Hopf Algebras of Labeled Trees and Some Associated Differential Algebra Structures

Robert Grossman
University of Illinois at Chicago

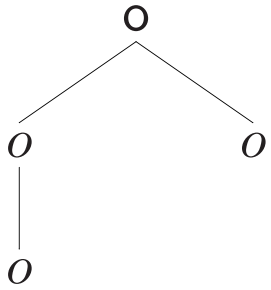
Joint Work with Richard Larson
University of Illinois at Chicago

April, 2007

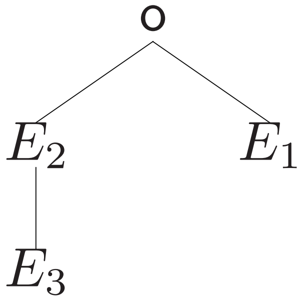
Part 1: Introduction and Overview

Notation

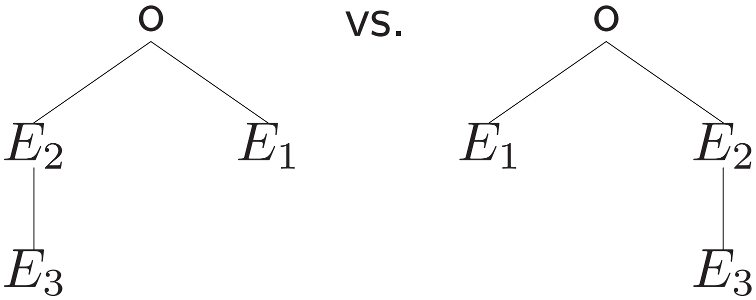
Tree:



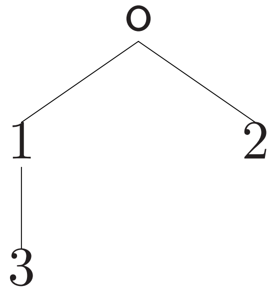
Labeled Tree:



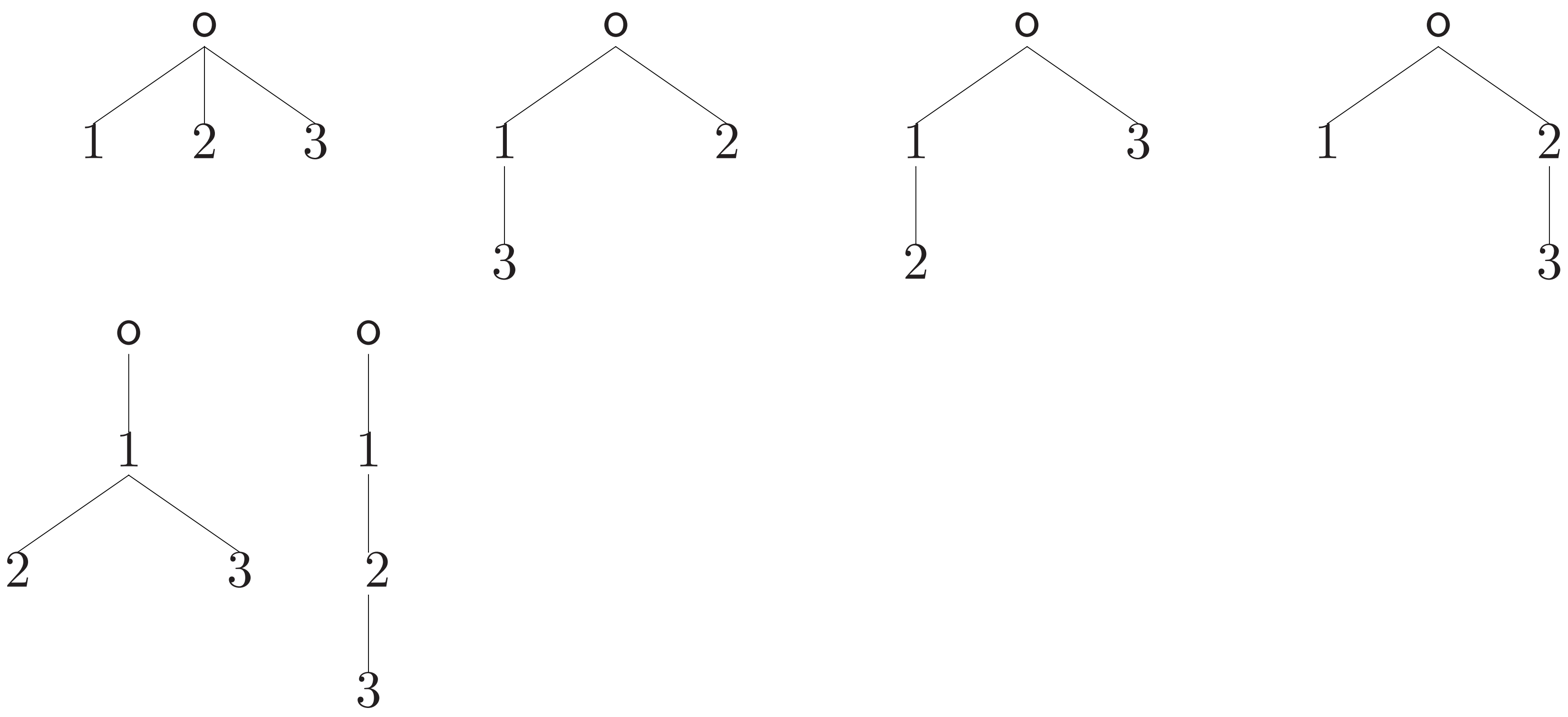
Ordered Tree:



Heap Ordered Tree:



Notation (con’d): Heap Ordered Trees



Problem 1: Simplifying Higher Order Derivations

Set up: Space $X = \mathbf{R}^n$. k-algebra R of functions on X, say polynomials $k[x_1, \ldots, x_n]$. Derivations $E_1, \ldots, E_M \in \mathrm{Der}(R)$ of the form $E_j = \sum_\mu a_j^\mu D_\mu$, where $a_j^\mu \in R$ and $D_\mu = \partial/\partial x_\mu$.

