Ames GRANT
IN-64-CR
309746
P-19

Evaluation of expressions involving higher order derivations

Robert Grossman*
University of Illinois at Chicago

May, 1990

Abstract

The local geometric properties of a nonlinear control system defined by vector fields $E_1, \ldots, E_M$ are determined by the algebraic properties of the iterated Lie brackets of the E_j's. Let $R = C^\infty(\mathbf{R}^N)$ and let

$$D = \frac{\partial}{\partial x_\mu}, \quad 1 \leq \mu \leq N.$$

Assume that the vector fields E_j are written in terms of the basis D_μ with coefficients from R:

$$E_j = \sum_{\mu=1}^{N} a_j^\mu D_\mu, \quad \text{where } a_j^\mu \in R, \quad 1 \leq j \leq M.$$

In general, when expressions involving the noncommuting E_j's are written in terms of the commuting D_μ's, there is cancellation. In this paper we examine the problem of rewriting expressions involving the E_j's in terms of the D_μ's in such a way as to handle efficiently any cancellation occurring due to the commuting of the D_μ's. Roughly speaking, we introduce a data structure which allows us to organize the computation to take advantage of the symmetries in the expression and reduce the operation count.

1 Introduction

Given two vector fields E_1 and E_2 on $\mathbf{R}^N$, let $[E_2, E_1] = E_2E_1 - E_1E_2$ denote their Lie bracket. Since at least the late 1960's, the local geometric

*This research was supported in part by NASA grant NAG 2-513.

(NASA-CR-187319) EVALUATION OF EXPRESSIONS INVOLVING HIGHER ORDER DERIVATIONS (Illinois Univ.) 19 p CSCL 12A N91-13949

Unclas
G3/64 0309746

properties of the nonlinear control system

$$\dot{x}(t) = \sum_{\mu=1}^{M} u_\mu(t) E_\mu(x(t)), \quad x(0) = x^0 \in \mathbf{R}^N$$

have been studied by the algebraic properties of the iterated Lie brackets

$$[E_{i_1}, \ldots, [E_{i_{k-1}}, E_{i_k}] \cdots], \quad 1 \leq i_1, \ldots, i_k \leq M$$

of the vector fields $E_1, \ldots, E_M$ and the series and algebras they generate. See, for example, [2] and [3]. Here $t \to u_1(t), \ldots, t \to u_M(t)$ are controls.

This paper is concerned with algorithms for the explicit symbolic computation of iterated Lie brackets and series built from them. In practice, the vector fields E_j will be written in terms of a basis $D_1, \ldots, D_N$ for vector fields on $\mathbf{R}^N$. For example, if we let $R = C^\infty(\mathbf{R}^N)$ and use the basis

$$D_i = \frac{\partial}{\partial x_i}, \quad i = 1, \ldots, N,$$

we can express any smooth vector field on $\mathbf{R}^N$ as a sum

$$E_j = \sum_{\mu=1}^{N} a_j^\mu D_\mu, \quad \text{where } a_j^\mu \in R, \quad 1 \leq j \leq M. \tag{1}$$

To rewrite the expression

$$p = E_3E_2E_1 - E_3E_1E_2 - E_2E_1E_3 + E_1E_2E_3 \tag{2}$$

in terms of the basis D_μ by naively substituting the sum (1) for each occurrence of E_j yields $24N^3$ terms. Because of the symmetry of p, the resulting expression simplifies to yield $6N^3$ terms. In § 5 of this paper, we give an algorithm which computes precisely the $6N^3$ in the simplified expression, without the necessity of computing the other $18N^3$ terms.

The basic idea is to map expressions p involving the vector fields E_j into expressions involving labeled trees. A multiplication on trees is defined corresponding to the composition of vector fields. There is also a map from labeled trees into expressions involving the vector fields D_μ. The composition of these two maps corresponds to the substitution (1), but does not involve the computation of terms which cancel in the end due to the symmetry of the expression p.

The paper is organized as follows: § 2 contains some algebraic preliminaries and a statement of the theorem. Section 3 contains preliminaries about trees, including the definition of the multiplication. Section 4 defines the map from expressions p to trees and § 5 defines the map from trees to expressions involving the vector fields D_μ. Section 6 contains the proof the main theorem.

This is a revision of the Technical Report [1].

Acknowledgements

I want to thank Robert Hermann, Hank Hermes, Arthur Krener, and Clyde Martin for their many suggestions concerning this paper.

