

On the Hopf Algebraic Structure of Lie Group Integrators

H. Z. Munthe-Kaas¹ and W. M. Wright²

¹Department of Mathematics
University of Bergen
Johannes Brunsgate 12
N-5008 Norway
Hans.Munthe-Kaas@mi.uib.no

²Department of Mathematics
La Trobe University
Victoria 3086, Australia
W.Wright@latrobe.edu.au

Abstract. A commutative but not cocommutative graded Hopf algebra \mathcal{H}_N , based on ordered (planar) rooted trees, is studied. This Hopf algebra is a generalization of the Hopf algebraic structure of unordered rooted trees \mathcal{H}_C , developed by Butcher in his study of Runge-Kutta methods and later rediscovered by Connes and Moscovici in the context of noncommutative geometry and by Kreimer where it is used to describe renormalization in quantum field theory. It is shown that \mathcal{H}_N is naturally obtained from a universal object in a category of noncommutative derivations and, in particular, it forms a foundation for the study of numerical integrators based on noncommutative Lie group actions on a manifold. Recursive and nonrecursive definitions of the coproduct and the antipode are derived. The relationship between \mathcal{H}_N and four other Hopf algebras is discussed. The dual of \mathcal{H}_N is a Hopf algebra of Grossman and Larson based on ordered rooted trees. The Hopf algebra \mathcal{H}_C of Butcher, Connes, and Kreimer is identified as a proper Hopf subalgebra of \mathcal{H}_N using the image of a tree symmetrization operator. The Hopf algebraic structure of the shuffle algebra \mathcal{H}_{Sh} is obtained from \mathcal{H}_N by a quotient construction. The Hopf algebra \mathcal{H}_P of ordered trees by Foissy differs from \mathcal{H}_N in the definition of the product (noncommutative

Date received: February 14, 2006. Final version received: August 29, 2006. Date accepted: September 6, 2006. Communicated by Arieh Iserles. Online publication: April 25, 2007.

AMS classification: 16W25, 16W30, 22E60, 37M99, 65L05, 65L06, 81R60, 81T15.

Key words and phrases: Hopf algebra, Ordered rooted trees, Lie group integrators, Lie–Butcher series, Butcher group, Connes–Kreimer Hopf algebra, Grossman–Larson Hopf algebra, Shuffle algebra.

concatenation for \mathcal{H}_P and shuffle for \mathcal{H}_N) and the definitions of the coproduct and the antipode, however, these are related through the tree symmetrization operator.

1. Introduction

Since Cayley [7] in 1857 and, 100 years later, Merson [24] and followed shortly after by Butcher [5], it has been known that rooted trees are extremely useful for structuring algebras of differential operators and elementary differentials. In 1972 Butcher [6] produced the far-reaching result that Runge-Kutta methods form a group. This was later named the Butcher group in the paper by Hairer and Wanner [18], who made significant contributions to this theory. The Butcher group is defined on the dual of the vector space spanned by the set of unordered rooted trees and it was pointed out, by Dür [14], that there exists a one-to-one correspondence between the Butcher group and the commutative graded Hopf algebra of unordered rooted trees. The Hopf algebra of unordered rooted trees has had far-reaching applications in various areas of mathematics and physics. In 1998 a Hopf subalgebra was discovered, by Connes and Moscovici [12], during work on an index theorem in noncommutative geometry and, by Kreimer [21], in the renormalization method of quantum field theory. Further collaborations between Connes and Kreimer [10], [11] have led to other surprising results; notably a connection with the Riemann–Hilbert problem was established. Brouder [3], [4] realized that the mathematical structure of Connes and Kreimer was the same as that of Butcher. Grossman and Larson [17] also developed cocommutative graded Hopf algebras on general classes of rooted trees, while investigating data structures to efficiently compute certain differential operators. The most well-known case of the Grossman–Larson (GL) algebras, henceforth denoted the *unordered GL Hopf algebra*, is based on defining a product of unordered trees by a grafting operation, and a coproduct as the dual of the free commutative product or commutative concatenation of trees. Another version, denoted *ordered GL Hopf algebra*, has grafting of ordered trees as the product and the dual of the shuffle product as the coproduct, and will be shown to be the dual of the Hopf algebra \mathcal{H}_N of this paper. It was shown, by Foissy [16] and Hoffman [19], that the dual of the commutative Hopf algebra of Butcher and of Connes and Kreimer is the unordered GL Hopf algebra, which corrected the original result of Panaite [35]. Foissy [15], [16] introduced a Hopf algebra of ordered (planar) trees \mathcal{H}_P which uses the noncommutative concatenation as the product, this generalized the Butcher–Connes–Kreimer Hopf algebra. Murua [31], [32] has developed series expansions of elementary differential operators and shown, among other results, that the logarithm of such a series is equivalent to the series expansions obtained from backward error analysis.

Recently, a great deal of interest has been focused on developing numerical methods which preserve geometric properties of the exact flow. In particular, Lie group integrators, which describe integrators that use Lie group actions on manifolds, were originally proposed by Crouch and Grossman [13] followed shortly

after by Lewis and Simo [22], [23]. Integrators of this type are now known as Lie group integrators; a survey of these methods is given in [20]. Series expansions for various classes of Lie group methods have been developed; these expansions are generally used to analyze order. Munthe-Kaas [25], [26] constructed the order conditions for a special subclass of Lie group methods, where the computations are performed in a Lie algebra. Later, in [27] it was shown that the classical order conditions could be used along with a certain transformation. Owren and Marthinsen [34] developed the general order conditions for the Crouch–Grossman methods with their analysis being based on ordered rooted trees. Recently, Owren [33] derived the order conditions for the commutator free Lie group methods [8], which were derived to overcome some of the problems associated with computing commutators.

In this paper we study a commutative graded Hopf algebraic structure arising in the order theory and backward error analysis of Lie group methods, as in [2]. The outline of this paper is as follows: In Section 2 we will introduce ordered trees and forests and describe some useful operations on them. We will motivate the present Hopf algebra as a universal object in a category of noncommutative derivations, and also briefly discuss Lie–Butcher theory which will be treated in more detail in [30]. In Section 3 we develop the Hopf algebra of ordered trees, giving both recursive and nonrecursive definitions of the coproduct and antipode, using certain cutting operations on ordered rooted trees. We show that the dual of the Hopf algebra described in this paper is the ordered GL Hopf algebra, thus generalizing the result of Foissy [15] and Hoffman [19]. Finally, in Section 4 we use a symmetrization operator to provide an injective Hopf algebra homomorphism which establishes \mathcal{H}_C as a Hopf subalgebra of \mathcal{H}_N and relates \mathcal{H}_N to the Hopf algebra of Foissy.

2. Algebras of Noncommutative Derivations

2.1. An Algebra of Trees

In this section we will define an algebra N spanned by forests of ordered (and possibly coloured) rooted trees. This algebra is a universal (‘free’) object in a general category of noncommutative derivation algebras, and plays a role in symbolic computing with Lie–Butcher series similar to the role of *free Lie algebras* [29], [36] in symbolic computing with Lie algebras.

Let OT denote the set of *ordered coloured rooted trees*, and let OF denote the (empty and nonempty) words over the alphabet OT, henceforth called the set of empty and nonempty *forests*. It should be noted that, unlike the classical Butcher theory, the ordering of the branches in the trees in OT is important, and likewise the ordering of the trees within the forest OF.

The basic operations involved in building OT and OF are:

- Create the empty forest $\mathbb{I} \in \text{OF}$.

- Create a longer forest from shorter forests by the noncommutative concatenation product, $(\omega_1, \omega_2) \mapsto \omega_1\omega_2$. The unit in the concatenation product is \mathbb{I} , where $(\omega, \mathbb{I}) = (\mathbb{I}, \omega) \mapsto \omega$.
- Create a tree from a forest by adding a root node, $B^+ : \text{OF} \rightarrow \text{OT}$. In the instance where we wish to colour the nodes using a set of colours \mathcal{I} , we introduce an indexed family of root-adding operations $B_i^+ : \text{OF} \rightarrow \text{OT}$ for all $i \in \mathcal{I}$. The inverse operation whereby we create a forest from a tree by removing the root node is written $B^- : \text{OT} \rightarrow \text{OF}$. This operation extends to OF by $B^-(\omega_1\omega_2) = B^-(\omega_1)B^-(\omega_2)$ and $B^-(\mathbb{I}) = \mathbb{I}$. Note $B^+(\mathbb{I}) = \bullet$ and $B^-(\bullet) = \mathbb{I}$.

The total number of forests, with n nodes coloured in i different ways, is defined by modifying the definition of the well-known Catalan numbers

$$C_i^n = \frac{i^n}{(n+1)} \binom{2n}{n}, \quad n = 0, 1, 2, \dots$$

See A000108 in [37] for various combinatorial representations of the Catalan numbers. For a forest $\omega \in \text{OF}$ we define the *degree*, $\#(\omega)$, as the number of trees in ω as

$$\begin{aligned} \#(\mathbb{I}) &= 0, \\ \#(B^+(\omega)) &= 1, \\ \#(\omega_1\omega_2) &= \#(\omega_1) + \#(\omega_2), \end{aligned}$$

and the *order*, $|\omega|$, as the total number of nodes in all the trees of ω as

$$\begin{aligned} |\mathbb{I}| &= 0, \\ |B^+(\omega)| &= 1 + |\omega|, \\ |\omega_1\omega_2| &= |\omega_1| + |\omega_2|. \end{aligned}$$

We let $N = \mathbb{R}\langle \text{OF} \rangle$ denote the linear space of all finite \mathbb{R} -linear combinations of elements in OF . This vector space is naturally equipped with an inner product such that all forests are orthogonal,

$$\langle \omega_1, \omega_2 \rangle = \begin{cases} 1, & \text{if } \omega_1 = \omega_2, \\ 0, & \text{otherwise,} \end{cases} \quad \text{for all } \omega_1, \omega_2 \in \text{OF}. \quad (1)$$

For $a \in N$ and $\omega \in \text{OF}$, we let $a(\omega) \in \mathbb{R}$ denote the coefficient of the forest ω , thus a can be written as a sum

$$a = \sum_{\omega \in \text{OF}} a(\omega)\omega,$$

where all but a finite number of terms are zero. The space of all infinite sums of this kind is denoted N^* and is the dual space of N , that is,

$$N^* = \{\alpha : N \rightarrow \mathbb{R} : \alpha \text{ linear}\}.$$

We let $\alpha(a)$ denote the value of $\alpha \in N^*$ on $a \in N$. All the operations on N extend to N^* by local finiteness, see [36]. For example, the inner product in (1) is extended to a dual pairing $\langle \cdot, \cdot \rangle : N^* \times N \rightarrow \mathbb{R}$, where the computation $\langle \alpha, b \rangle = \sum_{\omega \in \text{OF}} \alpha(\omega)b(\omega)$ is always finite, since b is required to be finite.

The operations B_i^+ , B^- and concatenation extend to N by linearity and the distributive law of concatenation, that is,

$$B_i^+(\omega_1 + \omega_2) = B_i^+(\omega_1) + B_i^+(\omega_2),$$

$$\omega(\omega_1 + \omega_2) = \omega\omega_1 + \omega\omega_2.$$

The vector space N with the concatenation product has two natural gradings. The first uses the number of trees $\#$ to grade $N = \bigoplus_{j \in \mathbb{Z}} N_j$, where N_j denotes the linear combination of forests with j trees. The second uses the number of nodes $|\cdot|$ to grade $N = \bigoplus_{j \in \mathbb{Z}} \tilde{N}_j$, where \tilde{N}_j denotes the linear combination of forests with the same number of nodes.

