t}')\nabla(\mathrm{t})$ includes all partitions for which the component containing the root does not contain any vertex from t', while $-\nabla(\mathrm{t}) * \nabla(\mathrm{t}')$ contains partitions for which the component containing the root also contains one or more vertex from t'. □

B^+ *recursion for antipodes*

Let

$$\mathrm{t} = [\mathrm{t}_1 \mathrm{t}_2 \cdots \mathrm{t}_n]. \tag{2.9 j}$$

Our aim is to write $\nabla(\mathrm{t})$ in terms of $\nabla(\mathrm{t}_1)$, $\nabla(\mathrm{t}_2)$, …, $\nabla(\mathrm{t}_n)$. This will give an alternative to Theorem 2.9H to evaluate $\nabla(\mathrm{t})$ and therefore antipode(t) for any tree.

Theorem 2.9I If t is given by (2.9 j), then

$$\nabla(\mathrm{t}) = \Big[\prod_{i=1}^{n} \big(\text{antipode}(\mathrm{t}_i) \otimes \varnothing - \nabla(\mathrm{t}_i)\big)\Big].$$

Proof. The terms in $\nabla(t)$ are of the form

$$C \otimes \Big[\prod_{i=1}^{n} \mathrm{t}_i'\Big],$$

where the numbering of the factors is the same as in (2.9 j) but with the possibility that some factors are missing. The factors that are present come from the corresponding $\nabla(\mathrm{t}_i)$ with $\mathrm{t}'_i \leq \mathrm{t}_i$, and the missing factors come from antipode$(\mathrm{t}_i) \otimes \varnothing$. □

Multiplicative functions on trees and forests

At this point we introduce a formal definition on mappings from $\mathrm{F} \to \mathbb{R}$.

Definition 2.9J A function on $a : \mathrm{F} \to \mathbb{R}$ is multiplicative if

$$\begin{aligned} a(1) &= 1, \\ a(\mathrm{ft}) &= a(\mathrm{f})a(\mathrm{t}), \qquad \mathrm{f} \in \mathrm{F}, \mathrm{t} \in \mathrm{T}. \end{aligned}$$

Another way of looking at this property is:

$$a\Big(\prod_{i=1}^{m} \mathrm{t}_i\Big) = \prod_{i=1}^{m} a(\mathrm{t}_i), \tag{2.9 k}$$

with the understanding that the forest $\prod_{i=1}^{m} \mathrm{t}_i$ is the identity forest if the product is empty.

If $a : \mathrm{T} \to \mathbb{R}$, then we will write $a : \mathrm{F} \to \mathbb{R}$ as the multiplicative extension of a defined by (2.9 k).

Subtrees as mappings

As an alternative to the use of the Sweedler notation, it is convenient to use functions on trees and related graphs to characterize specific tree prunings. If $\mathrm{t}' \leq \mathrm{t}$, then the corresponding term $\mathrm{f} \otimes \mathrm{t}'$ can be written as

$$a(\mathrm{f})b(\mathrm{t}'),$$

where a and b are unspecified mappings.

We will formally write

$$(ab)(\mathrm{t}) = \sum_{\mathrm{T}^{\#} \ni \mathrm{t}' \leq \mathrm{t}} a(\mathrm{t} \smallsetminus \mathrm{t}')b(\mathrm{t}'),$$

where a is multiplicative.

For example,

$$\begin{aligned} (ab)(\varnothing) &= b(\varnothing), \\ (ab)(\mathbf{t}_1) &= a(\mathbf{t}_1)b(\varnothing) + b(\mathbf{t}_1), \\ (ab)(\mathbf{t}_2) &= a(\mathbf{t}_2)b(\varnothing) + a(\mathbf{t}_1)b(\mathbf{t}_1) + b(\mathbf{t}_2), \\ (ab)(\mathbf{t}_3) &= a(\mathbf{t}_3)b(\varnothing) + a(\mathbf{t}_1)^2 b(\mathbf{t}_1) + 2a(\mathbf{t}_1)(\mathbf{t}_2) + b(\mathbf{t}_3), \\ (ab)(\mathbf{t}_4) &= a(\mathbf{t}_4)b(\varnothing) + a(\mathbf{t}_2)b(\mathbf{t}_1) + a(\mathbf{t}_1)(\mathbf{t}_2) + b(\mathbf{t}_4). \end{aligned}$$

Symbolic and algebraic interpretations

As a symbolic statement, a formula for $(ab)(\mathrm{t})$ is nothing more than an alternative to the Sweedler notation and expresses exactly the same information on the set of all possible prunings of t. However, if two specific mappings, a, b, from T to $\mathbb{R}$ are given then $(ab)(\mathrm{t})$ is the definition of a new mapping constructed from a and b.