2 Derivations

The example above involving the expression (2) concerned vector fields with coefficients in the ring R of C^∞ functions on $\mathbf{R}^N$. This paper considers the vector fields E_j to be derivations of the ring R and considers expressions p to be elements in the free associative algebra generated by the symbols E_1, $\ldots$, E_M.

More generally, let R be a commutative ring with a unit element. A derivation of R is a map D of R to itself satisfying

$$\begin{aligned} D(a+b) &= D(a)+D(b) \\ D(ab) &= aD(b)+bD(a) \qquad \text{for all } a,b \in R. \end{aligned}$$

Let $D_1, \ldots, D_N$ be commuting derivations of R; that is

$$D_i D_j a = D_j D_i a, \quad \text{for all } a \in R, \quad 1 \le i,j \le N.$$

Suppose that we are also given derivations $E_1, \ldots, E_M$ of R which can be expressed as R-linear combinations of the former derivations; that is

$$E_j = \sum_{\mu=1}^{N} a_j^\mu D_\mu, \quad \text{where } a_j^\mu \in R, \quad 1 \le j \le M.$$

Let K denote the subring of constants of R

$$D_1 a = 0, \ldots, D_N a = 0, \quad \text{for all } a \in K,$$

and let

$$K{<}E_1, \ldots, E_M{>}$$

denote the free associative K-algebra in the symbols $E_1, \ldots, E_M$. Note that elements p of the free associative algebra may be thought of as a higher order derivation of R generated by the $E_1, \ldots, E_M$. Let $\mathrm{Diff}(D_1, \ldots, D_N; R)$ denote the space of formal linear differential operators with coefficients from R; that is $\mathrm{Diff}(D_1, \ldots, D_N; R)$ consists of all formal expressions

$$L = \sum_{\mu_1=1}^{N} a_{\mu_1} D_{\mu_1} + \sum_{\mu_1,\mu_2=1}^{N} a_{\mu_1,\mu_2} D_{\mu_2} D_{\mu_1} + \ldots + \sum_{\mu_1,\ldots,\mu_k=1}^{N} a_{\mu_1,\ldots,\mu_k} D_{\mu_k} \cdots D_{\mu_1},$$

where $a_{\mu_1}, a_{\mu_1,\mu_2}, \ldots, a_{\mu_1,\ldots,\mu_k} \in R$. We say that $L \in \mathrm{Diff}(D_1, \ldots, D_N; R)$ and $L' \in \mathrm{Diff}(D_1, \ldots, D_N; R)$ *agree* if

$$L(a) = L'(a), \qquad \text{for all } a \in R.$$

Fix an element $p \in K{<}E_1, \ldots, E_M{>}$ and suppose that p contains l terms, each of which is homogeneous of degree m. If we make the substitution

$$E_j = \sum_{\mu=1}^{N} a_j^{\mu} D_{\mu}, \qquad \text{where } a_j^{\mu} \in R,$$

in the expression p, and use the fact that $D_1, \ldots, D_N$ are derivations of R, we get a differential operator in $\mathrm{Diff}(D_1, \ldots, D_N; R)$. It is easy to see that before any cancellation this differential operator contains $lm!\,N^m$ terms involving $D_1, \ldots, D_N$.

In this paper we ask whether we can find an operator L in $\mathrm{Diff}(D_1, \ldots, D_N; R)$ which agrees with this differential operator but requires fewer operations involving the derivations $D_1, \ldots, D_N$. We show that such an L does indeed exist in case p has a certain symmetry, which we call a symmetry decomposition. In § 6 we give the exact definitions and prove the following theorem.

Theorem 2.1 *Fix an expression $p \in K{<}E_1, \ldots, E_M{>}$ which is homogeneous of degree m. Let $\chi(p) \in \mathrm{Diff}(D_1, \ldots, D_N; R)$ denote the differential operator obtained by the substitution*

$$E_j = \sum_{\mu=1}^{N} a_j^{\mu} D_{\mu}, \quad \text{where } a_j^{\mu} \in R, \quad 1 \le j \le M.$$

Expression p	$\|\chi(p)\|$	$\|L\|$	$\|\chi(p)\| - \|L\|$
$E_2E_1 - E_1E_2$	$4N^2$	$2N^2$	$2N^2$
$E_3E_2E_1 + E_3E_1E_2$	$12N^3$	$6N^3$	$6N^3$
$E_3E_2E_1 - E_3E_1E_2 - E_2E_1E_3 + E_1E_2E_3$	$24N^3$	$6N^3$	$18N^3$

Table 1: Some expressions and the resulting cancellations.