Now we introduce a *left grafting* which has the algebraic structure of a derivation.

Definition 1. For $a_1, a_2 \in N$, define the left grafting $a_1[a_2] \in N$ by the following recursion formulae, where $\tau \in \text{OT}$ and $\omega, \omega_1, \omega_2 \in \text{OF}$:

$$\tau[\mathbb{I}] = 0,$$

$$\tau[\omega_1\omega_2] = (\tau[\omega_1])\omega_2 + \omega_1(\tau[\omega_2]), \tag{2a}$$

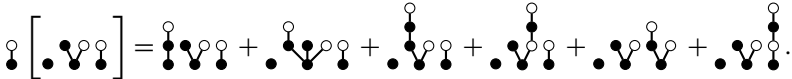
$$\tau[B_i^+(\omega)] = B_i^+(\tau[\omega]) + B_i^+(\tau\omega),$$

$$\mathbb{I}[a_2] = a_2,$$

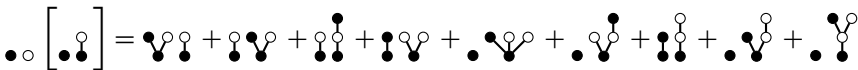
$$(\tau\omega)[a_2] = \tau[\omega[a_2]] - (\tau[\omega])[a_2]. \tag{2b}$$

The definition of left grafting is extended to the general case $a_1[a_2]$ by bilinearity.

The left grafting does not form a unital algebra as there is no consistent unit $\tau[\mathbb{I}] \neq \mathbb{I}[\tau] = \tau$. It is not associative either as $\tau_1[\tau_2[\tau_3]] \neq (\tau_1[\tau_2])[\tau_3]$. It does however satisfy the relation $\#(a_1[a_2]) = \#(a_2)$. It is useful to understand left grafting directly rather than via the recursive definition. From (2a) we verify that if $\tau \in \text{OT}$ and $\omega \in \text{OF}$, then $\tau[\omega]$ is a sum of $|\omega|$ words, each word obtained by attaching the root of τ with an edge to the left side of a node of ω ,



From (2b) we see that if $\tau_1, \tau_2 \in \text{OT}$ and $\omega \in \text{OF}$, then $(\tau_1\tau_2)[\omega]$ is obtained by first left grafting τ_2 to all nodes of ω and then left grafting τ_1 to all the nodes of the resulting expression, except to the nodes coming originally from τ_2 ,



Lemma 1. *If $\tau_1, \dots, \tau_k \in \text{OT}$ and $\omega \in \text{OF}$, then $(\tau_1 \cdots \tau_k)[\omega]$ is a sum of $|\omega|^k$ words obtained by, in the order $j = k, k - 1, \dots, 1$, attaching the root of the tree τ_j with an edge to the left side of any node in ω . In particular, we have*

$$\omega[B_i^+(\mathbb{I})] = B_i^+(\omega), \quad \text{for all } \omega \in \text{OF}. \tag{3}$$

Equations (2a) and (2b) imply that for any $d \in N_1$ and $a, b \in N$, we have the Leibniz rule and a composition rule of the form

$$d[ab] = (d[a])b + a(d[b]), \tag{4}$$

$$d[a[b]] = (da)[b] + (d[a])[b]. \tag{5}$$

Thus $d \in N_1$ acts as a first degree derivation on N .

Definition 2. The ordered Grossman–Larson (GL) product $\circ : N \otimes N \rightarrow N$ is defined as

$$\omega_1 \circ \omega_2 = B^-(\omega_1[B_i^+(\omega_2)]), \quad \text{for all } \omega_1, \omega_2 \in \text{OF},$$

and is extended to the general $a_1 \circ a_2$ for $a_1, a_2 \in N$ by linearity.

The ordered GL product can be understood by adding an invisible root to ω_2 (turning it into a tree), and left grafting ω_1 onto all nodes of $B_i^+(\omega_2)$, including the invisible root. The root is then removed from each of the resulting trees, with the ordered GL product resulting in a total of $(|\omega_2| + 1)^{\#(\omega_1)}$ forests. Some examples of the left grafting product from Definition 1 and the ordered GL product from Definition 2 are given in Table 1. Grossman and Larson define their ordered product essentially¹ as

$$\omega_1 \circ \omega_2 = B^-\left(\sum_d \omega_1 \xrightarrow{d} B^+(\omega_2)\right),$$

where \xrightarrow{d} denotes an attachment of each of the trees in the left argument to one of the nodes in the right argument, preserving the order of the left-hand trees. The sum runs over all possible attachments.

Lemma 2. *The ordered GL product is an associative $\#$ -graded product, $\circ : N_j \otimes N_k \rightarrow N_{j+k}$, and an $|\cdot|$ -graded product, $\circ : \tilde{N}_j \otimes \tilde{N}_k \rightarrow \tilde{N}_{j+k}$ satisfying, for all $a, b, c \in N$,*

$$a \circ \mathbb{I} = \mathbb{I} \circ a, \tag{6}$$

$$(a \circ b)[c] = a[b[c]]. \tag{7}$$

¹ The theory of Grossman and Larson [17] is formulated on trees, not on forests of trees. To a forest ω in our terminology, they add a (proper) root to turn it into a tree. The definition of the ordered GL product is modified accordingly.

Table 1. Examples of the left grafting, from Definition 1, and the ordered GL product, from Definition 2, for all forests up to order 4. The ordered GL product is the dual of the coproduct in \mathcal{H}_N described in Section 3.

$\omega_1 \otimes \omega_2$	$\omega_1[\omega_2]$	$\omega_1 \circ \omega_2$

Proof. Associativity is proven in [17]. The grading properties and (6) are evident. The last relation (7) is proven by induction in the degree of the leftmost argument, $\#(a)$. If $a = \tau \in N_1$ we find

$$\begin{aligned} (\tau \circ b)[c] &= B^-(\tau[B_i^+(b)])[c] = B^-(B_i^+(\tau[b]) + B_i^+(\tau b))[c] \\ &= (\tau[b] + (\tau b))[c] = \tau[b[c]]. \end{aligned}$$

Thus (2b) becomes $\tau\omega = \tau \circ \omega - \tau[\omega]$. Assuming (7) holds for $a \in N_k$ we find for $a = \tau\omega \in N_{k+1}$ that

$$((\tau\omega) \circ b)[c] = ((\tau \circ \omega - \tau[\omega]) \circ b)[c] = (\tau \circ \omega - \tau[\omega])[b[c]] = (\tau\omega)[b[c]].$$

Thus (7) holds for a general $a \in N_{k+1}$ by linearity. □

2.2. N as a Universal Object

Definition 3. Let D be an associative \mathbb{Z} -graded algebra $D = \bigoplus_{j=0}^{\infty} D_j$ with associative product $(a, b) \mapsto ab$, a unit $\mathbb{1}$ and grading $\#(D_j) = j$ such that $\#(D_j D_k) = j + k$. We call D a D -algebra if it is also equipped with a linear derivation $(\cdot)[\cdot] : D \otimes D \rightarrow D$ such that (4) and (5) hold for any $d \in D_1$ and any $a, b \in D$.

Define a D -algebra homomorphism as a linear degree-preserving map \mathcal{F} between D -algebras satisfying, for any $a, b \in D$,

$$\mathcal{F}(ab) = \mathcal{F}(a)\mathcal{F}(b), \tag{8}$$

$$\mathcal{F}(a[b]) = \mathcal{F}(a)[\mathcal{F}(b)]. \tag{9}$$

Proposition 1. Let N be the algebra of forests coloured with a set \mathcal{I} . For any D -algebra D and any map $i \mapsto f_i : \mathcal{I} \rightarrow D_1 \subset D$, there exists a unique homomorphism $\mathcal{F} : N \rightarrow D$ such that $\mathcal{F}(B_i^+(\mathbb{1})) = f_i$.

Proof. From (3) and (9) we find $\mathcal{F}(B_i^+(\omega)) = \mathcal{F}(\omega)[f_i]$ for any $\omega \in \text{OF}$. Together with (8) and linearity, this shows that, by recursion, we can extend \mathcal{F} to a uniquely defined homomorphism defined on all of N . □

This shows that N is a universal object, free over the set \mathcal{I} , in the category of D -algebras.

2.3. The Algebra of \mathfrak{G} -Sections on a Manifold

As an example of a D -algebra, we consider an algebra related to the numerical Lie group integrators. Let \mathfrak{g} be a Lie algebra of vector fields on a manifold \mathcal{M} and

let $\exp : \mathfrak{g} \rightarrow \text{Diff}(\mathcal{M})$ denote the flow operator. A basic assumption of numerical Lie group integrators [20], [27], [34] is the existence of a \mathfrak{g} which is transitive (i.e. spans all tangent directions in any point on \mathcal{M}), and for which the exponential map can be computed efficiently and exactly. Transitivity implies that *any* vector field can be written in terms of a function $f : \mathcal{M} \rightarrow \mathfrak{g}$. The goal of numerical Lie group integrators is to approximate the flow of a general differential equation

$$y'(t) = f(y)(y), \quad \text{where } f : \mathcal{M} \rightarrow \mathfrak{g}, \quad (10)$$

by composing exponentials of elements in \mathfrak{g} . The study of order conditions for Lie group integrators leads to a need for understanding the algebraic structure of noncommuting vector fields on \mathcal{M} , generated from f .

Elements $V \in \mathfrak{g}$ are often called invariant or ‘frozen’ vector fields on \mathcal{M} . These define first-degree invariant differential operators through the Lie derivative. Let \mathcal{V} be any normed vector space and denote by $C^\infty(\mathcal{M}, \mathcal{V})$ the set of all smooth functions from \mathcal{M} to \mathcal{V} , called the space of \mathcal{V} -sections. For $V \in \mathfrak{g}$ and $\psi \in C^\infty(\mathcal{M}, \mathcal{V})$, the Lie derivative, $V[\psi] \in C^\infty(\mathcal{M}, \mathcal{V})$, is defined as

$$V[\psi](p) = \left. \frac{d}{dt} \right|_{t=0} \psi(\exp(tV)(p)), \quad \text{for any point } p \in \mathcal{M}.$$

For two elements $V, W \in \mathfrak{g}$ we iterate this definition and define the concatenation VW as the second-degree invariant differential operator $VW[\psi] = V[W[\psi]]$. The linear space spanned by the 0-degree identity operator $\mathbb{I}[\psi] = \psi$ and all higher-degree invariant derivations is called the *universal enveloping algebra* of \mathfrak{g} , denoted \mathfrak{G} . This is a graded algebra with the noncommutative concatenation product and degree $\#(\mathbb{I}) = 0$, $\#(\mathfrak{g}) = 1$ and $\#(VW) = \#(V) + \#(W)$.

Given a norm on the vector space \mathfrak{G} , we consider the space of \mathfrak{G} -sections² $C^\infty(\mathcal{M}, \mathfrak{G})$. For two sections $f, g \in C^\infty(\mathcal{M}, \mathfrak{G})$ we define $f[g] \in C^\infty(\mathcal{M}, \mathfrak{G})$ pointwise from the Lie derivative as

$$f[g](p) = (f(p)[g])(p), \quad p \in \mathcal{M}.$$

Similarly, the concatenation on \mathfrak{G} is extended pointwise to a concatenation $fg \in C^\infty(\mathcal{M}, \mathfrak{G})$ as

$$(fg)(p) = f(p)g(p), \quad p \in \mathcal{M}.$$

From these definitions we find:

Lemma 3. *Let $f \in C^\infty(\mathcal{M}, \mathfrak{g})$ and let $g, h \in C^\infty(\mathcal{M}, \mathfrak{G})$. Then*

$$\begin{aligned} f[gh] &= f[g]h + gf[h], \\ (f \circ g)[h] &\equiv f[g[h]] = fg[h] + f[g][h]. \end{aligned}$$

² Thus \mathfrak{G} is a trivial vector bundle over \mathcal{M} , \mathfrak{g} a trivial subbundle and the tangent bundle $T\mathcal{M}$ is a nontrivial subbundle of \mathfrak{g} .