Linear algebra representations

A matrix group over F

Consider the set G of infinite matrices with components of the form $g \in G$, where

$$g_{ij} = \begin{cases} \text{a member of } \mathsf{F}, & j < i, \\ 1, & j = i, \\ 0, & j > i. \end{cases}$$

Theorem 2.9K G is a group.

In applications of G, it will be convenient to index the rows and columns using T with the members of T, arranged in a convenient order, such as using the standard numbering of trees.

Symbolic vector and matrix

Let v denote the symbolic vector $[\,\varnothing \quad \mathbf{t}_1 \quad \mathbf{t}_2 \quad \cdots\,]^{\mathsf{T}}$, and $\boldsymbol{\Lambda}$ the symbolic matrix

$$\boldsymbol{\Lambda} = \begin{bmatrix} 1 & & & \\ \mathbf{t}_1 & 1 & & \\ \mathbf{t}_2 & \mathbf{t}_1 & 1 & \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

in which the $(\mathrm{t}, \mathrm{t}')$ element is the sum of the forests $\sum \mathrm{f}_i$ over all terms of the form $\mathrm{f}_i \otimes \mathrm{t}$ occurring in $\Delta(\mathrm{t})$. For example, from

$$\Delta(\mathrm{t}_3) = \mathrm{t}_3 \otimes \varnothing + \mathrm{t}_1^2 \otimes \mathrm{t}_1 + 2\mathrm{t}_1\mathrm{t}_2 + 1 \otimes \mathrm{t}_3,$$

it follows that row number 4 of $\boldsymbol{\Lambda}$ is

$$\begin{bmatrix} \mathrm{t}_3 & \mathrm{t}_1^2 & 2\mathrm{t}_1 & 1 & \cdots \end{bmatrix}.$$

If the product $\boldsymbol{\Lambda} \otimes \mathsf{v}$ is interpreted as the usual matrix-vector multiplication, but with $\otimes$ inserted within every term that arises, then we can write

$$\boldsymbol{\Lambda} \otimes \mathsf{v} = \Delta(\mathsf{v}).$$

Furthermore, many results arising in Sections 2.8 and 2.9 have a fresh and useful interpretation when written in terms of v and $\boldsymbol{\Lambda}$.

Theorem 2.9L

$$\text{antipode}(\boldsymbol{\Lambda}) = \boldsymbol{\Lambda}^{-1}.$$

Proof. This follows from Theorem 2.9E. □

To illustrate this result, the values of $\boldsymbol{\Lambda}$ and antipode$(\boldsymbol{\Lambda})$ have been calculated up to trees of order 4. A simple matrix multiplication confirms that $\boldsymbol{\Lambda}$ antipode$(\boldsymbol{\Lambda}) = I$. The matrices are