Assume that p has a symmetry decomposition. Then there exists a differential operator L such that (i) L involves cN^m fewer occurrences of terms containing $D_1, \ldots, D_N$ than does the naive computation of $\chi(p)$; (ii) the higher order derivations $\chi(p)$ and L agree. Here c is a constant depending upon the symmetry decomposition.

We end this section with two remarks. First, it is important to note that the algorithm we give in § 6 does not require the explicit identification of the symmetry decomposition; rather, if such a symmetry decomposition is present the differential operator L produced by the algorithm will contain fewer terms than the differential operator $\chi(p)$.

Second, it is quite common for expressions in the E_j's to involve cancellation. Table 2 details the reduction in the number of terms for three different expressions. The last column lists the number of terms which either cancel or combine: these terms need not be computed. We hope to return in a later paper to the classification of those expressions in $E_1, \ldots, E_M$ which result in cancellation or combination when they are written in terms of the $D_1, \ldots, D_N$. In this paper we simply point out that Lie brackets are expessions of this type and that is easy to write down other such expressions.

3 Trees

In this section we describe a data structure that is useful for computations involving higher order derivations. By a tree we always mean a rooted, finite tree. We define the set H_m of ***heap-ordered trees*** or ***heaps*** on m nodes to be the set of all trees h consisting of $m+1$ nodes together with a ***key map***

$$\kappa : \text{nodes } h \longrightarrow \{0, 1, \ldots, m\}$$

satisfying

1. κ is bijective;

2. v a child of w implies $\kappa(v) > \kappa(w)$, where $v, w \in$ nodes h.

From now on we use the map κ to identify a node with its key $j \in \{0, \ldots, m\}$.

The following definition and notation will be used throughout the paper. Let $E_1, \ldots, E_M$ be M arbitrary symbols. Given a heap $h \in H_m$, let

$$h(E_{\gamma_1}, \ldots, E_{\gamma_m})$$

denote the *labeled heap* where the node j carries the label

$$E_{\gamma_j} \in \{E_1, \ldots, E_M\}, \quad 1 \leq j \leq m.$$

Denote the set of all such labeled heaps by

$$LH_m = LH_m(E_1, \ldots, E_M)$$

and put

$$LH = LH(E_1, \ldots, E_M) = \bigcup_{m \geq 0} LH_m(E_1, \ldots, E_M).$$

We say that $h \in LH(E_1, \ldots, E_M)$ is *homogeneous* of degree m in case $h \in LH_m(E_1, \ldots, E_M)$.

We now describe a multiplication on labeled heaps. Let

$$\mathcal{O}(LH(E_1, \ldots, E_M))$$

denote the vector space over K whose basis consists of labeled heaps h in $LH(E_1, \ldots, E_M)$. The product

$$h_2 \cdot h_1 \in \mathcal{O}(LH(E_1, \ldots, E_M))$$

of two labeled heaps

$$\begin{aligned} h_1(E_{\gamma_1}, \ldots, E_{\gamma_{m_1}}) &\in LH_{m_1}(E_1, \ldots, E_M) \\ h_2(E_{\eta_1}, \ldots, E_{\eta_{m_2}}) &\in LH_{m_2}(E_1, \ldots, E_M) \end{aligned}$$

is defined as follows:

Step 1. Recall that each node of heap h_2 has a name $0, 1, \ldots, m_2$ and each node of heap h_1 has a name $0, 1, \ldots, m_1$. Rename the names of the nodes of heap 2 as follows:

old name	1	2	$\cdots$	m_2
new name	$m_1 + 1$	$m_1 + 2$	$\cdots$	$m_1 + m_2$
old label	E_{η_1}	E_{η_2}	$\cdots$	$E_{\eta_{m_2}}$
new label	E_{η_1}	E_{η_2}	$\cdots$	$E_{\eta_{m_2}}$

Keep the names and labels of h_1 the same.

Step 2. Delete the root of h_2. This produces several subtrees $t_1, \ldots, t_l$ with roots $c_1, \ldots, c_l$. Write this as

$$\text{deleteroot } h_2 = \{t_1, \ldots, t_l\}.$$

Step 3. Choose l nodes of h_1, allowing repetition; that is, choose $n_1, \ldots, n_l \in (\text{nodes } h_1)^l$.