Proof. For $p \in \mathcal{M}$ let $V = f(p) \in \mathfrak{g}$. Then

$$\begin{aligned}
 f[gh](p) &= \left. \frac{d}{dt} \right|_{t=0} (gh)(\exp(tV)(p)) \\
 &= \left. \frac{d}{dt} \right|_{t=0} g(\exp(tV)(p))h(\exp(tV)(p)) \\
 &= \left. \frac{d}{dt} \right|_{t=0} g(\exp(tV)(p))h(p) + g(p)h(\exp(tV)(p)) \\
 &= (f[g]h + gf[h])(p), \\
 (f[g[h]])(p) &= \left. \frac{d}{dt} \right|_{t=0} (g(\exp(tV)(p))[h])(\exp(tV)(p)) \\
 &= \left. \frac{d}{dt} \right|_{t=0} g(p)[h](\exp(tV)(p)) + g(\exp(tV)(p))[h](p) \\
 &= (fg)[h](p) + (f[g])[h](p). \quad \square
 \end{aligned}$$

Note the difference between fg and $f \circ g$. In the concatenation the value of g is frozen to $g(p)$ before the differentiation with f is done, whereas in the latter case the spatial variation of g is seen by the differentiation using f .

Lemma 3 shows that $C^\infty(\mathcal{M}, \mathfrak{G})$ is a D -algebra. Thus if we, for every $i \in \mathcal{I}$, pick a vector field $f_i \in C^\infty(\mathcal{M}, \mathfrak{g})$, then there exists a unique homomorphism $\mathcal{F} : N \rightarrow C^\infty(\mathcal{M}, \mathfrak{G})$ such that $\mathcal{F}(B_i^+(\mathbb{I})) = f_i$. The images of the trees $\mathcal{F}(\tau)$, for $\tau \in \text{OT}$, are called the *elementary differentials* in Butcher’s theory (see [5]) and the images of the forests $\mathcal{F}(\omega)$, for $\omega \in \text{OF}$, are called *elementary differential operators* in Merson’s theory (see [24]).

2.4. Elements of Lie–Butcher Theory

To motivate the algebraic structures of the next section, we briefly introduce some elements of Lie–Butcher theory. This theory is the noncommutative generalization of the classical Butcher theory and is the general foundation behind the construction of order conditions for Lie group integrators. Various aspects of this theory have been developed in [2], [25], [26], [28], [34]. A comprehensive treatment is given in [30].

Consider the homomorphism \mathcal{F} introduced in Section 2.3 extended to a homomorphism of infinite series $\mathcal{F} : N^* \rightarrow C^\infty(\mathcal{M}, \mathfrak{G})$, where $C^\infty(\mathcal{M}, \mathfrak{G})$ should now be understood as a space of formal series. The series might not converge, but all definitions make sense termwise, and any finite truncation yields a proper \mathfrak{G} -section. In classical (commutative) Butcher theory the image of $\alpha \in N^*$ is called an S-series, see Murua [31]. Similarly, we define an LS-series as an infinite formal

series in $C^\infty(\mathcal{M}, \mathfrak{G})$ given by³

$$\text{LS}(\alpha) = \sum_{\omega \in \text{OF}} h^{|\omega|} \alpha(\omega) \mathcal{F}(\omega). \tag{11}$$

Classical Lie series on manifolds is a generalization of Taylor series, where the fundamental result is the following ‘pull-back formula’: Let $f \in C^\infty(\mathcal{M}, \mathfrak{g})$ be a vector field and let $\exp(f) : \mathcal{M} \rightarrow \mathcal{M}$ be the $t = 1$ -flow. For any $g \in C^\infty(\mathcal{M}, \mathfrak{g})$ we have (see [1]) that

$$g(\exp(f)(p)) = \sum_{j=0}^{\infty} \frac{1}{j!} f^j[g](p) \equiv \text{Exp}(f)[g](p),$$

where $f^0 = \mathbb{I}$ and $f^j[g] = f[\dots f[f[g]]\dots] = (f \circ \dots \circ f)[g]$. Note that if $f = \mathcal{F}(\bullet)$, then $f^j = \mathcal{F}(\bullet \circ \dots \circ \bullet)$, thus the operator exponential $\text{Exp}(f) = \sum_{j=0}^{\infty} f^j/j!$ is an LS-series.

Two special cases of LS-series are of particular importance: An LS-series $\text{LS}(\alpha)$ is called *logarithmic* or *algebra-like* if $\text{LS}(\alpha) \in C^\infty(\mathcal{M}, \mathfrak{g})$ represents a vector field, and it is called *exponential* or *group-like* if $\text{LS}(\alpha)$ is the (formal) operator exponential of a logarithmic series.

Note that if $\tau_1, \tau_2 \in \text{OT}$, then $a = \tau_1\tau_2 - \tau_2\tau_1$ is a logarithmic series since it represents the commutator of two vector fields. More generally, a series $\alpha \in N^*$ is logarithmic if, and only if, all its finite components belong to the free Lie algebra generated by OT . A Hall basis for this space is characterized in [28]. Reutenauer [36] presents several alternative characterizations of logarithmic and exponential series.

We find the characterization in terms of *shuffle products* particularly useful. The shuffle product $\sqcup : N \otimes N \rightarrow N$ is defined for two forests as the summation over all permutations of the trees in the forests while preserving the ordering of the trees in each of the initial forests, and is extended to $N \otimes N$ by linearity. It can also be recursively defined in the asymmetric way $\mathbb{I} \sqcup \omega = \omega \sqcup \mathbb{I} = \omega$ for any forest $\omega \in \text{OF}$, and if $\omega_1 = v_1\tau_1$ and $\omega_2 = v_2\tau_2$ for $\tau_1, \tau_2 \in \text{OT}$ and $v_1, v_2 \in \text{OF}$, then

$$(v_1\tau_1) \sqcup (v_2\tau_2) = (v_1 \sqcup (v_2\tau_2))\tau_1 + ((v_1\tau_1) \sqcup v_2)\tau_2. \tag{12}$$

The shuffle product is associative and commutative, for all $\omega_1, \omega_2, \omega_3 \in \text{OF}$ we have

$$\begin{aligned} (\omega_1 \sqcup \omega_2) \sqcup \omega_3 &= \omega_1 \sqcup (\omega_2 \sqcup \omega_3), \\ \omega_1 \sqcup \omega_2 &= \omega_2 \sqcup \omega_1. \end{aligned}$$

Table 2 gives some simple, but nontrivial, examples of the shuffle product.

³ See the comments at the end of Section 4.1 on the chosen normalization.

Table 2. All nontrivial examples of the shuffle product μ_N , defined recursively in (12), for all forests up to and including order 4. These are the dual of the coproduct in the ordered GL Hopf algebra.

$\omega_1 \otimes \omega_2$	$\mu_N(\omega_1 \otimes \omega_2)$
	2
	3
	+
	4
	2 +
	2 +
	+ +
	+ +
	6
	+ +
	2 +

Lemma 4 [36]. A series $\alpha \in N^*$ is logarithmic if and only if

$$\alpha(\mathbb{I}) = 0,$$

$$\alpha(\omega_1 \sqcup \omega_2) = 0, \quad \text{for all } \omega_1, \omega_2 \in \text{OF} \setminus \{\mathbb{I}\}.$$

A series $\alpha \in N^*$ is exponential if and only if

$$\alpha(\mathbb{I}) = 1,$$

$$\alpha(\omega_1 \sqcup \omega_2) = \alpha(\omega_1)\alpha(\omega_2), \quad \text{for all } \omega_1, \omega_2 \in \text{OF}.$$

The LS-series of an exponential series $\alpha \in N^*$ represents pull-backs, or finite motions on \mathcal{M} . They form a group under composition with the ordered GL product. This is a generalization of the *Butcher group* to the case of noncommutative group actions. Computing the product and the inverse in this group is facilitated by the *coproduct* Δ_N and the *antipode* S_N introduced in the next section, see the discussion after Corollary 1.

3. Hopf Algebras

In this section we will study a commutative graded Hopf algebra \mathcal{H}_N of ordered forests. The coproduct Δ_N and the antipode S_N in the Hopf algebra are defined

by recursive and nonrecursive formulae. We also show that the coproduct Δ_N is the dual of the ordered GL product introduced in Definition 2 of the previous section.

We begin by briefly reviewing the definition of a Hopf algebra, see [38] for details. A real associative algebra \mathcal{A} is a real vector space with an associative product $\mu : \mathcal{A} \otimes \mathcal{A} \rightarrow \mathcal{A}$ and a unit $u : \mathbb{R} \rightarrow \mathcal{A}$ such that $\mu(a \otimes u(1)) = \mu(u(1) \otimes a) = a$ for all $a \in \mathcal{A}$. The dual of an algebra is called a coalgebra, \mathcal{C} , which is a vector space equipped with a coassociative coproduct $\Delta : \mathcal{C} \rightarrow \mathcal{C} \otimes \mathcal{C}$ and counit $e : \mathcal{C} \rightarrow \mathbb{R}$. A bialgebra \mathcal{B} is a linear space which has both an algebra and also a coalgebra structure such that the coproduct and the counit are compatible with the product, in the sense that

$$e(\mu(\omega_1 \otimes \omega_2)) = \mu(e(\omega_1) \otimes e(\omega_2)), \tag{13}$$

$$\Delta(\mu(\omega_1 \otimes \omega_2)) = (\mu \otimes \mu)(I \otimes T \otimes I)(\Delta(\omega_1) \otimes \Delta(\omega_2)), \tag{14}$$

where $T(\omega_1 \otimes \omega_2) = \omega_2 \otimes \omega_1$ is the twist map. Let $\text{End}(\mathcal{B})$ denote all linear maps from \mathcal{B} to itself. We define the convolution $\star : \text{End}(\mathcal{B}) \otimes \text{End}(\mathcal{B}) \rightarrow \text{End}(\mathcal{B})$ as

$$(A \star B)(a) = \mu((A \otimes B)\Delta(a)), \quad \text{for } A, B \in \text{End}(\mathcal{B}) \text{ and } a \in \mathcal{B}. \tag{15}$$

Let $I \in \text{End}(\mathcal{B})$ denote the identity matrix. An antipode is a linear map $S \in \text{End}(\mathcal{B})$, which is the two-sided inverse of the identity matrix under convolution, with the antipode satisfying

$$(I \star S)(a) = (S \star I)(a) = u(e(a)), \quad \text{for all } a \in \mathcal{B}. \tag{16}$$

Definition 4. A Hopf algebra \mathcal{H} is a bialgebra equipped with an antipode.

3.1. The Hopf Algebra of Ordered Trees

We will study a particular Hopf algebra based on the vector space of ordered forests $N = \mathbb{R}\langle \text{OF} \rangle$, where the coproduct is defined by the following recursion.