$$\boldsymbol{\Lambda} = \begin{bmatrix} 1 \\ \mathbf{t}_1 & 1 \\ \mathbf{t}_2 & \mathbf{t}_1 & 1 \\ \mathbf{t}_3 & \mathbf{t}_1^2 & 2\mathbf{t}_1 & 1 \\ \mathbf{t}_4 & \mathbf{t}_2 & \mathbf{t}_1 & 0 & 1 \\ \mathbf{t}_5 & \mathbf{t}_1^3 & 3\mathbf{t}_1^2 & 3\mathbf{t}_1 & 0 & 1 \\ \mathbf{t}_6 & \mathbf{t}_1\mathbf{t}_2 & \mathbf{t}_1^2+\mathbf{t}_2 & \mathbf{t}_1 & \mathbf{t}_1 & 0 & 1 \\ \mathbf{t}_7 & \mathbf{t}_3 & \mathbf{t}_1^2 & 0 & 2\mathbf{t}_1 & 0 & 0 & 1 \\ \mathbf{t}_8 & \mathbf{t}_4 & \mathbf{t}_2 & 0 & \mathbf{t}_1 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\text{antipode}(\boldsymbol{\Lambda}) = \begin{bmatrix} 1 \\ -\mathbf{t}_1 & 1 \\ \mathbf{t}_1^2-\mathbf{t}_2 & -\mathbf{t}_1 & 1 \\ -\mathbf{t}_1^3+2\mathbf{t}_1\mathbf{t}_2-\mathbf{t}_3 & \mathbf{t}_1^2 & -2\mathbf{t}_1 & 1 \\ -\mathbf{t}_1^3+2\mathbf{t}_1\mathbf{t}_2-\mathbf{t}_4 & \mathbf{t}_1^2-\mathbf{t}_2 & -\mathbf{t}_1 & 0 & 1 \\ \mathbf{t}_1^4-3\mathbf{t}_1^2\mathbf{t}_2+3\mathbf{t}_1\mathbf{t}_3-\mathbf{t}_5 & -\mathbf{t}_1^3 & 3\mathbf{t}_1^2 & -3\mathbf{t}_1 & 0 & 1 \\ \mathbf{t}_1^4-3\mathbf{t}_1^2\mathbf{t}_2+\mathbf{t}_1\mathbf{t}_3+\mathbf{t}_1\mathbf{t}_4+\mathbf{t}_2^2-\mathbf{t}_6 & \mathbf{t}_1\mathbf{t}_2-\mathbf{t}_1^3 & 2\mathbf{t}_1^2-\mathbf{t}_2 & -\mathbf{t}_1 & -\mathbf{t}_1 & 0 & 1 \\ \mathbf{t}_1^4-3\mathbf{t}_1^2\mathbf{t}_2+\mathbf{t}_1\mathbf{t}_3+2\mathbf{t}_1\mathbf{t}_4-\mathbf{t}_7 & -\mathbf{t}_1^3+2\mathbf{t}_1\mathbf{t}_2-\mathbf{t}_3 & \mathbf{t}_1^2 & 0 & -2\mathbf{t}_1 & 0 & 0 & 1 \\ \mathbf{t}_1^4-3\mathbf{t}_1^2\mathbf{t}_2+2\mathbf{t}_1\mathbf{t}_4+\mathbf{t}_2^2-\mathbf{t}_8 & -\mathbf{t}_1^3+2\mathbf{t}_1\mathbf{t}_2-\mathbf{t}_4 & \mathbf{t}_1^2-\mathbf{t}_2 & 0 & -\mathbf{t}_1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Summary of Chapter 2 and the way forward

Summary

The basic terminology of graphs, as vertex-edge pairs is presented, leading to the definitions of trees, in the sense of rooted trees, and unrooted, or free, trees. It is shown how to build up trees from the tree τ, with a single vertex, using the beta-product $\mathrm{t} * \mathrm{t}'$ and the B^+ operation $[\mathrm{t}_1\mathrm{t}_2\cdots\mathrm{t}_n]$. The prefix (Polish) operator τ_n ($n = 1, 2, \ldots$) is introduced as denoting $\tau_n \mathrm{f} = [\mathrm{f}]$, where $n = |\mathrm{f}|$.

Formal linear combinations of forests are introduced as the "forest space" and, as an application, the enumeration of trees is considered. Partitions of sets and numbers are introduced and related to trees. The evolution of partitions and of trees, is discussed as the complexity is increased step by step. Subtrees and prunings are introduced and the function $\Delta(\mathrm{t})$ is introduced and its properties are reviewed. The antipode is introduced and its properties discussed, including the involution property. The interplay between groups, linear algebra and trees is briefly investigated. Truncated trees and forests are introduced. This leads to the algebra of linear combinations of uni-valent stumps which act as linear operators on the tree space.

The way forward

Trees, and related graph-theoretic structures, are at the heart of this subject and a good understanding of the present chapter is desirable before venturing too far further ahead.

Teaching and study notes

The subject of this chapter is an essential component of any study of numerical methods for ordinary differential equations. In particular it is a prelude to a serious study of B-series. Many students of this subject will already have an acquaintance with graph-theory and combinatorics and this will provide a useful background. However, the presentation in this chapter should not be taken lightly because of the distinctive notations and conventions introduced here; later chapters are built on this and related terminology. Rooted trees are given a standard numbering, displayed in Table 6 (p. 63), and this will be used throughout the book. The most important constructions and results, which come as the culmination of earlier work, are in Sections 2.8 and 2.9. These are essential background for the algebraic structures introduced in Chapter 3.

Projects

Project 5 Read up about the asymptotic behaviour of the sequences $[a_1, a_2, \ldots]$, $[b_1, b_2, \ldots]$, $[c_1, c_2, \ldots]$, in (2.4 b).

Project 6 Learn about Hopf Algebras starting with [4] and [89].

Project 7 Find out about the life of J. Łukasiewicz, the founder of Polish notation.

Project 8 Write your own implementation of Algorithm 3 (p. 64).