Example: Three vector fields on $\mathbf{R}^8$:

$$\begin{aligned}
E_1 &= \frac{\partial}{\partial x_1} \\
E_2 &= \frac{\partial}{\partial x_2} - x_1 \frac{\partial}{\partial x_3} + \frac{1}{2} x_1^2 \frac{\partial}{\partial x_4} + x_1 x_2 \frac{\partial}{\partial x_5} \\
&\quad - \frac{1}{6} x_1^3 \frac{\partial}{\partial x_6} - \frac{1}{2} x_1^2 x_2 \frac{\partial}{\partial x_7} - \frac{1}{2} x_1 x_2^2 \frac{\partial}{\partial x_8} \\
E_3 &= \frac{\partial}{\partial x_3} - x_1 \frac{\partial}{\partial x_4} - x_2 \frac{\partial}{\partial x_5} + \frac{1}{2} x_1^2 \frac{\partial}{\partial x_6} + x_1 x_2 \frac{\partial}{\partial x_7} + \frac{1}{2} x_2^2 \frac{\partial}{\partial x_8}
\end{aligned}$$

Then how do I compute efficiently $p, q \in k{<}E_1, E_2, E_3{>}$?:

$$p = E_3E_2E_1 - E_3E_1E_2 - E_2E_1E_3 + E_1E_2E_3$$

$$q = [[E_2, E_1], E_1] = \frac{\partial}{\partial x_4} - x_1\frac{\partial}{\partial x_6} - x_2\frac{\partial}{\partial x_7}$$

Space X	**Trees**	**Algebraic Structure**	**Comment**
$X = \mathbf{R}^n$	Labeled Trees	bialgebra	dual is CK-Algebra
$X = G$ compact group	Labeled, Ordered Trees	bialgebra	assoc. graded dual is LR-Algebra of planar binary trees

Problem 2: Simplifying The Action of Higher Order Derivations on Functions

Set up: Space $X = \mathbf{R}^n$. k-algebra R of functions on X, say polynomials $k[x_1, \ldots, x_n]$. Derivations $E_j \in \mathrm{Der}(R)$, with coefficients from R.

Example: Three vector fields on $\mathbf{R^8}$:

$$\begin{aligned}
E_1 &= \frac{\partial}{\partial x_1} \\
E_2 &= \frac{\partial}{\partial x_2} - x_1 \frac{\partial}{\partial x_3} + \frac{1}{2} x_1^2 \frac{\partial}{\partial x_4} + x_1 x_2 \frac{\partial}{\partial x_5} \\
&\quad - \frac{1}{6} x_1^3 \frac{\partial}{\partial x_6} - \frac{1}{2} x_1^2 x_2 \frac{\partial}{\partial x_7} - \frac{1}{2} x_1 x_2^2 \frac{\partial}{\partial x_8}
\end{aligned}$$

Then how do I compute efficiently $p(r)$, where $p \in k{<}E_1, E_2, E_3{>}$ and $r \in R$?

$(E_2E_2E_1+E_1)(3x_1^3x_2-2x_1x_7) \quad [E_2, E_1](x_3) = 1, \quad [[E_2, E_1], E_1](x_4) = 1,$

Space X	**Trees**	**Algebraic Structure**	**Comment**
$X = \mathbf{R}^n$	Labeled Trees	H-Module Algebra	H = bialgebra of trees
$X = G$ compact group	Labeled, Ordered Trees	R/k-bialgebra	requires connection on G

R/k-bialgebras

R/k-bialgebras encode differential algebra structure and were introduced by Nichols and Weisfeiler in 1982:

W. Nichols and B. Weisfeiler, Differential formal groups of J. F. Ritt, Amer. J. Math. 104 (1982) 943-1003.

W. Nichols, The Kostant structure theorems for K/k-Hopf algebras, Joural Algebra 97 (1985) 313-328.

Problem 3: Computing Flows and their Symmetries

Set up: Space $X = G$, a group. Fix a basis $\{Y_1, \ldots, Y_M\}$ of the Lie algebra. Fix a derivation $E = \sum_\mu a^\mu Y_\mu$ with coefficients from the coordinate ring R.

Example: Compute symbolic appoximations to the flow $\exp tE$ or

$$\exp\Big(t \begin{array}{c}\circ\\ |\\ E\end{array}\Big) = 1 + t \begin{array}{c}\circ\\ |\\ E\end{array} + \frac{t^2}{2!} \begin{array}{c}\circ\\ |\\ E\end{array} \cdot \begin{array}{c}\circ\\ |\\ E\end{array} + \ldots$$

Space X	**Trees**	**Algebraic Structure**	**Comment**
$X = G$	Heap Ordered Trees	bialgebra	assoc. graded dual is MR-Hopf algebra of permutations

Cayley 1857

Cayley noted that repeated differentation led to trees and showed the following:

Let t_n be the number of trees with n nodes. Caley showed that if $T(z) = \sum_{n=1}^{\infty} t_n z^n$, then

$$T(z) = z \prod_{n=1}^{\infty} (1 - z^n)^{-t_n}.$$

A. Cayley, On the theory of the analytical forms called trees, in “Collected Mathematical Papers of Arthur Cayley,” Cambridge University Press, Cambridge, 1890, 3, 242-246.

Part 2: Simplifying Higher Order Derivations in $\mathbf{R}^n$

This is Problem 1 in the commutative case.

Example: Simplifying Higher Order Derivations

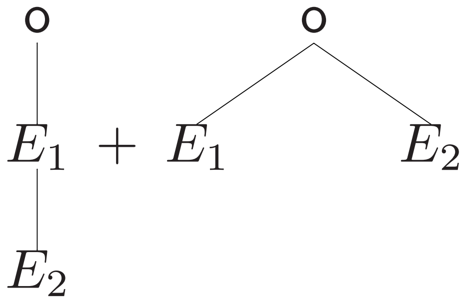
Consider three vector fields

$$E_1 = a_1(x)D_1 + \cdots a_N(x)D_N, \qquad E_2 = b_1(x)D_1 + \cdots b_N(x)D_N,$$