Definition 5. Let $\Delta_N : N \rightarrow N \otimes N$ be defined by linearity and the recursion

$$\begin{aligned} \Delta_N(\mathbb{I}) &= \mathbb{I} \otimes \mathbb{I}, \\ \Delta_N(\omega\tau) &= \omega\tau \otimes \mathbb{I} + \Delta_N(\omega) \sqcup \cdot (I \otimes B_i^+) \Delta_N(B^-(\tau)), \end{aligned} \tag{17}$$

where $\tau = B_i^+(\omega_1) \in \text{OT}$ and $\omega, \omega_1 \in \text{OF}$. The linear operation $\sqcup \cdot : N \otimes N \otimes$

$N \otimes N \rightarrow N \otimes N$ is a shuffle on the left and concatenation on the right, satisfying

$$(\omega_1 \otimes \tau_1) \sqcup \cdot (\omega_2 \otimes \tau_2) = (\omega_1 \sqcup \omega_2) \otimes (\tau_1 \tau_2).$$

Note that letting $\omega = \mathbb{I}$ yields the special recursion formula for a tree τ :

$$\Delta_N(\tau) = \tau \otimes \mathbb{I} + (I \otimes B_i^+) \Delta_N(B^-(\tau)).$$

This expression for the coproduct of a tree is similar to the coproduct Δ_C in Connes and Kreimer [10] and Δ_P in Foissy [16], but differences arise from different rules for forests. We illustrate this with an example. Compare the Foissy coproduct

$$\begin{aligned} \Delta_P \left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} \right) &= \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} \otimes \mathbb{I} + \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} \otimes \bullet + \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} \otimes \bullet + \begin{array}{c} \bullet \\ \bullet \end{array} \otimes \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \bullet \end{array} \otimes \begin{array}{c} \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \bullet \end{array} \otimes \begin{array}{c} \bullet \\ \bullet \end{array} + \mathbb{I} \otimes \begin{array}{c} \bullet \\ \bullet \end{array} \\ \Delta_P \left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} \right) &= \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} \otimes \mathbb{I} + \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} \otimes \bullet + \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} \otimes \bullet + \begin{array}{c} \bullet \\ \bullet \end{array} \otimes \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \bullet \end{array} \otimes \begin{array}{c} \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \bullet \end{array} \otimes \begin{array}{c} \bullet \\ \bullet \end{array} + \mathbb{I} \otimes \begin{array}{c} \bullet \\ \bullet \end{array} \end{aligned}$$

with Δ_N listed in Table 5. Whereas Δ_P commutes with branch permutations, and treats different permutations of a given tree equivalently; different permutations are treated essentially differently by Δ_N . The coproduct Δ_C of the Butcher–Connes–Kreimer Hopf algebra is the same as the coproduct Δ_P in the Foissy Hopf algebra, forgetting the order of the branches. The relationship is elaborated in Section 4, where we show that

$$\begin{aligned} \Delta_N \left(\Omega \left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} \right) \right) &= \Delta_N \left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} + \begin{array}{c} \bullet \\ \diagdown \quad \diagup \\ \bullet \quad \bullet \end{array} \right) = (\Omega \otimes \Omega) \Delta_P \left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} \right) = (\Omega \otimes \Omega) \Delta_C \left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} \right) \\ &= \left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} + \begin{array}{c} \bullet \\ \diagdown \quad \diagup \\ \bullet \quad \bullet \end{array} \right) \otimes \mathbb{I} + \left(\begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} \right) \otimes \bullet + 2 \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} \otimes \bullet + \begin{array}{c} \bullet \\ \bullet \end{array} \otimes \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} \\ &\quad + 2 \begin{array}{c} \bullet \\ \bullet \end{array} \otimes \begin{array}{c} \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \bullet \end{array} \otimes \begin{array}{c} \bullet \\ \bullet \end{array} + \mathbb{I} \otimes \left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} + \begin{array}{c} \bullet \\ \diagdown \quad \diagup \\ \bullet \quad \bullet \end{array} \right), \end{aligned}$$

where the symmetrization operator Ω is defined in Definition 9. A further significant difference between these three Hopf algebras is the product. The Butcher–Connes–Kreimer Hopf algebra is based on the free commutative product or commutative concatenation, Foissy on the free associative product or noncommutative concatenation and \mathcal{H}_N on the shuffle product. These two differences result in a difference in the antipodes of each of the three Hopf algebras. We again illustrate this with an example. Compare the Foissy antipode

$$\begin{aligned} S_P \left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} \right) &= - \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} + \begin{array}{c} \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} - \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} - 2 \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} + \dots, \\ \Delta_P \left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} \right) &= - \begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} + \begin{array}{c} \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} - \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} - 2 \begin{array}{c} \bullet \quad \bullet \\ \bullet \end{array} + \dots, \end{aligned}$$

with S_N listed in Table 6. Again S_P commutes with branch permutations, and treats different permutations of a given tree equivalently, that is, forgetting the ordering of branches and the ordering of trees within a forest implies that S_C and S_P are equivalent, whereas S_N is different. An example of the connection between the three antipodes is

$$\begin{aligned}
 S_N\left(\Omega\left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \end{array}\right)\right) &= S_N\left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \diagdown \quad \diagup \\ \bullet \end{array}\right) = \Omega\left(S_P\left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \end{array}\right)\right) = \Omega\left(\Delta_C\left(\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \end{array}\right)\right) \\
 &= -\begin{array}{c} \bullet \\ \diagup \quad \diagdown \\ \bullet \end{array} - \begin{array}{c} \bullet \\ \diagdown \quad \diagup \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} + \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} + 2 \cdot \begin{array}{c} \bullet \\ \bullet \quad \bullet \end{array} + 2 \begin{array}{c} \bullet \\ \bullet \quad \bullet \end{array} + 2 \begin{array}{c} \bullet \\ \bullet \quad \bullet \end{array} \\
 &\quad - 2 \cdot \begin{array}{c} \bullet \\ \bullet \quad \bullet \end{array} - 2 \cdot \begin{array}{c} \bullet \\ \bullet \quad \bullet \end{array} - 2 \begin{array}{c} \bullet \\ \bullet \quad \bullet \end{array} + 24 \cdot \begin{array}{c} \bullet \\ \bullet \quad \bullet \end{array}
 \end{aligned}$$

We explain in detail these connections in Section 4.

Theorem 1. *Let \mathcal{H}_N be the vector space $N = \mathbb{R}\langle \text{OF} \rangle$ with the operations*

$$\begin{aligned}
 \text{product: } \mu_N(a \otimes b) &= a \sqcup b, && \text{(Shuffle product)} \\
 \text{coproduct: } \Delta_N, &&& \text{(Definition 5)} \\
 \text{unit: } u_N(1) &= \mathbb{I}, \\
 \text{counit: } e_N(\omega) &= \begin{cases} 1, & \text{if } \omega = \mathbb{I}, \\ 0, & \text{otherwise.} \end{cases}
 \end{aligned}$$

Then \mathcal{H}_N is a Hopf algebra with an antipode S_N given by the recursion

$$\begin{aligned}
 S_N(\mathbb{I}) &= \mathbb{I}, \\
 S_N(\omega\tau) &= -\mu_N((S_N \otimes I)(\Delta_N(\omega) \sqcup \cdot (I \otimes B_i^+) \Delta_N(B^-(\tau)))), \tag{18}
 \end{aligned}$$

where $\tau = B_i^+(\omega_1) \in \text{OT}$ and $\omega, \omega_1 \in \text{OF}$. In particular,

$$S_N(\tau) = -\mu_N((S_N \otimes I)(I \otimes B_i^+) \Delta_N(B^-(\tau))).$$

Proof. The coassociativity of the coalgebra will be established once we have shown that the algebra and coalgebra are compatible. From the fact that $e_N(\omega) = 0$ for all $\omega \in \text{OF} \setminus \{\mathbb{I}\}$, and that the shuffle product of two scalars is just standard multiplication, we immediately get (13). To show (14) we find it convenient to introduce the linear operation $\sqcup\sqcup : N \otimes N \otimes N \otimes N \rightarrow N \otimes N$ with the shuffle product both on the left and on the right, satisfying

$$(v_1 \otimes v_1) \sqcup\sqcup (v_2 \otimes v_2) = (v_1 \sqcup v_2) \otimes (v_1 \sqcup v_2).$$

The compatibility condition (14) is now equivalent to

$$\Delta_N(\omega_1 \sqcup \omega_2) = \Delta_N(\omega_1) \sqcup\sqcup \Delta_N(\omega_2). \tag{19}$$

To simplify the notation we use

$$\Delta_N(\omega\tau) = \omega\tau \otimes \mathbb{I} + \tilde{\Delta}_N(\omega\tau), \quad \tilde{\Delta}_N(\omega\tau) = \Delta_N(\omega) \sqcup \cdot \tilde{\Delta}_N(\tau).$$

Let $\omega_1 = v_1\tau_1$ and $\omega_2 = v_2\tau_2$ for $\tau_1, \tau_2 \in \text{OT}$ and $v_1, v_2 \in \text{OF}$, now using the recursive definition of the shuffle product and substituting the expression for the coproduct of an ordered forest yields

$$\begin{aligned} \Delta_N(v_1\tau_1 \sqcup v_2\tau_2) &= \Delta_N((v_1 \sqcup v_2\tau_2)\tau_1) + \Delta((v_1\tau_1 \sqcup v_2)\tau_2) \\ &= (\Delta_N(v_1) \sqcup \Delta_N(v_2\tau_2)) \sqcup \cdot \tilde{\Delta}_N(\tau_1) \\ &\quad + (\Delta_N(v_1\tau_1) \sqcup \Delta_N(v_2)) \sqcup \cdot \tilde{\Delta}_N(\tau_2) \\ &\quad + (v_1\tau_1 \sqcup v_2)\tau_2 \otimes \mathbb{I} + (v_1 \sqcup v_2\tau_2)\tau_1 \otimes \mathbb{I} \\ &= (\Delta_N(v_1) \sqcup \tilde{\Delta}_N(v_2\tau_2)) \sqcup \cdot \tilde{\Delta}_N(\tau_1) \\ &\quad + (\tilde{\Delta}_N(v_1\tau_1) \sqcup \Delta_N(v_2)) \sqcup \cdot \tilde{\Delta}_N(\tau_2) \\ &\quad + (\Delta_N(v_1) \sqcup v_2\tau_2 \otimes \mathbb{I}) \sqcup \cdot \tilde{\Delta}_N(\tau_1) \\ &\quad + (\Delta_N(v_2) \sqcup v_1\tau_1 \otimes \mathbb{I}) \sqcup \cdot \tilde{\Delta}_N(\tau_2) \\ &\quad + (v_1\tau_1 \sqcup v_2\tau_2) \otimes \mathbb{I} \\ &= v_1\tau_1 \sqcup v_2\tau_2 \otimes \mathbb{I} + \tilde{\Delta}_N(v_1\tau_1) \sqcup (v_2\tau_2 \otimes \mathbb{I}) \\ &\quad + \tilde{\Delta}_N(v_2\tau_2) \sqcup (v_1\tau_1 \otimes \mathbb{I}) \\ &\quad + \tilde{\Delta}_N(v_1\tau_1) \sqcup \tilde{\Delta}_N(v_2\tau_2) \\ &= \Delta_N(v_1\tau_1) \sqcup \Delta(v_2\tau_2). \end{aligned}$$

We now have the tools needed to show that the coalgebra is coassociative, which follows from the fact that the coproduct Δ_N satisfies

$$\begin{aligned} (I \otimes \Delta_N)\Delta_N(\omega_1 \sqcup \omega_2) &= (I \otimes \Delta_N)(\Delta_N(\omega_1) \sqcup \Delta_N(\omega_2)) \\ &= (I \otimes \Delta_N)\Delta_N(\omega_1) \sqcup (I \otimes \Delta_N)\Delta_N(\omega_2) \\ &= (\Delta_N \otimes I)\Delta_N(\omega_1) \sqcup (\Delta_N \otimes I)\Delta_N(\omega_2) \\ &= (\Delta_N \otimes I)\Delta_N(\omega_1 \sqcup \omega_2). \end{aligned}$$

Thus we have established the structure of a bialgebra. Substituting Definition 5 in (16) yields the recursion for the antipode (18). \square

Both the definition of the coproduct Δ_N and thus the antipode S_N are recursive and difficult to use in practice. To develop nonrecursive formulae for these, it is first necessary to define certain cutting operations.