Step 4. Form the new tree obtained by linking each root c_j to the node n_j, for $j = 1, \ldots, l$. This is not merely a tree but also a heap, denoted

$$\text{link}(t_1, \ldots, t_l; n_1, \ldots, n_l).$$

Step 5. Sum over all possible choices of nodes $n_1, \ldots, n_l$ in Step 3. This defines the product of the labeled heaps h_1 and h_2:

$$h_2 \cdot h_1 = \bigoplus_{n_1, \ldots, n_l \in (\text{nodes } h_1)^l} \text{link}(t_1, \ldots, t_l; n_1, \ldots, n_l),$$

where deleteroot $h_2 = \{t_1, \ldots, t_l\}$.

Step 6. Finally, extend the product to all of $\mathcal{O}(LH(E_1, \ldots, E_M))$ by K-linearity.

We call the space $\mathcal{O}(LH(E_1, \ldots, E_M))$ the space of *orchards*. We conclude this section by showing that the space of orchards is a K-algebra.

Lemma 3.1 *The space of orchards $\mathcal{O}(LH(E_1, \ldots, E_M))$ is a K-algebra.*

Proof. We need only show that the product of labeled heaps is associative. Recall that a heap is characterized by a table listing the parents of each node. For example

node	parent
0	$\emptyset$
1	0
2	0
3	1
4	3
5	2

is an element of H_5. Notice that one of the nodes (the root) does not have a parent: this is denoted with an $\emptyset$.

node	parent
0	$\emptyset$
1	*
$\vdots$	$\vdots$
m_1	*
– – – – – – – –	– – – – – – – –
(root)	$\emptyset$
$m_1 + 1$	*
$\vdots$	$\vdots$
$m_1 + m_2$	*
– – – – – – – –	– – – – – – – –
(root)	$\emptyset$
$m_1 + m_2 + 1$	*
$\vdots$	$\vdots$
$m_1 + m_2 + m_3$	*

Figure 1: The labeled heaps in $h_3 \cdot h_3 \cdot h_1$.

Let $h_j \in H_{m_j}$, for $j = 1, 2, 3$. The product

$$h_3 \cdot (h_2 \cdot h_1)$$

contains the heaps of Figure 1. Figure 1 uses a number of conventions which we now describe. We use a dashed line to indicate which heaps the nodes belonged to before the product was formed. We say that all the nodes above the first dashed line belong to the first layer; that all the nodes between the two dashed lines belong to the second layer, etc. For example all nodes from the third layer were originally elements of h_3. If a node is enclosed in parentheses, that indicates that it was a node of one of the heaps comprising the product, but was deleted during the formation of the product. As already remarked, a $\emptyset$ indicates that a node has no parent. Finally a * is used to indicate the name of any node higher up in the table. For example the first table says that node 3 is the parent of node 4. With this notation we could replace the 3 with a *, since 3 occurs above 4 in the first column of the table.

We will show that

$$h_3 \cdot (h_2 \cdot h_1) = (h_3 \cdot h_2) \cdot h_1.$$

Given a term $h \in h_3 \cdot (h_2 \cdot h_1)$, we will show that $h \in (h_3 \cdot h_2) \cdot h_1$ also. Notice that the table consists of three layers, corresponding to the three heaps h_1, h_2, h_3, and multiplication of heaps simply consists of replacing any * which is a 0 with the names of all the nodes in the layer(s) above it in the table and forming the corresponding heaps. For example, in computing $h_3 \cdot (h_2 \cdot h_1)$, the *'s in h_3 which are 0 are replaced with the names of nodes in the two layers above. If the node is in the segment immediately above (corresponding to h_2), then it is clear that such a heap is formed when computing $(h_3 \cdot h_2) \cdot h_1$. On the other hand if the node is in the segment corresponding to h_1, then this heap does not correspond to any term in the sum $h_3 \cdot h_2$. In this case the * can first be replaced by 0, corresponding to the root of h_2, and then replaced by the proper node from h_1 when computing $(h_3 \cdot h_2) \cdot h_1$. We have shown that the product is associative. ▮

4 From expressions to orchards

In this section we show how computational problems involving higher order derivations can be translated into problems involving orchards, by defining a map

$$\phi : K\langle E_1, \ldots, E_M\rangle \longrightarrow \mathcal{O}(LH(E_1, \ldots, E_M)).$$

We begin by defining ϕ on monomials $p \in K\langle E_1, \ldots, E_M\rangle$ of the form $E_{\gamma_m} \cdots E_{\gamma_1}$. Put

$$\phi(E_{\gamma_m} \cdots E_{\gamma_1}) = \bigoplus_{h \in LH_m(E_1, \ldots, E_M)} h(E_{\gamma_1}, \ldots, E_{\gamma_m}).$$

Next extend the map ϕ to all of $K\langle E_1, \ldots, E_M\rangle$ by K-linearity. Recall that $h(E_{\gamma_1}, \ldots, E_{\gamma_m})$ denotes the labeled heap with node j labeled with E_{γ_j}.