$$E_3 = c_1(x)D_1 + \ldots c_N(x)D_N.$$

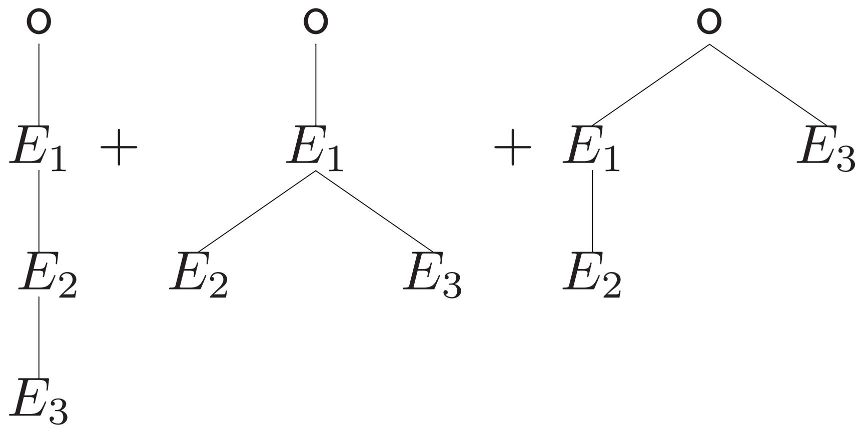
where $D_i = \partial/\partial x_i$, and $a_i(x)$, $b_i(x)$ and $c_i(x)$ are smooth functions on $\mathbf{R}^N$. Now

$$E_2 \cdot E_1 = \sum b_j(D_j a_i)D_i + \sum b_j a_i D_j D_i$$

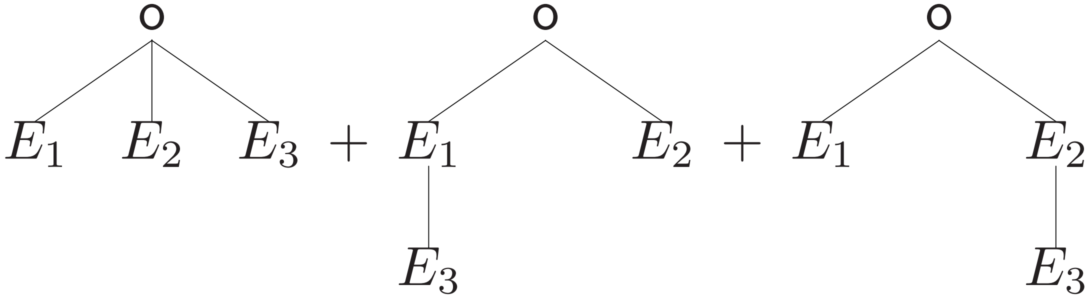


Now $E_3 \cdot E_2 \cdot E_1$ is equal to

$$\sum c_k(D_k b_j)(D_j a_i)D_i + \sum c_k b_j(D_k D_j a_i)D_i + \sum c_k b_j(D_j a_i)D_k D_i$$



$$+\sum c_k b_j a_i D_k D_j D_i + \sum c_k b_j (D_k a_i) D_j D_i + \sum c_k (D_k b_j) a_i D_j D_i.$$



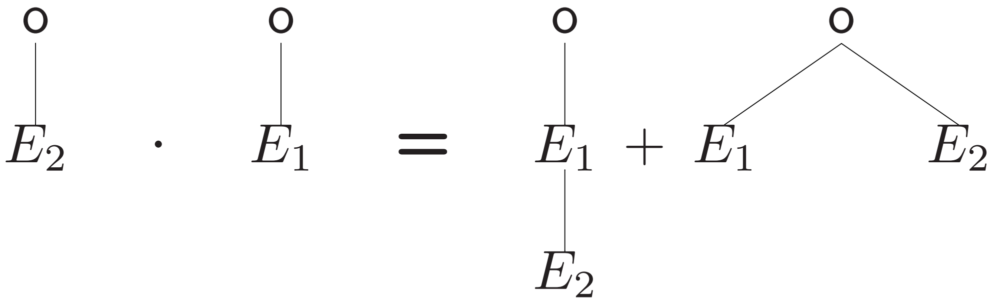
Theorem (1989).

If a family of rooted trees (possibly with additional structure) and four operations (DeleteRoot, Nodes, Restriction, AttachRoot) satisfies 9 axioms, the then family forms a Hopf algebra.

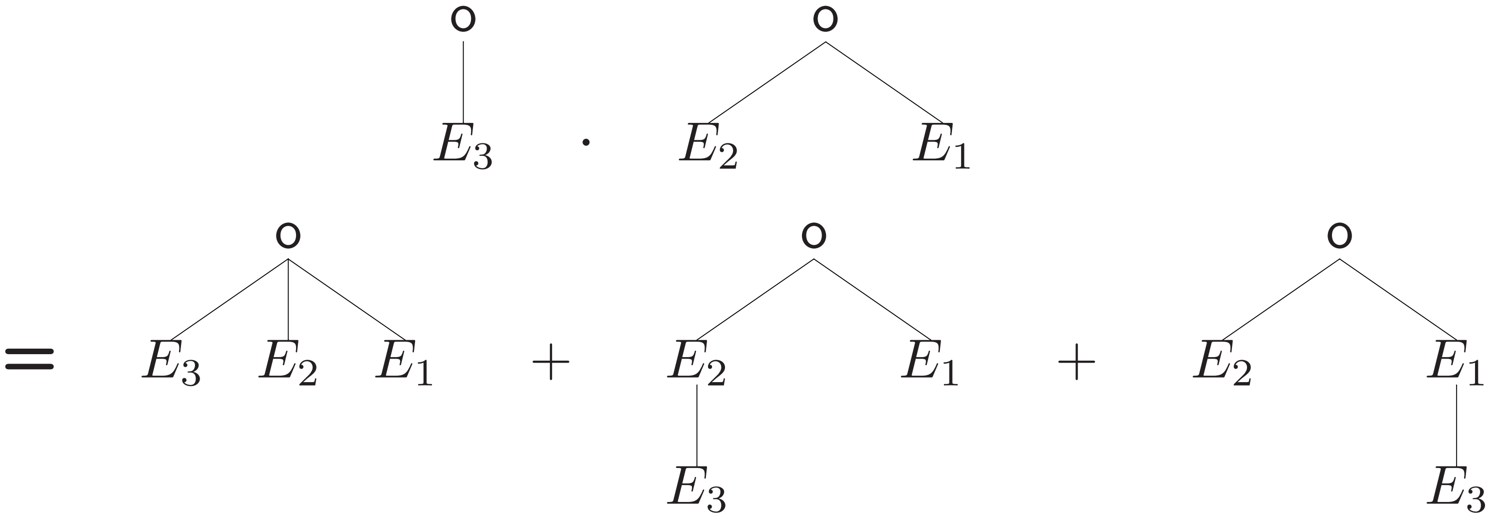
Examples:

- trees
- labeled trees
- ordered trees
- heap ordered trees

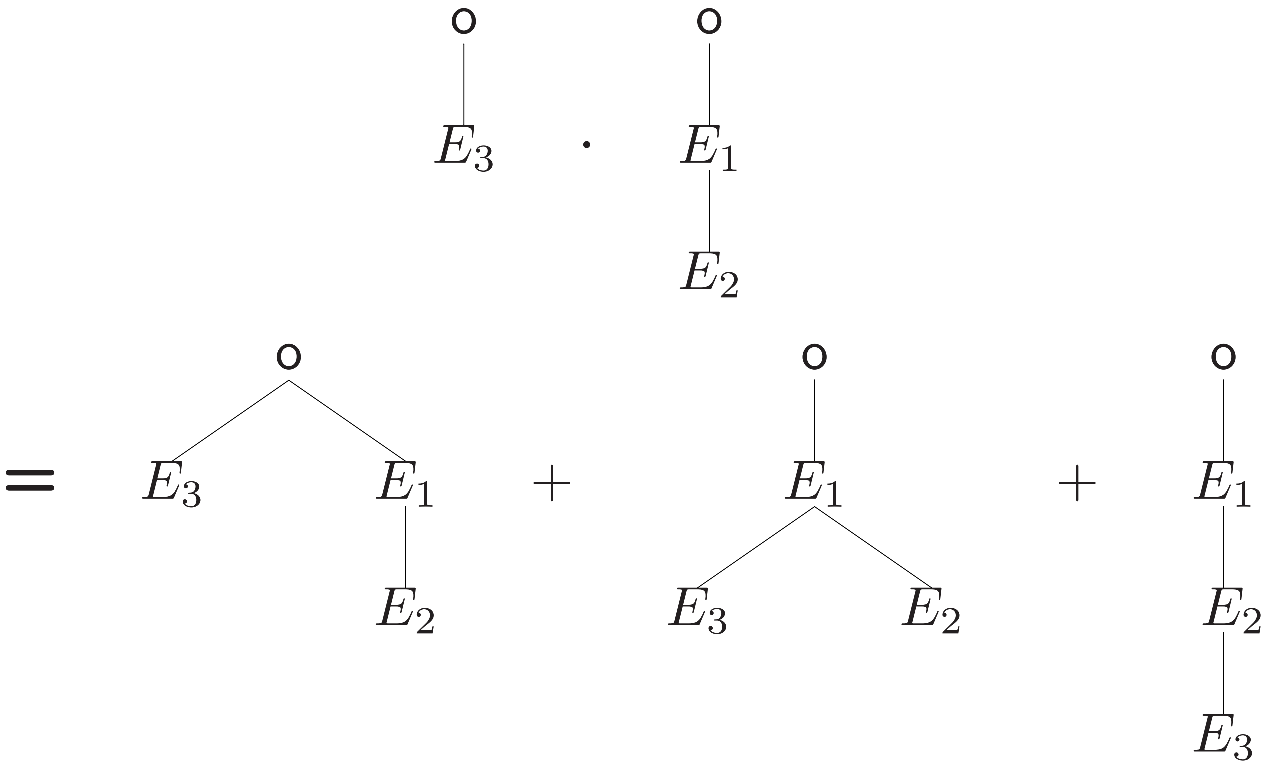
Product - Example



Product - Example



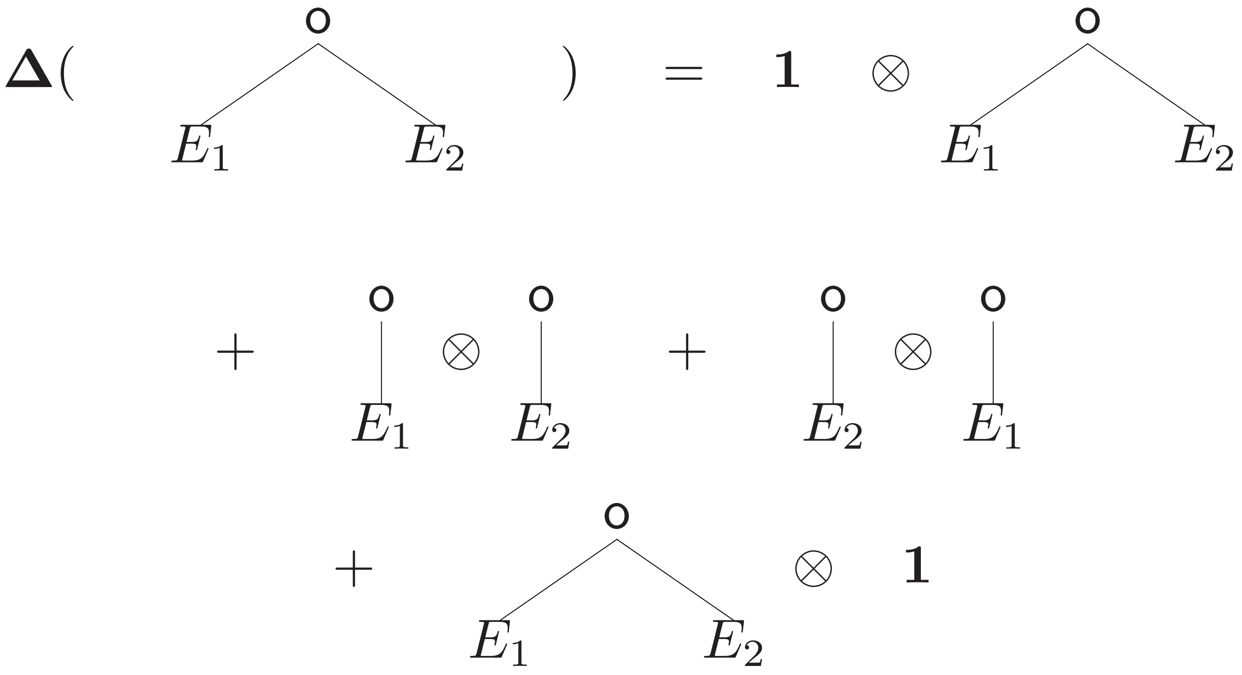
Product - Example



Coproduct

$$\boldsymbol{\Delta}\Big(\overset{\circ}{\underset{E_1}{\Big|}} \Big) = \mathbf{1} \otimes \overset{\circ}{\underset{E_1}{\Big|}} + \overset{\circ}{\underset{E_1}{\Big|}} \otimes \mathbf{1}$$