Definition 6. For a given forest $\omega \in \text{OF}$, a *parent* is any node p with at least one branch growing from that node and the *children* are the nodes branching from p . Let p_c denote the number of children of p . Cutting off a child node equates to removing the edge connecting the child to its parent.

- A *nodal left cut* of degree c is a cut where the c leftmost children of a given parent node p are cut off. We can write a nodal left cut as $\ell_p(c)$ where $0 < c \leq p_c$. The cut splits a forest ω into two subforests, $P^{\ell_p(c)}(\omega)$ and $R^{\ell_p(c)}(\omega)$, where P is the part cut off, with the forest containing the c leftmost children of p as root nodes and R is the remaining bottom part of ω .
- A *left cut* is a collection of $0 \leq k$ nodal left cuts $\ell = \{\ell_{p_i}(c_i)\}_{i=1}^k$, where $\{p_i\}_{i=1}^k$ are distinct nodes of ω . This splits ω in k cut-off forests $\{\omega_i\}_{i=1}^k$ and a remaining forest $R^\ell(\omega)$, where ω_i is the forest containing the nodes connected to the c_i leftmost children of p_i and $R^\ell(\omega)$ is the forest of the nodes connected to the original root nodes. We define $P^\ell(\omega) \in N$ as

$$P^\ell(\omega) = \omega_1 \sqcup \omega_2 \sqcup \dots \sqcup \omega_k. \tag{20}$$

Note that the definition of a left cut includes the case $k = 0$, called the *empty cut*, where $R^\ell(\omega) = \omega$ and we define $P^\ell(\omega) = \mathbb{I}$.

- An *admissible left cut* is a left cut, containing the restriction that any path from a node in ω to the corresponding root is cut no more than once.

We denote by LC, NLC and ALC the set of all left cuts, nodal left cuts and admissible left cuts. To define the coproduct we need to slightly extend the definition of an admissible left cut, which we choose to call a *full admissible left cut*. The full admissible left cuts of $\omega \in \text{OF}$ are obtained by adding an (invisible) root node to form the tree $\tau = B_1^+(\omega) \in \text{OT}$, applying an admissible left cut on τ , and finally removing the invisible root node again. We denote by FALC the set of all full admissible left cuts. Thus $\text{FALC}(\omega) = \text{ALC}(\tau)$ and for any $\ell \in \text{FALC}(\omega)$ we have $P^\ell(\omega) = P^\ell(\tau)$ and $R^\ell(\omega) = B^-(R^\ell(\tau))$. Note that $\text{FALC}(\omega)$ contains the ‘cut everything’, where $k = 1$, $P^\ell(\omega) = \omega$ and $R^\ell(\omega) = \mathbb{I}$, as well as the empty cut with $k = 0$, $P^\ell(\omega) = \mathbb{I}$ and $R^\ell(\omega) = \omega$. It is useful to note that the order in which the cuts are performed does not affect $P^\ell(\omega)$ or $R^\ell(\omega)$. The order of the cuts is taken care of by the use of the shuffle product in the definition of $P^\ell(\omega)$ given by equation (20).

As an example, we list all the cuttings of an example tree in Table 3.

Proposition 2. *The coproduct Δ_N of \mathcal{H}_N is nonrecursively defined as*

$$\Delta_N(\omega) = \sum_{\ell \in \text{FALC}(\omega)} P^\ell(\omega) \otimes R^\ell(\omega). \tag{21}$$

Proof. To prove that the recursive definition (17) and the nonrecursive definition (21) of the coproduct are identical, an induction argument on the number of vertices

Table 3. The cuts ℓ_i of an example tree τ , where ℓ_{12} is the ‘cut everything’ full cut. Thus $\text{NLC}(\tau) = \{\ell_1, \dots, \ell_4\}$, $\text{LC}(\tau) = \{\ell_0, \dots, \ell_{11}\}$, $\text{ALC}(\tau) = \{\ell_0, \dots, \ell_5\}$ and $\text{FALC}(\tau) = \{\ell_0, \dots, \ell_5\} \cup \{\ell_{12}\}$.

i	ℓ_i	k	$P^{\ell_i}(\tau)$	$R^{\ell_i}(\tau)$	i	ℓ_i	k	$P^{\ell_i}(\tau)$	$R^{\ell_i}(\tau)$
0		0	\mathbb{I}		7		2		\bullet
1		1	\bullet		8		2		\bullet
2		1	\bullet		9		2		\bullet
3		1	$\bullet \bullet$		10		3		\bullet
4		1		\bullet	11		3		\bullet
5		2	$\bullet \sqcup \bullet$		12		1		\mathbb{I}
6		2	$\bullet \bullet \sqcup \bullet$						

is used. First recall that $\text{FALC}(B^-(\tau)) = \text{ALC}(\tau)$, and for any $\ell \in \text{FALC}(B^-(\tau))$ we have $P^\ell(B^-(\tau)) = P^\ell(\tau)$ and $R^\ell(B^-(\tau)) = B^-(R^\ell(\tau))$, this implies that

$$\sum_{j \in \text{FALC}(B^-(\tau))} P^j(B^-(\tau)) \otimes B_i^+(R^j(B^-(\tau))) = \sum_{j \in \text{ALC}(\tau)} P^j(\tau) \otimes R^j(\tau).$$

Using this fact, the coproduct now takes the form

$$\begin{aligned} \Delta_N(\omega\tau) &= \omega\tau \otimes \mathbb{I} + \left(\sum_{\ell \in \text{FALC}(\omega)} P^\ell(\omega) \otimes R^\ell(\omega) \right) \\ &\quad \sqcup \cdot \left(\sum_{j \in \text{ALC}(\tau)} P^j(\tau) \otimes R^j(\tau) \right) \\ &= \omega\tau \otimes \mathbb{I} + \sum_{\ell \in \text{FALC}(\omega)} \sum_{j \in \text{ALC}(\tau)} P^\ell(\omega) \sqcup P^j(\tau) \otimes R^\ell(\omega) R^j(\tau) \\ &= \sum_{\ell \in \text{FALC}(\omega\tau)} P^\ell(\omega\tau) \otimes R^\ell(\omega\tau). \end{aligned}$$

The last equality is true because the sum over $\ell \in \text{FALC}(\omega)$ and $j \in \text{ALC}(\tau)$ is equivalent to the sum over $\ell \in \text{FALC}(\omega\tau)$ except for the ‘cut everything’ cut which is equal to the term $\omega\tau \otimes \mathbb{I}$. \square

Corollary 1. *The dual of the coproduct Δ_N is \circ the ordered GL product, that is, for any $\omega \in N$ and $\omega_1, \omega_2 \in N^*$ we have*

$$\langle \omega_1 \circ \omega_2, \omega \rangle = \langle \omega_1 \otimes \omega_2, \Delta_N(\omega) \rangle.$$

Proof. If the sum in (21) had been over ALC instead of FALC, then the dual would have been the left grafting. To see this, we use the characterization of left grafting in Lemma 1, and observe that the nodal left cut corresponds to the dual operation of attaching a number of trees in a given order to a common node, while the shuffles in $P^\ell(\omega)$ correspond to the dual operation of attaching the forests in all possible ways to different nodes. From Definition 2, we see that when the sum is extended from ALC to FALC, then we obtain the dual of the ordered GL product. \square

As an example of the practical use of this duality, we return to the task of computing the product and the inverse in the noncommutative version of the Butcher group, that is, the exponential elements of N^* . To compute the composition $\alpha \circ \beta$ for general $\alpha, \beta \in N^*$, we use Corollary 1 and find

$$(\alpha \circ \beta)(\omega) = \langle \circ(\alpha \otimes \beta), \omega \rangle = \langle \alpha \otimes \beta, \Delta_N(\omega) \rangle = \sum_{\ell \in \text{FALC}(\omega)} \alpha(P^\ell(\omega))\beta(R^\ell(\omega)).$$

Thus, we read from Table 5 that we have as an example

$$(\alpha \circ \beta)(\bullet \circ \bullet) = \alpha(\bullet \circ \bullet)\beta(\mathbb{I}) + 2\alpha(\bullet \bullet)\beta(\bullet) + \alpha(\bullet)\beta(\bullet \circ \bullet) + \alpha(\bullet)\beta(\bullet \bullet) + \alpha(\mathbb{I})\beta(\bullet \circ \bullet).$$

The inverse in the group is found from the *antipode*, the linear map $S_N : N \rightarrow N$. It can be shown that the dual of the antipode $S_N^* : N^* \rightarrow N^*$ also defines the inverse in the group of exponential elements of N^* . Let $\alpha^{-1} = S_N^*(\alpha)$, thus,

$$\alpha^{-1}(\omega) = \langle S_N^*(\alpha), \omega \rangle = \langle \alpha, S_N(\omega) \rangle = \alpha(S_N(\omega)),$$

and it can be shown that $\alpha \circ \alpha^{-1} = \mathbb{I}$ for any exponential $\alpha \in N^*$.

To present a nonrecursive definition of the antipode, we define the reversal map $S_R : N \rightarrow N$ as

$$S_R(\mathbb{I}) = \mathbb{I},$$

$$S_R(\tau_1 \tau_2 \cdots \tau_j) = (-1)^j \tau_j \tau_{j-1} \cdots \tau_1, \quad \text{for all } \tau_1 \cdots \tau_j \in \text{OF}, \quad (22)$$

extended to N by linearity. Thus S_R is the unique antiautomorphism of the concatenation algebra which sends $\tau \mapsto -\tau$.

Proposition 3. *The antipode S_N of \mathcal{H}_N is nonrecursively defined as*

$$S_N(\omega) = S_R \left(\sum_{\ell \in \text{LC}(\omega)} P^\ell(\omega) \sqcup R^\ell(\omega) \right). \tag{23}$$

Proof. In order to prove this result, we need some results about the Hopf algebraic structures of the Free Associative Algebra (FAA) and the Shuffle Algebra (Sh), following [36]. Given an alphabet A , the FAA is the vector space formed by taking all finite linear combinations of words over A and the noncommutative concatenation as product. In our case, the alphabet is OT, the words are OF and the vector space is N . The Hopf algebra structure of the FAA has the dual of the shuffle product as coproduct. The dual of the FAA Hopf algebra we call the *Hopf algebra structure of the shuffle algebra*, denoted \mathcal{H}_{Sh} . It is obtained by taking the product $\mu_{\text{Sh}} = \mu_N$ as the shuffle product, the coproduct Δ_{Sh} defined as the dual of the concatenation product. The antipode is the map S_R defined in (22). We need a characterization of Δ_{Sh} and S_R in terms of cutting operations. For an $\omega \in \text{OF}$ let the set of Word Cuts (WC) be a simple cut ℓ which splits a word ω into two parts $\omega_1 = P^\ell(\omega)$ and $\omega_2 = R^\ell(\omega)$ such that $\omega = \omega_1\omega_2$. WC contains both the empty cut where $P^\ell(\omega) = \mathbb{I}$, $R^\ell(\omega) = \omega$ and cuts everything where $P^\ell(\omega) = \omega$, $R^\ell(\omega) = \mathbb{I}$. Note that the difference between ALC and FALC is that FALC may contain a nonempty cut from WC. A direct definition of Δ_{Sh} is

$$\Delta_{\text{Sh}}(\omega) = \sum_{\ell \in \text{WC}} P^\ell(\omega) \otimes R^\ell(\omega), \quad \text{for all } \omega \in \text{OF}.$$

From (16) we find for $\omega \in \text{OF} \setminus \{\mathbb{I}\}$ that

$$0 = (S_R \star I)(\omega) = \mu_{\text{Sh}}((S_R \otimes I)\Delta_{\text{Sh}}(\omega)) = \sum_{\ell \in \text{WC}(\omega)} S_R(P^\ell(\omega)) \sqcup R^\ell(\omega).$$

Thus we find a recursive definition of the antipode S_R ,

$$\begin{aligned} S_R(\mathbb{I}) &= \mathbb{I}, \\ S_R(\omega) &= - \sum_{\ell \in \text{WC}(\omega) \setminus \text{c.e.}} S_R(P^\ell(\omega)) \sqcup R^\ell(\omega), \end{aligned} \tag{24}$$

where c.e. denotes *cut everything*. Now we repeat the same computation for S_N , using (21). This gives the recursive definition of the antipode S_N ,

$$\begin{aligned} S_N(\mathbb{I}) &= \mathbb{I}, \\ S_N(\omega) &= - \sum_{\ell \in \text{FALC}(\omega) \setminus \text{c.e.}} S_N(P^\ell(\omega)) \sqcup R^\ell(\omega). \end{aligned} \tag{25}$$

We prove (23) by induction on the number of nodes. Plugging (23) into (25), we

find for $\omega \in \text{OF} \setminus \{\mathbb{I}\}$ that

$$\begin{aligned}
 S_N(\omega) &= - \sum_{\ell \in \text{FALC}(\omega) \setminus \text{c.e.}} S_R \left(\sum_{j \in \text{LC}(P^\ell(\omega))} P^j(P^\ell(\omega)) \sqcup R^j(P^\ell(\omega)) \right) \sqcup R^\ell(\omega) \\
 &= - \sum_{j \in \text{LC}(\omega)} \sum_{\ell \in \text{WC}(\omega) \setminus \text{c.e.}} S_R(P^j(\omega) \sqcup P^\ell(R^j(\omega))) \sqcup R^\ell(R^j(\omega)) \\
 &= \sum_{j \in \text{LC}(\omega)} S_R(P^j(\omega)) \sqcup \left(- \sum_{\ell \in \text{WC}(\omega) \setminus \text{c.e.}} S_R(P^\ell(R^j(\omega))) \sqcup R^\ell(R^j(\omega)) \right) \\
 &= \sum_{j \in \text{LC}(\omega)} S_R(P^j(\omega)) \sqcup S_R(R^j(\omega)) \\
 &= S_R \left(\sum_{j \in \text{LC}(\omega)} P^j(\omega) \sqcup R^j(\omega) \right).
 \end{aligned}$$

We have used the relation $S_R(\omega_1 \sqcup \omega_2) = S_R(\omega_1) \sqcup S_R(\omega_2)$, see Corollary 2 and the recursion (24), as well as a careful replacement of the summations over $\ell \in \text{FALC}(\omega)$ and $j \in \text{LC}(P^\ell(\omega))$ with an equivalent sum over $j \in \text{LC}(\omega)$ and $\ell \in \text{WC}(\omega)$. \square

As an example, we compute Δ_N and S_N for the word $\omega = \begin{matrix} \bullet \\ \bullet \\ \bullet \\ \bullet \end{matrix}$. The cuts LC and FALC are shown in Table 4. From the direct formulae we find $\Delta_N(\omega)$ and $S_N(\omega)$ as listed in Tables 5 and 6.

We complete this section by listing some well-known but very useful relations of Hopf algebras, see Sweedler [38] for further details.

Table 4. Cuts ℓ_i of an example word ω . The cuts $\{\ell_4, \ell_5, \ell_6\}$ are full cuts where the leftmost child of the invisible root is cut and ℓ_7 is the full cut where both the children of the invisible root are cut. Thus $\text{NLC}(\omega) = \{\ell_1, \ell_2\}$, $\text{LC}(\omega) = \{\ell_0, \dots, \ell_3\}$, $\text{ALC}(\omega) = \{\ell_0, \dots, \ell_2\}$, $\text{FALC}(\omega) = \{\ell_0, \dots, \ell_2\} \cup \{\ell_4, \dots, \ell_7\}$ and $\text{WC}(\omega) = \{\ell_0, \ell_4, \ell_7\}$.

i	ℓ_i	k	$P^{\ell_i}(\omega)$	$R^{\ell_i}(\omega)$	i	ℓ_i	k	$P^{\ell_i}(\omega)$	$R^{\ell_i}(\omega)$
0		0	\mathbb{I}		4		1		
1		1			5		2		
2		1			6		2		
3		2			7		1		\mathbb{I}

Table 6. Examples of the antipode S_N , defined recursively in (18) and nonrecursively in (23), of the Hopf algebra \mathcal{H}_N , given in Theorem 1, for all forests up to and including order 4. The dual of the antipode S_N is the antipode in the ordered GL Hopf algebra.

ω	$S_N(\omega)$
II	II
	$-\bullet$
	$-\bullet\bullet + 2\bullet\bullet$
	$\bullet\bullet$
	$-\bullet\bullet\bullet + 2\bullet\bullet\bullet + 2\bullet\bullet\bullet - 6\bullet\bullet\bullet$
	$-\bullet\bullet\bullet + \bullet\bullet\bullet + \bullet\bullet\bullet - 3\bullet\bullet\bullet$
	$\bullet\bullet\bullet - 3\bullet\bullet\bullet$
	$\bullet\bullet\bullet - 3\bullet\bullet\bullet$
	$-\bullet\bullet\bullet\bullet$
	$-\bullet\bullet\bullet\bullet + 2\bullet\bullet\bullet\bullet + 2\bullet\bullet\bullet\bullet + 2\bullet\bullet\bullet\bullet - 6\bullet\bullet\bullet\bullet - 6\bullet\bullet\bullet\bullet - 6\bullet\bullet\bullet\bullet + 24\bullet\bullet\bullet\bullet$
	$-\bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet - 3\bullet\bullet\bullet\bullet - 3\bullet\bullet\bullet\bullet - 3\bullet\bullet\bullet\bullet + 12\bullet\bullet\bullet\bullet$
	$-\bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet - 2\bullet\bullet\bullet\bullet - 3\bullet\bullet\bullet\bullet - 4\bullet\bullet\bullet\bullet + 12\bullet\bullet\bullet\bullet$
	$-\bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet + 2\bullet\bullet\bullet\bullet - 4\bullet\bullet\bullet\bullet - 3\bullet\bullet\bullet\bullet - 2\bullet\bullet\bullet\bullet + 12\bullet\bullet\bullet\bullet$
	$-\bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet + \bullet\bullet\bullet\bullet - \bullet\bullet\bullet\bullet - \bullet\bullet\bullet\bullet - \bullet\bullet\bullet\bullet + 4\bullet\bullet\bullet\bullet$
	$\bullet\bullet\bullet\bullet - \bullet\bullet\bullet\bullet - 2\bullet\bullet\bullet\bullet - 3\bullet\bullet\bullet\bullet + 12\bullet\bullet\bullet\bullet$
	$\bullet\bullet\bullet\bullet - 3\bullet\bullet\bullet\bullet - 2\bullet\bullet\bullet\bullet - \bullet\bullet\bullet\bullet + 12\bullet\bullet\bullet\bullet$
	$\bullet\bullet\bullet\bullet - \bullet\bullet\bullet\bullet - 2\bullet\bullet\bullet\bullet + 6\bullet\bullet\bullet\bullet$
	$\bullet\bullet\bullet\bullet - 2\bullet\bullet\bullet\bullet - \bullet\bullet\bullet\bullet + 6\bullet\bullet\bullet\bullet$
	$\bullet\bullet\bullet\bullet - 2\bullet\bullet\bullet\bullet - 2\bullet\bullet\bullet\bullet - 2\bullet\bullet\bullet\bullet + 12\bullet\bullet\bullet\bullet$
	$-\bullet\bullet\bullet\bullet + 4\bullet\bullet\bullet\bullet$
	$-\bullet\bullet\bullet\bullet + 4\bullet\bullet\bullet\bullet$
	$-\bullet\bullet\bullet\bullet + 4\bullet\bullet\bullet\bullet$
	$\bullet\bullet\bullet\bullet$

Corollary 2 [38]. \mathcal{H}_N is a commutative Hopf algebra, with shuffle product \sqcup , coproduct Δ_N and antipode S_N , then, for all $\omega, \omega_1, \omega_2 \in \text{OF}$,

$$\begin{aligned} \Delta_N(S_N(\omega)) &= T((S_N \otimes S_N)\Delta_N(\omega)), \\ S_N(\omega_1) \sqcup S_N(\omega_2) &= S_N(\omega_1 \sqcup \omega_2), \end{aligned}$$

where T is the twist map. It also follows that $S_N(S_N(\omega)) = \omega$ for all $\omega \in \text{OF}$.

4. Hopf Algebras Related to \mathcal{H}_N

There are two interesting commutative graded Hopf subalgebras of \mathcal{H}_N obtained by restricting from the set of ordered rooted trees to either the set of tall trees (i.e., trees where each parent has one child) or bushy trees (i.e., trees where there is only one parent). These Hopf subalgebras are useful, respectively, for determining the order conditions for the problem (10), when $f(y)$ is constant, or when the numerical scheme has high stage order.

In this section we will also show that the Hopf algebra \mathcal{H}_C of Butcher, Connes, and Kreimer, based on unordered trees, can be identified as a subalgebra of \mathcal{H}_N , and the Hopf algebra \mathcal{H}_P of Foissy is related to \mathcal{H}_N through the symmetrization operator. Finally, we find that the Hopf algebra structure \mathcal{H}_{Sh} of the shuffle algebra is related to \mathcal{H}_N through the operation of *freezing* vector fields, which can be defined as a quotient construction on \mathcal{H}_N .

4.1. Connections to the Butcher–Connes–Kreimer and Foissy Theory

Let \mathbb{T} denote all unordered trees and let \mathbb{F} denote all unordered forests, defined as the set of all empty or nonempty unordered words over the alphabet \mathbb{T} . Recall from [10] the following definition of the Butcher–Connes–Kreimer Hopf algebra.

Definition 7. Given the real vector space $C = \mathbb{R}\langle \mathbb{F} \rangle$, denote the commutative and noncocommutative Hopf algebra of unordered forests as $\mathcal{H}_C = (C, \mu_C, u_C, \Delta_C, e_C, S_C)$. The product $\mu_C : C \otimes C \rightarrow C$ is defined as the commutative concatenation

$$\mu_C(\omega_1 \otimes \omega_2) = \omega_1\omega_2 = \omega_2\omega_1.$$

The unit element $u_C : \mathbb{R} \rightarrow C$ is given by $u_C(1) = \mathbb{I}$. The coproduct $\Delta_C : C \rightarrow C \otimes C$ is defined by linearity and for any $\tau = B_i^+(\omega_1) \in \mathbb{T}$ and $\omega, \omega_1 \in \mathbb{F}$ by the recursion

$$\begin{aligned} \Delta_C(\mathbb{I}) &= \mathbb{I} \otimes \mathbb{I}, \\ \Delta_C(\tau) &= \tau \otimes \mathbb{I} + (I \otimes B_i^+) \Delta_C(B^-(\tau)), \\ \Delta_C(\omega\tau) &= \Delta_C(\omega)\Delta_C(\tau). \end{aligned} \tag{26}$$

The counit $e_C : C \rightarrow \mathbb{R}$ is defined by $e_C(\mathbb{I}) = 1$ and $e_C(\omega) = 0$ for $\omega \in \text{OF} \setminus \{\mathbb{I}\}$. The antipode $S_C : C \rightarrow C$ is, as usual, the two-sided inverse of the convolution in \mathcal{H}_C , see [10] for details.