Coproduct (cont'd)



Example:

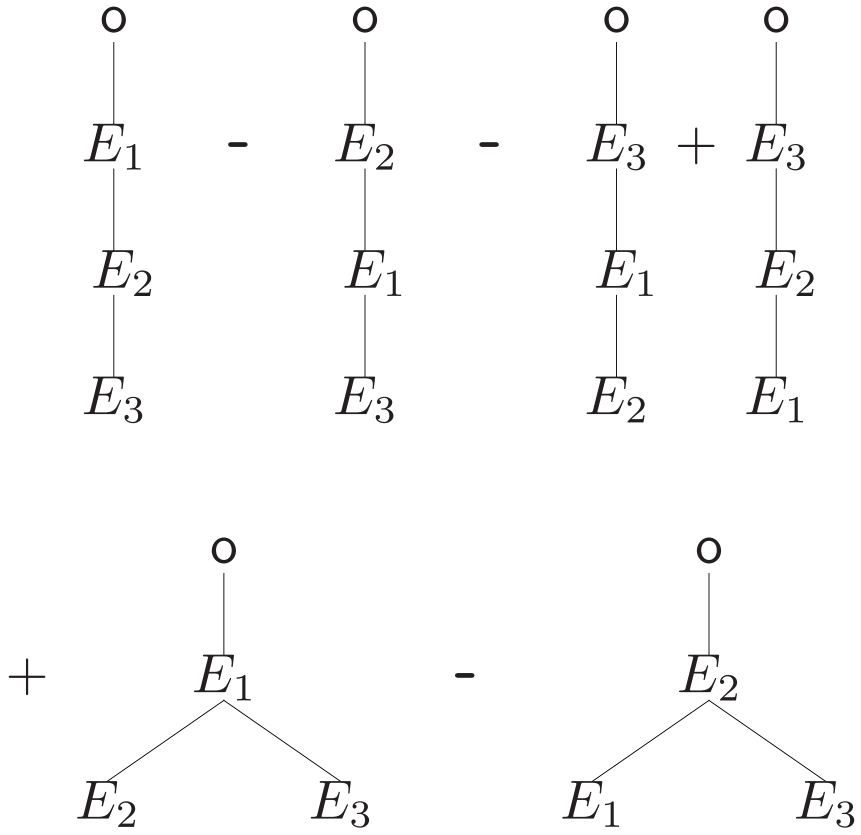
An expression such as

$$p = E_3E_2E_1 - E_3E_1E_2 - E_2E_1E_3 + E_1E_2E_3$$

corresponds to 24 trees, 18 of which cancel (each cancellation saves $O(N^3)$ differentiations.

$$\begin{aligned}
&\sum a_3^{\mu_3}(D_{\mu_3}a_2^{\mu_2})(D_{\mu_2}a_1^{\mu_1})D_{\mu_1} - \sum a_3^{\mu_3}(D_{\mu_3}a_1^{\mu_2})(D_{\mu_2}a_2^{\mu_1})D_{\mu_1} \\
&- \sum a_2^{\mu_3}(D_{\mu_3}a_1^{\mu_2})(D_{\mu_2}a_3^{\mu_1})D_{\mu_1} + \sum a_1^{\mu_3}(D_{\mu_3}a_2^{\mu_2})(D_{\mu_2}a_3^{\mu_1})D_{\mu_1} \\
&+ \sum a_3^{\mu_3}a_2^{\mu_2}(D_{\mu_3}D_{\mu_2}a_1^{\mu_1})D_{\mu_1} - \sum a_3^{\mu_3}a_1^{\mu_2}(D_{\mu_3}D_{\mu_2}a_2^{\mu_1})D_{\mu_1},
\end{aligned}$$

The surviving six trees are:



Summary

Space X	**Trees**	**Algebraic Structure**	**Comment**
$X = \mathbf{R}^n$	Labeled Trees	bialgebra	helpful with simplification of expressions involving higher order derivations (each tree that cancels results in $O(N^r)$ fewer computations)

Part 3: Simplifying The Action of Higher Order Derivations on Functions

This is Problem 2 in the commutative case.

Recall the Co-Product

$$\mathbf{\Delta}\left(\begin{array}{c}\circ\\ |\\ E\end{array} \right) = \mathbf{1} \otimes \begin{array}{c}\circ\\ |\\ E\end{array} + \begin{array}{c}\circ\\ |\\ E\end{array} \otimes \mathbf{1}$$

We want:

$$\begin{array}{c}\circ\\ |\\ E\end{array} (f) \quad \leftrightarrow \quad E(f),$$

where $E \in \mathrm{Der}(R)$ and $f \in R$.

Example

Analysis of actions of higher order derivations on algebras of functions using H-module algebras.

$$\begin{aligned} \begin{array}{c}\mathsf{o}\\ \vert\\ E\end{array}\;(\mathbf{fg}) &= \mathbf{f}\;\begin{array}{c}\mathsf{o}\\ \vert\\ E\end{array}\;\mathbf{g}+\mathbf{g}\;\begin{array}{c}\mathsf{o}\\ \vert\\ E\end{array}\;\mathbf{f}\\ &= (\mathbf{1}\otimes\;\begin{array}{c}\mathsf{o}\\ \vert\\ E\end{array}\;+\;\begin{array}{c}\mathsf{o}\\ \vert\\ E\end{array}\;\otimes\mathbf{1})(\mathbf{f}\otimes\mathbf{g})\\ &= \boldsymbol{\Delta}(\;\begin{array}{c}\mathsf{o}\\ \vert\\ E\end{array}\;)\,(\mathbf{f}\otimes\mathbf{g}) \end{aligned}$$

H-Module Algebras

Let R be a commutative k-algebra, and let H be a k-bialgebra. The algebra R is a *left H-module algebra* if R is a left H-module for which

$$h \cdot (ab) = \sum_{(h)} (h_{(1)} \cdot a)(h_{(2)} \cdot b),$$

where $h \in H$, $\Delta(h) = \sum_{(h)} h_{(1)} \otimes h_{(2)}$, and a, $b \in R$.

Why is this important? Many practical computations involve analyzing the actions of differential operators on specific functions. This translates into understanding the actions of specific bialgebras algebras of trees on specific algebras of functions or H-module algebras.

Theorem (1990).

- Fix derivations $E_j = \sum_\mu a_j^\mu D_\mu$, with $a_j^\mu \in R$, $D_\mu = \frac{\partial}{\partial x_\mu}$.

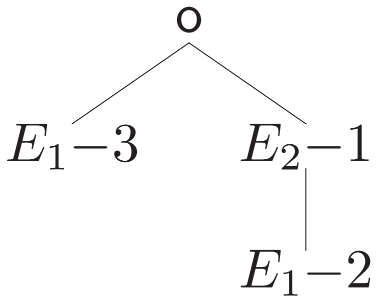
- Let $k{<}E_1, \ldots, E_M{>}$ be the free associative algebra generated by the formal symbols E_1, …, E_M. Let $H = k\{\mathcal{LT}(E_1, \ldots, E_M)\}$ be the Hopf algebra consisting of rooted trees labeled with derivations E_j. Let $\mathrm{Diff}(E_1, \ldots, E_M)$ be the higher order derivations generated by the derivations E_1, …, E_M.

Then one can define a morphism ψ and an action so that R is an H-module and the following diagram commutes:

$$\begin{tikzcd}
k{<}E_1, \ldots, E_M{>} \arrow[r] \arrow[dr] & k\{\mathcal{LT}(E_1, \ldots, E_M)\} \arrow[d, "\psi"] \\
 & \text{Diff}(E_1, \ldots, E_M)
\end{tikzcd} \tag{1}$$

Example of ψ-map:

h =



o

E_1–3

E_2–1

E_1–2

Define:

$$\psi(h) = \sum_{\mu_1,\mu_2,\mu_3} a_1^{\mu_3} a_1^{\mu_2} (D_{\mu_2} a_2^{\mu_1}) D_{\mu_3} D_{\mu_1} \in \text{End}(R),$$

where $h \in H$.

ψ map in general:

Let $h \in k\{\mathcal{LT}(E_1, \ldots, E_M)\}$ be a labeled tree. Number the root of h with 0, and number the other nodes 1, . . . , m. Suppose that node i, $i \geq 0$, has children j_1, . . . , j_k. Define

$$R(i) = \begin{cases} D_{\mu_{j_k}} \cdots D_{\mu_{j_1}} f & \text{if } i = 0, \\ D_{\mu_{j_k}} \cdots D_{\mu_{j_1}} a_i^{\mu_i} & \text{otherwise.} \end{cases}$$

Define

$$h \cdot f = \sum_{\mu_1, \ldots, \mu_m = 1}^{N} R(m) \cdots R(1) R(0),$$

for $f \in R$. This makes R into a left H-module algebra.

Theorem (1990)

For all functions $a, b \in R$ and for all labeled trees $h \in k\{\mathcal{LT}(E_1, \ldots, E_M)\}$,

$$((\psi \otimes \psi) \cdot \Delta(h))(a \otimes b) = \psi(h)(ab)$$

We conclude: The ψ map defines an H-module algebra structure on R which makes the diagram commute.

Summary

Space X	**Trees**	**Algebraic Structure**	**Comment**
$X = \mathbf{R}^n$	Labeled Trees	H-Module Algebra	helpful when simplifying action of higher order derivations on functions

Question

- What happens when X is not $\mathbf{R}^N$ and the D_μ defining the derivations

$$E_j = \sum_\mu a_j^\mu(x) D_\mu$$

  do not commute?

Part 4: Simplifying Higher Order Derivations for Diff(G)

This is Problem 1 in the non-commutative case.

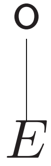
Key Observation - Defining natural H-modules in the non-Abelian case requires using a connection.

Let R be a commutative k-algebra, and let $\mathbf{D}$ be a Lie algebra of derivations of R. A *connection* is a map $\mathbf{D} \times \mathbf{D} \to \mathbf{D}$ sending $(E, F) \in \mathbf{D} \times \mathbf{D}$ to $\nabla_E F \in \mathbf{D}$ satisfying

- $\nabla_{E_1+E_2} F = \nabla_{E_1} F + \nabla_{E_2} F$
- $\nabla_E (F_1 + F_2) = \nabla_E F_1 + \nabla_E F_2$
- $\nabla_{f \cdot E} F = f \cdot \nabla_E F$
- $\nabla_E (f \cdot F) = f \cdot \nabla_E F + E(f) F$

where E, $F \in \mathbf{D}$, $f \in R$.

We use this connection as follows:



$$(r) = E(r)$$

$$(r) = \nabla_F E(r)$$

and extend using induction and a consistency requirement to larger trees.

Theorem (2005)

Let R be a commutative k-algebra, and let $\nabla_E F$ be a connection on the Lie algebra $\mathbf{D}$ of derivations of R. Then the construction above gives a $k\{\mathcal{LT}(\mathbf{D})\}$-module structure on R. This module structure induces a map $\psi : k\{\mathcal{LT}(\mathbf{D})\} \longrightarrow \mathrm{End}(R)$. so that the following diagram commutes:

$$\begin{tikzcd}
k{<}E_1, \ldots, E_M{>} \arrow[r] \arrow[dr] & k\{\mathcal{LT}(\mathbf{D})\} \arrow[d, "\psi"] \\
& \mathrm{Diff}(R) \subset \mathrm{End}(R)
\end{tikzcd} \tag{2}$$

R/k-**Algebras**

Let R be a k-algebra. A R/k-algebra is a k-algebra B into which R is embedded. Note that this makes B into a left and right R-module, and that $(rb)s = r(bs)$ for all r, $s \in R$, $b \in B$, and that also $(rb)c = r(bc)$, $(br)c = b(rc)$, and $(bc)r = b(cr)$ for all b, $c \in B$, $r \in R$.

Example: Diff(Der(R)) is a R/k-algebra.

It turns out that there is a version of the construction above using R/k-algebras and R/k-bialgebras.