The Butcher–Connes–Kreimer Hopf algebra was extended to ordered (planar) forests by Foissy. For ease of notation let $P = N$, then recall from [16] the following definition of the Foissy Hopf algebra.

Definition 8. Given the real vector space $P = \mathbb{R}\langle \text{OF} \rangle$, denote the noncommutative and noncocommutative Hopf algebra of ordered forests as $\mathcal{H}_P = (P, \mu_P, u_P, \Delta_P, e_P, S_P)$. The product $\mu_P : P \otimes P \rightarrow P$ is defined as the noncommutative concatenation

$$\mu_P(\omega_1 \otimes \omega_2) = \omega_1 \omega_2.$$

The unit, counit, coproduct and antipode are defined as in the Butcher–Connes–Kreimer Hopf algebra.

The Butcher–Connes–Kreimer is a Hopf subalgebra of the Foissy Hopf algebra. In fact, they are identical on forests which can be regarded as elements of both F and OF. The main tool used to provide the relationship between the Hopf algebras $\mathcal{H}_C, \mathcal{H}_P$ and \mathcal{H}_N is the symmetrization operator defined below.

Definition 9. The symmetrization operator $\Omega : N \rightarrow N$ is a map defined by linearity and the relations

$$\begin{aligned} \Omega(\mathbb{I}) &= \mathbb{I}, \\ \Omega(\omega\tau) &= \Omega(\omega) \sqcup \Omega(\tau), \\ \Omega(B_i^+(\omega)) &= B_i^+(\Omega(\omega)). \end{aligned}$$

The shuffle product permutes the trees in a forest in all possible ways, and the symmetrization of a tree is a recursive splitting in sums over all permutations of the branches. The symmetrization defines an equivalence relation on OF, that is, if

$$\Omega(\omega_1) = \Omega(\omega_2) \iff \omega_1 \sim \omega_2.$$

Thus $\omega_1 \sim \omega_2$ if and only if ω_2 can be obtained from ω_1 by permuting the order of the trees in the forest and the order of the branches of the trees. We see that an alternative characterization of Ω is

$$\Omega(\omega) = \sigma(\omega) \sum_{\substack{\tilde{\omega} \in \text{OF} \\ \tilde{\omega} \sim \omega}} \tilde{\omega}. \tag{27}$$

The integer $\sigma(\omega)$ is the classical symmetry coefficient, defined for trees and forests as

$$\begin{aligned} \sigma(\mathbb{I}) &= 1, \\ \sigma(\tau_1 \tau_2 \cdots \tau_k) &= \sigma(\tau_1) \cdots \sigma(\tau_k) \mu_1! \mu_2! \cdots, \\ \sigma(B_i^+(\tau_1 \cdots \tau_k)) &= \sigma(\tau_1 \cdots \tau_k), \end{aligned}$$

where the integers μ_1, μ_2, \dots count the number of equivalent trees among τ_1, \dots, τ_k . In other words, if we consider the full group of all possible permutations of trees and branches acting on a forest $\omega \in \text{OF}$, then $\sigma(\omega)$ is the size of the isotropy subgroup, that is the number of permutations leaving ω invariant. The total number of permutations acting on a given forest ω is given by the integer $\pi(\omega)$ defined as

$$\begin{aligned} \pi(\mathbb{I}) &= 1, \\ \pi(\tau_1 \tau_2 \cdots \tau_k) &= k! \sigma(\tau_1) \cdots \sigma(\tau_k), \\ \pi(B_i^+(\tau_1 \cdots \tau_k)) &= \pi(\tau_1 \cdots \tau_k). \end{aligned}$$

Note that once a forest ω is symmetrized, then another application of the symmetrization yields the scaling

$$\Omega(\Omega(\omega)) = \pi(\omega)\Omega(\omega). \tag{28}$$

Let F be the unordered forests. Clearly, there is a one-to-one correspondence between unordered forests and equivalence classes of ordered forests, thus there is a natural isomorphism $F \simeq \text{OF}/\sim$. Through this identification, we can interpret Ω as an injection $\Omega : C \rightarrow N$ where $C = \mathbb{R}\langle F \rangle$ and $N = \mathbb{R}\langle \text{OF} \rangle$. From (28) we see that the map $\Omega^{-1} : N \rightarrow C$ is defined as

$$\Omega^{-1}(a) = \sum_{\omega \in \text{OF}} \frac{a(\omega)}{\pi(\omega)} \text{forget}(\omega),$$

where $\text{forget} : N \rightarrow C$ is the natural identification of an ordered forest with the corresponding unordered forest, defines a left-sided inverse $\Omega^{-1}(\Omega(b)) = b$ for all $b \in C$.

Theorem 2. *The symmetrization operator $\Omega : C \rightarrow N$ defines an injective Hopf algebra homomorphism from the Hopf algebra \mathcal{H}_C of unordered forests into the Hopf algebra \mathcal{H}_N of ordered forests.*

Proof. A Hopf algebra homomorphism is a bialgebra homomorphism, which is a linear map that is both an algebra and a coalgebra homomorphism. Ω is an algebra homomorphism if

$$\begin{aligned} \Omega(u_C(1)) &= u_N(1), \\ \mu_N(\Omega(\omega_1) \otimes \Omega(\omega_2)) &= \Omega(\mu_C(\omega_1 \otimes \omega_2)). \end{aligned}$$

These conditions are automatically satisfied by Definition 9. Ω is a coalgebra homomorphism if

$$e_N(\Omega(\omega)) = e_C(\omega), \tag{29}$$

$$\Delta_N(\Omega(\omega)) = (\Omega \otimes \Omega)\Delta_C(\omega). \tag{30}$$

This implies that $\Omega(N)$ is a subcoalgebra of N . The first condition (29) follows immediately given that the counits are only nonzero when the argument is the empty forest.

The second relation follows using an induction argument. First we need to establish a useful relationship between Ω and Δ_C . Using the *sumless Sweedler notation*

$$\Delta(\omega) = \sum_{\omega_1 \otimes \omega_2 \in \Delta(\omega)} \omega_1 \otimes \omega_2 = \sum_i \omega_{(1)}^{(i)} \otimes \omega_{(2)}^{(i)} \equiv \sum \omega_{(1)} \otimes \omega_{(2)} \equiv \omega_{(1)} \otimes \omega_{(2)},$$

we find

$$\begin{aligned} (\Omega \otimes \Omega)\Delta_C(\omega) \sqcup (\Omega \otimes \Omega)\Delta_C(\tau) &= (\Omega \otimes \Omega)(\omega_{(1)} \otimes \omega_{(2)}) \\ &\quad \sqcup (\Omega \otimes \Omega)(\tau_{(1)} \otimes \tau_{(2)}) \\ &= \Omega(\omega_{(1)}) \otimes \Omega(\omega_{(2)}) \sqcup \Omega(\tau_{(1)}) \otimes \Omega(\tau_{(2)}) \\ &= \Omega(\omega_{(1)}) \sqcup \Omega(\tau_{(1)}) \otimes \Omega(\omega_{(2)}) \sqcup \Omega(\tau_{(2)}) \\ &= \Omega(\omega_{(1)}\tau_{(1)}) \otimes \Omega(\omega_{(2)}\tau_{(2)}) \\ &= (\Omega \otimes \Omega)\omega_{(1)}\tau_{(1)} \otimes \omega_{(2)}\tau_{(2)} \\ &= (\Omega \otimes \Omega)\Delta_C(\omega\tau). \end{aligned} \tag{31}$$

Now we prove (30) by induction on the number of vertices. The result is obvious for the identity \mathbb{I} . For a forest $\omega\tau$ we find, using (19) and (31), that

$$\begin{aligned} \Delta_N(\Omega(\omega\tau)) &= \Delta_N(\Omega(\omega) \sqcup \Omega(\tau)) \\ &= \Delta_N(\Omega(\omega)) \sqcup \Delta_N(\Omega(\tau)) \\ &= (\Omega \otimes \Omega)\Delta_C(\omega) \sqcup (\Omega \otimes \Omega)\Delta_C(\tau) \\ &= (\Omega \otimes \Omega)\Delta_C(\omega\tau). \end{aligned}$$

For a tree $\tau = B_i^+(\omega) \in \mathbf{T}$ and $\omega_1 \in \mathbf{F}$ we find, using the definition of Ω and the recursion formulas (17) and (26), that

$$\begin{aligned} \Delta_N(\Omega(\tau)) &= \Delta_N(\Omega(B_i^+(\omega))) = \Delta_N(B_i^+(\Omega(\omega))) \\ &= B_i^+(\Omega(\omega)) \otimes \mathbb{I} + (\mathbb{I} \otimes B_i^+)\Delta_N(\Omega(\omega)) \\ &= \Omega(\tau) \otimes \mathbb{I} + (\mathbb{I} \otimes B_i^+)(\Omega \otimes \Omega)\Delta_C(\omega) \\ &= (\Omega \otimes \Omega)(\tau \otimes \mathbb{I} + (\mathbb{I} \otimes B_i^+)\Delta_C(\omega)) \\ &= (\Omega \otimes \Omega)\Delta_C(\tau). \end{aligned}$$

The proof extends to a general $a \in N$ by linearity. □

A similar explanation is used to show that Ω is a Hopf algebra homomorphism from \mathcal{H}_P into \mathcal{H}_N . However, the map $\Omega : \mathcal{H}_P \rightarrow \mathcal{H}_N$ is *not* injective, and therefore we cannot identify \mathcal{H}_P with a Hopf subalgebra of \mathcal{H}_N . Given that both \mathcal{H}_C and \mathcal{H}_P are related to \mathcal{H}_N by a Hopf algebra homomorphism the respective antipodes are related.

Corollary 3. *Let $\Omega : N \rightarrow N$ be a bialgebra homomorphism, then*

$$\begin{aligned} S_N(\Omega(\omega)) &= \Omega(S_C(\omega)), & \text{for all } \omega \in F, \\ S_N(\Omega(\omega)) &= \Omega(S_P(\omega)), & \text{for all } \omega \in OF. \end{aligned}$$

Note that the symmetrization operator $\Omega : C \rightarrow N$ is left invertible, so expressions for the product μ_C , coproduct Δ_C and antipode S_C of the Hopf algebra of unordered forests \mathcal{H}_C , can be directly expressed in terms of the corresponding functions in the Hopf algebra of ordered forests \mathcal{H}_N , they are

$$\begin{aligned} \mu_C(\omega_1 \otimes \omega_2) &= \Omega^{-1}(\mu_N(\Omega(\omega_1) \otimes \Omega(\omega_2))), \\ \Delta_C(\omega) &= (\Omega \otimes \Omega)^{-1} \Delta_N(\Omega(\omega)), \\ S_C(\omega) &= \Omega^{-1}(S_N(\Omega(\omega))). \end{aligned}$$

In the final part of this section we will elaborate on the connections between the LS-series built on ordered forests, and their commutative counterpart, the S-series. These series belong, respectively, to the dual spaces N^* and C^* and are naturally associated through the dual map $\Omega^* : N^* \rightarrow C^*$ taking the series of ordered forests to unordered forests. If $\alpha \in N^*$ and $\beta = \Omega^*(\alpha) \in C^*$, we find, from (27), that

$$\beta(\omega) = \langle \Omega^*(\alpha), \omega \rangle = \langle \alpha, \Omega(\omega) \rangle = \alpha(\Omega(\omega)) = \sigma(\omega) \sum_{\tilde{\omega} \sim \omega} \alpha(\tilde{\omega}).$$

On a manifold with a commutative Lie group action the elementary differential operators $\mathcal{F}(\omega)$ do not depend on the ordering. Thus we find that the S-series of β as defined in [31] equals the LS-series of α as given in (11),

$$LS(\alpha) = \sum_{\omega \in OF} h^{|\omega|} \alpha(\omega) \mathcal{F}(\omega) = \sum_{\omega \in F} h^{|\omega|} \frac{\beta(\omega)}{\sigma(\omega)} \mathcal{F}(\omega) = S(\beta).$$

This shows that the normalization $1/\sigma(\omega)$ in the commutative case is compatible with our normalization in the noncommutative case.