R/k-**bialgebra**

A R/k-*bialgebra* is a R/k-algebra B together with R-module maps $\Delta : B \to B \otimes_R B$ and $\epsilon : B \to R$ satisfying

- B together with the maps Δ and ϵ is a coalgebra over R.
- $\Delta(1) = 1 \otimes 1$.
- For all b, $c \in B$, if $\Delta(b) = \sum_i b_i \otimes b'_i$ and $\Delta(c) = \sum_j c_j \otimes c'_j$ are any representations of $\Delta(b)$, $\Delta(c) \in B \otimes_R B$, then $\Delta(bc) = \sum_{i,j} b_i c_j \otimes b'_i c'_j$.
- $\epsilon(1) = 1$.
- $\epsilon(bc) = \epsilon(b\epsilon(c))$.

Theorem. $R \otimes_k k\{\mathcal{LT}(\mathbf{Der}(R))\}$ **is a** R/k**-bialgebra.**

Let R be a commutative k-algebra, let Der(R) denote derivations of R, and let Diff(Der(R)) denote differentional operators generated by Der(R).

- The tensor product $R \otimes_k$ Diff(Der(R)) is $R/k - bialgebra$.
- Let $k\{\mathcal{LT}(\mathrm{Der}(R))\}$ denote the bialgebra of trees labeled with elements of Der(R). Then the tensor product $R \otimes_k k\{\mathcal{LT}(\mathrm{Der}(R))\}$ is a R/k-bialgebra.
- The map

$$R \otimes_k k\{\mathcal{LT}(\mathrm{Der}(R))\} \to R \otimes_k \mathrm{Diff}(\mathrm{Der}(R))$$

is a R/k-bialgebra homomorphism.

Summary

Space X	**Trees**	**Algebraic Structure**	**Comment**
$X = G$ compact group	Labeled, Ordered Trees	R/k-bialgebra	This structures helps with simplifying higher order derivations on groups

Part 5: Computing Flows on Groups G

This is Problem 3 in the non-commutative case.

Computing geometrically stable numerical algorithms.

Goal: Develop a numerical algorithm to approximate flow on a Lie Group G of

$$\dot{x}(t) = E(x(t)), \qquad x(0) = x^0 \in G$$

- $E = \sum_\mu a^\mu(x) Y_\mu$
- $a^\mu(x)$ in coordinate ring R
- Y_μ basis Lie algebra $\mathbf{g}$

Classical Runge Kutta Algorithms:

Consider the differential equation: $x'(t) = E(x(t))$, subject to the initial condition $x(0) = x^0$. Find coefficients $c_{i,j}$ and b_i

$$\begin{aligned}
X_1 &= x_0 \\
X_2 &= x_0 + hc_{2,1}E(X_1) \\
X_3 &= x_0 + hc_{3,1}E(X_1) + hc_{3,2}E(X_2) \\
&\vdots \\
X_r &= x_0 + hc_{r,1}E(X_1) + hc_{r,2}E(X_2) + \cdots + hc_{r,r-1}E(X_{r-1})
\end{aligned}$$

and let

$$x_1 = x_0 + \sum_{i=1}^{r} hb_iE(X_i).$$

so that

$$x_1 - x(h) = O(h^k)$$

Runge Kutta Type Algorithms on Lie Groups:

Fix a point $p \in G$ and consider the ansatz:

$$\begin{aligned}
\bar{E}_1 &= \sum a^\mu(p) Y_\mu \in \mathbf{g} \\
\bar{E}_2 &= \sum a^\mu(\exp(hc_{21}\bar{E}_1 \cdot p)) Y_\mu \in \mathbf{g} \\
\bar{E}_3 &= \sum a^\mu(\exp(hc_{32}\bar{E}_2 \exp(hc_{31}\bar{E}_1 \cdot p))) Y_\mu \in \mathbf{g}
\end{aligned}$$

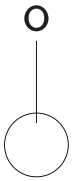
Compute:

$$LHS = \exp hc_3\bar{E}_3 \exp hc_2\bar{E}_2 \exp hc_1\bar{E}_1 - \exp hE = 0 + O(h^4)$$

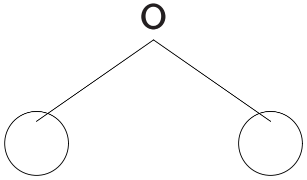
$$LHS = hc_1E + hc_2E + hc_3E$$

$$+h^2c_2c_{21}DE \cdot E + h^2c_3c_{31}DE \cdot E + h^2c_3c_{32}DE \cdot E$$
$$+h^3c_3c_{32}c_{21}DE \cdot DE \cdot E$$
$$+h^3c_2\frac{c_{21}^2}{2!}D^2E(E,E) + h^3c_3\frac{c_{31}^2}{2!}D^2E(E,E)$$
$$+h^3c_3\frac{c_{32}^2}{2!}D^2E(E,E) + h^3c_3c_{31}c_{32}D^2E(E,E)$$
$$-hE - \frac{h^2}{2!}DE \cdot E - \frac{h^3}{3!}D^2E(E,E) - \frac{h^3}{3!}DE \cdot DE \cdot E$$

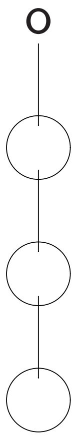
$$LHS = hc_1E + hc_2E + hc_3E$$



$$+h^2c_2c_{21}DE \cdot E + h^2c_3c_{31}DE \cdot E + h^2c_3c_{32}DE \cdot E$$

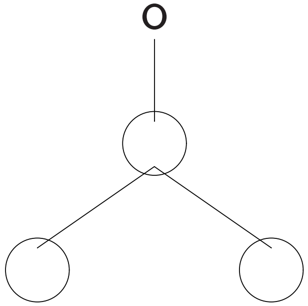


$$+h^3 c_3 c_{32} c_{21} DE \cdot DE \cdot E$$



$$+h^3 c_2 \frac{c_{21}^2}{2!} D^2E(E,E) + h^3 c_3 \frac{c_{31}^2}{2!} D^2E(E,E)$$

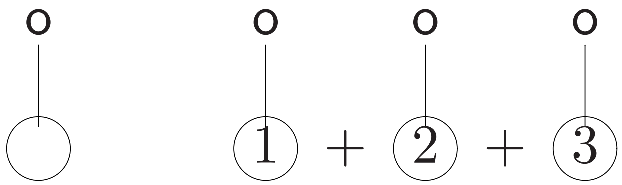
$$h^3 c_3 \frac{c_{32}^2}{2!} D^2E(E,E) + h^3 c_3 c_{31} c_{32} D^2E(E,E)$$