It is interesting to characterize the image of the *logarithmic* and *exponential* series under Ω^* . If α is logarithmic (Lemma 4), then

$$\begin{aligned} \beta(\mathbb{I}) &= \alpha(\mathbb{I}) = 0, \\ \beta(\omega\tau) &= \langle \alpha, \Omega(\omega\tau) \rangle = \langle \alpha, \Omega(\omega) \sqcup \Omega(\tau) \rangle = 0, \quad \text{for } \omega, \tau \neq \mathbb{I}, \end{aligned}$$

thus β is nonzero only on trees. If α is exponential, then

$$\begin{aligned} \beta(\mathbb{I}) &= \alpha(\mathbb{I}) = 1, \\ \beta(\tau_1 \cdots \tau_k) &= \langle \alpha, \Omega(\tau_1 \cdots \tau_k) \rangle = \langle \alpha, \Omega(\tau_1) \sqcup \cdots \sqcup \Omega(\tau_k) \rangle \\ &= \alpha(\Omega(\tau_1)) \cdots \alpha(\Omega(\tau_k)) = \beta(\tau_1) \cdots \beta(\tau_k). \end{aligned}$$

This is a well-known condition in the composition of S-series, see [9].

4.2. The Shuffle Algebra and Frozen Vector Fields

In the proof of Proposition 3 we defined the Hopf algebraic structure \mathcal{H}_{Sh} as the dual of the free associative algebra built of words over an alphabet A , where μ_{Sh} is the shuffle and Δ_{Sh} the dual of the concatenation product. We will briefly comment upon the connection between \mathcal{H}_N and \mathcal{H}_{Sh} in the context of Lie group integrators.

In the theory of Lie group integrators, it is common to call constant sections $C^\infty(\mathcal{M}, \mathfrak{g})$ *frozen vector fields*. If f is frozen, then the Lie derivative $g[f] = 0$ for all vector fields g . On the algebraic side, a tree $\tau \in \text{OT}$ represents a frozen vector field if the left grafting of anything nonconstant to the tree is zero,

$$\tau_1[\tau_2] = 0, \quad \text{for all } \tau_1, \tau_2 \in \text{OT}.$$

In this case we see, from (2), that the ordered GL product becomes just the non-commutative concatenation product $\tau_1 \circ \tau_2 = \tau_1 \tau_2$. The freezing of certain vector fields can be understood as the quotient \mathcal{H}_N/G , where G is the linear span of any $\tau_1 \in \text{OT}$ grafted to a frozen vector field.

As a special example, we consider the case where all single-node trees are frozen, so that taller trees cannot be produced. Letting $A = \{B_i^+(\mathbb{I})\}_{i \in \mathcal{I}} = \{a_i\}_{i \in \mathcal{I}}$ be the alphabet of all single-node trees, we find from (17), given that the single node trees are primitive elements, the following well-known recursion for the coproduct (dual of noncommutative concatenation)

$$\begin{aligned} \Delta_{\text{Sh}}(\mathbb{I}) &= \mathbb{I} \otimes \mathbb{I}, \\ \Delta_{\text{Sh}}(\omega a_i) &= \omega a_i \otimes \mathbb{I} + \Delta_{\text{Sh}}(\omega)(\mathbb{I} \otimes a_i), \end{aligned}$$

thus $\mathcal{H}_{\text{Sh}} = \mathcal{H}_N/G$.

5. Concluding Remarks

In this paper we have investigated the algebraic structure of the Hopf algebra underlying numerical Lie group integrators. We have developed both recursive and direct formulae for the coproduct and the antipode, and we have in particular emphasized the connection to the Hopf algebra of classical Butcher theory and to

the Hopf algebra structure of the shuffle algebra. We believe that this work is of particular interest for the construction of symbolic software packages dealing with computations involving algebras of noncommutative derivations. The algebraic structure of \mathcal{H}_N is of a universal nature and should also be of interest outside the field of numerical integration, for example, in the renormalization of quantum field theory and the Chen–Fliess theory for optimal control.

Acknowledgements

This research was partly done during the Special Year in Geometric Integration at the Centre for Advanced Study, Oslo 2002–2003. We would like to thank Brynjulf Owren and Håvard Berland for discussions on the topic of the paper. Also the anonymous referees who have provided many useful comments to the first manuscript.

References

- [1] R. Abraham, J. E. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications*, 2nd ed., AMS 75, Springer-Verlag, New York, 1988.
- [2] H. Berland and B. Owren. Algebraic structures on ordered rooted trees and their significance to Lie group integrators, in *Group Theory and Numerical Analysis, CRM Proc. Lecture Notes*, Vol. 39, pp. 49–63, Amer. Math. Soc., Providence, RI, 2005.
- [3] Ch. Brouder, Runge-Kutta methods and renormalization, *Eur. Phys. J. C* **12** (2000), 521–534.
- [4] Ch. Brouder, Trees, renormalization and differential equations, *BIT* **44**(3) (2004), 425–438.
- [5] J. C. Butcher, Coefficients for the study of Runge-Kutta integration processes, *J. Austr. Math. Soc.* **3** (1963), 185–201.
- [6] J. C. Butcher, An algebraic theory of integration methods, *Math. Comp.* **26** (1972), 79–106.
- [7] A. Cayley, On the theory of analytic forms called trees, *Phil. Mag.* **13** (1857), 19–30.
- [8] E. Celledoni, A. Marthinsen, and B. Owren, Commutator-free Lie group methods, *FGCS* **19**(3) (2003), 341–352.
- [9] P. Chartier, E. Faou, and A. Murua, An algebraic approach to invariant preserving integrators: The case of quadratic and Hamiltonian invariants, *Numer. Math.* **103**(4) (2006), 575–590.
- [10] A. Connes and D. Kreimer, Hopf algebras, renormalization and noncommutative geometry, *Comm. Math. Phys.* **199**(1) (1998), 203–242.
- [11] A. Connes and D. Kreimer, Renormalization in quantum field theory and the Riemann–Hilbert problem, I. The Hopf algebra structure of graphs and the main theorem, *Comm. Math. Phys.* **210**(1) (2000), 249–273.
- [12] A. Connes and H. Moscovici, Hopf algebras, cyclic cohomology and the transverse index theorem, *Comm. Math. Phys.* **198**(1) (1998), 199–246.
- [13] P. E. Crouch and R. Grossman, Numerical integration of ordinary differential equations on manifolds, *J. Nonlinear Sci.* **3** (1993), 1–33.
- [14] A. Dür, *Möbius Functions, Incidence Algebras and Power Series Representations*, Lecture Notes in Mathematics, Vol. 1202, Springer-Verlag, Berlin, 1986.
- [15] L. Foissy, Les algèbres de Hopf des arbres enracinés décorés, I, *Bull. Sci. Math.* **126**(3) (2002), 193–239.
- [16] L. Foissy, Les algèbres de Hopf des arbres enracinés décorés, II, *Bull. Sci. Math.* **126**(4) (2002), 249–288.

- [17] R. Grossman and R. G. Larson, Hopf-algebraic structure of families of trees, *J. Algebra* **126**(1) (1989), 184–210.
- [18] E. Hairer and G. Wanner, On the Butcher group and general multi-value methods, *Computing (Arch. Elektron. Rechnen)* **13**(1) (1974), 1–15.
- [19] M. E. Hoffman, Combinatorics of rooted trees and Hopf algebras, *Trans. Amer. Math. Soc.* **355**(9) (2003), 3795–3811.
- [20] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, Lie-group methods, *Acta Numer.* **9** (2000), 215–365.
- [21] D. Kreimer, On the Hopf algebra structure of perturbative quantum field theories, *Adv. Theor. Math. Phys.* **2** (1998), 303–334.
- [22] D. Lewis and J. C. Simo, Conserving algorithms for the dynamics of Hamiltonian systems of Lie groups, *J. Nonlinear Sci.* **4** (1994), 253–299.
- [23] D. Lewis and J. C. Simo, Conserving algorithms for the n -dimensional rigid body, *Fields Inst. Commun.*, Vol. 10, Amer. Math. Soc., Providence, RI, 1995.
- [24] R. H. Merson, An operational method for the study of integration processes, in *Proc. Conf., Data Processing & Automatic Computing Machines*, pp. 110–1–110–25, 1957.
- [25] H. Z. Munthe-Kaas, Lie–Butcher theory for Runge-Kutta methods, *BIT* **35**(4) (1995), 572–587.
- [26] H. Z. Munthe-Kaas, Runge-Kutta methods on Lie groups, *BIT* **38**(1) (1998), 92–111.
- [27] H. Z. Munthe-Kaas, High order Runge-Kutta methods on manifolds, in *Proceedings of the NSF/CBMS Regional Conference on Numerical Analysis of Hamiltonian Differential Equations* (Golden, CO, 1997), Vol. 29,1, pp. 115–127, 1999.
- [28] H. Z. Munthe-Kaas and S. Krogstad, On enumeration problems in Lie–Butcher theory, *FGCS* **19**(7) (2003), 1197–1205.
- [29] H. Z. Munthe-Kaas and B. Owren, Computations in a free Lie algebra, *Roy. Soc. London Phil. Trans. Ser. A Math. Phys. Engrg. Sci.* **357** (1754), 957–981, 1999.
- [30] H. Z. Munthe-Kaas and W. M. Wright, Lie–Butcher theory, 2006 (in preparation).
- [31] A. Murua, Formal series and numerical integrators, I. Systems of ODEs and symplectic integrators, *Appl. Numer. Math.* **29**(2) (1999), 221–251.
- [32] A. Murua, The Hopf algebra of rooted trees, free Lie algebras, and Lie series, *Found. Comput. Math.* **6** (2006), 387–426.
- [33] B. Owren, Order conditions for commutator-free Lie group methods, *J. Phys. A: Math. Gen.* **39** (2006), 5585–5599.
- [34] B. Owren and A. Marthinsen, Runge-Kutta methods adapted to manifolds and based on rigid frames, *BIT* **39**(1) (1999), 116–142.
- [35] F. Panaite, Relating the Connes–Kreimer and Grossman–Larson Hopf algebras built on rooted trees, *Lett. Math. Phys.* **51**(3) (2000), 211–219.
- [36] Ch. Reutenauer, *Free Lie Algebras*, Oxford University Press, New York, 1993.
- [37] N. J. A. Sloane, The on-line encyclopedia of integer sequences, *Notices Amer. Math. Soc.* **50**(8) (2003), 912–915.
- [38] M. E. Sweedler, *Hopf Algebras*, Mathematics Lecture Note Series, W. A. Benjamin, New York, 1969.