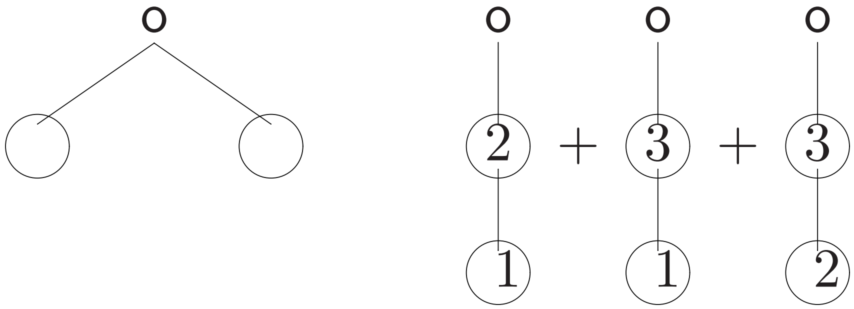
$$-hE - \frac{h^2}{2!} DE \cdot E - \frac{h^3}{3!} D^2E(E,E) - \frac{h^3}{3!} DE \cdot DE \cdot E$$

$$+O(h^4)$$

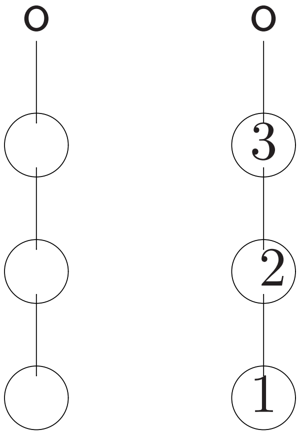
We have:



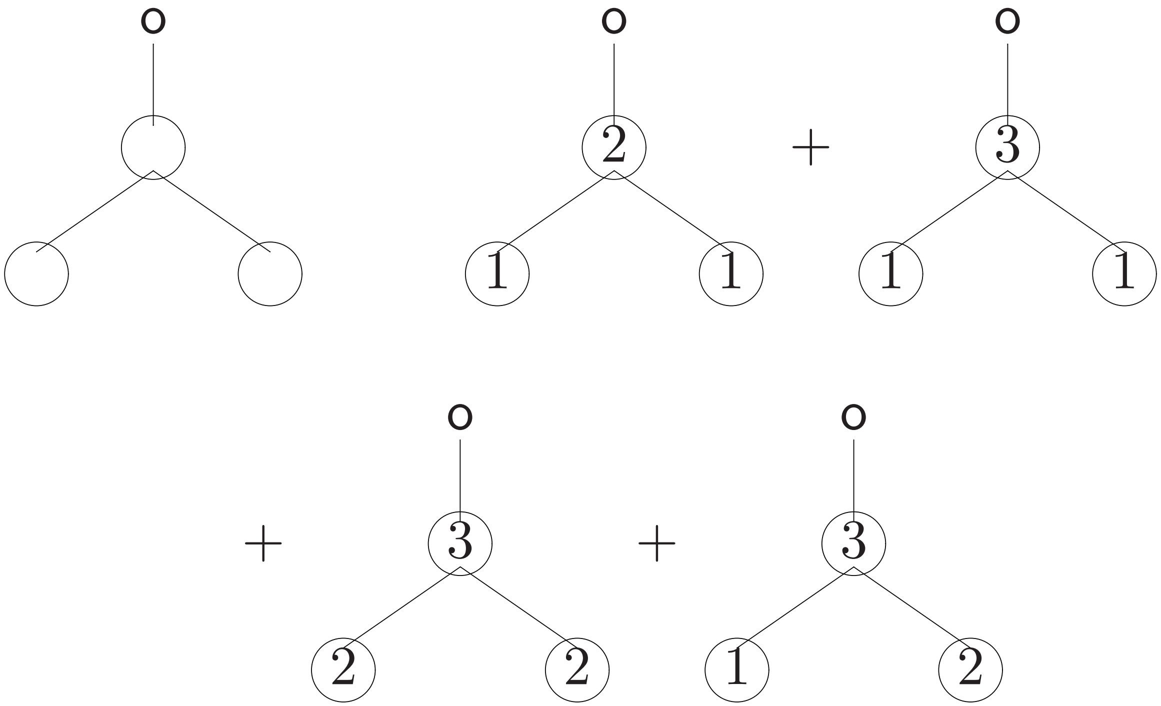
$$c_1 + c_2 + c_3 = 1$$



$$c_2 c_{21} + c_3 c_{31} + c_3 c_{32} = \frac{1}{2!}$$



$$c_3 + c_{32} + c_{21} = \frac{1}{3!}$$



$$\frac{c_2 c_{21}^2}{2!} + \frac{c_3 c_{31}^2}{2!} + \frac{c_3 c_{32}^2}{2!} + c_3 c_{31} c_{32} = \frac{1}{3!}$$

Theorem (1993)

Third order RK algorithms on Lie Groups arise by solving the constraint equations:

1. $c_1 + c_2 + c_3 = 1$
2. $c_2 c_{21} + c_3 c_{31} + c_3 c_{32} = 1$
3. $\frac{c_2 c_{21}^2}{2!} + \frac{c_3}{2!}(c_{31} + c_{32})^2 = 1/3!$
4. $c_3 c_{32} c_{21} = 1/3!$
5. $3(c_2^2 c_{21} + c_3^2(c_{32} + c_{31})) + 6 c_2 c_3 c_{21} = 1$

Summary

Space X	**Trees**	**Algebraic Structure**	**Comment**
$X = G$	RK-Ordered Trees	bialgebra	This codes RK-algorithms on groups.

References

- R. Grossman and R. Larson, Hopf algebraic structures of families of trees, Journal Algebra, Volume 26, 1989, pages 184-210.

- R. Grossman and R. Larson, Hopf-algebraic structure of combinatorial objects and differential operators, Israel Journal Mathematics, Volume 72, 1990, pages 109-117.

- R. Grossman and R. Larson, The symbolic computation of derivations using labeled trees, Journal of Symbolic Computation, Volume 13, pages 511-523, 1992.

- P. Crouch and R. Grossman, Numerical integration of ordinary differential equations on manifolds, Journal of Nonlinear Science, Volume 3, pages 1-33, 1993.

- Robert L. Grossman and Richard G. Larson, Differential Algebra Structures on Families of Trees, Advances in Applied Mathematics, Volume 35, pages 97-119, 2005.

Thank you.