Lemma 4.1

$$\bigoplus_{h' \in H_m} h' = \bigoplus_{h \in H_{m-1}} \bigoplus_{v \in \mathit{nodes}\ h} \mathit{attach}(m, v)$$

where $attach(m, v)$ is the heap which arises when the node with name m is attached to node v of a heap in H_{m-1}.

Proof. Given a heap $h' \in H_m$, we obtain a heap $h \in H_{m-1}$ by removing the node labeled m. Since the heaps $h \in H_{m-1}$ are distinct, so are the heaps we obtain by attaching the node labeled m to a node v of h. ▮

Theorem 4.1 *The map ϕ is a K-algebra homomorphism.*

Proof. Let $\gamma = (\gamma_1, \ldots, \gamma_{m_1})$ and $\delta = (\delta_1, \ldots, \delta_{m_2})$ and define

$$E_\gamma = E_{\gamma_{m_1}} \cdots E_{\gamma_1}, \quad E_\delta = E_{\delta_{m_2}} \cdots E_{\delta_1}.$$

It is clear that

$$\phi(E_\gamma + E_\delta) = \phi(E_\gamma) + \phi(E_\delta).$$

We need only prove that

$$\phi(E_\gamma \cdot E_\delta) = \phi(E_\gamma) \cdot \phi(E_\delta).$$

We do this by induction on the length of the multi-index γ. If $m_1 = 1$, the assertion follows from Lemma 4.1. Assume the assertion is true for $m = 1, \ldots, m_1$. We compute

$$\begin{aligned}
\phi(E_\gamma) \cdot \phi(E_\delta) &= \phi(E_{\gamma_{m_1+1}} \cdots E_{\gamma_1}) \cdot \phi(E_{\delta m_2} \cdots E_{\delta 1}) \\
&= \phi(E_{\gamma_{m_1+1}}) \cdot \phi(E_{\gamma_{m_1}} \cdots E_{\gamma_1}) \cdot \phi(E_{\delta m_2} \cdots E_{\delta 1}) \\
&= \phi(E_{\gamma_{m_1+1}}) \cdot \phi(E_{\gamma_{m_1}} \cdots E_{\gamma_1} E_{\delta m_2} \cdots E_{\delta 1}) \\
&= \phi(E_{\gamma_{m_1+1}} \cdots E_{\gamma_1} E_{\delta m_2} \cdots E_{\delta 1})
\end{aligned}$$

showing that the assertion holds for $m = m_1 + 1$. ∎

5 From orchards to differential operators

In this section we define a map

$$\psi : \mathcal{O}(LH(E_1, \ldots, E_M)) \longrightarrow \mathrm{Diff}(D_1, \ldots, D_N; R).$$

We do this in several steps.

Step 1. Let $h \in LH_m(E_1, \ldots, E_M)$ and let $k \in$ nodes h, and suppose that $l, \ldots, l'$ are the children of k. Fix $\mu_l, \ldots, \mu_{l'}$ with

$$0 \le \mu_l, \ldots, \mu_{l'} \le N$$

and define

$$\begin{aligned}
R_h(k; \mu_l, \ldots, \mu_{l'}) &= D_{\mu_l} \cdots D_{\mu_{l'}} a_{\gamma_k}^{\mu_k} && \text{if } k \text{ is not the root} \\
&= D_{\mu_l} \cdots D_{\mu_{l'}} && \text{if } k \text{ is the root .}
\end{aligned}$$

We abbreviate this to $R(k)$.

Step 2. Define

$$\psi(h) = \sum_{\mu_1, \ldots, \mu_m} R(m) \cdots R(1)R(0).$$

Step 3. Extend ψ to all $\mathcal{O}(LH(E_1, \ldots, E_M))$ by K-linearity.

Lemma 5.1 *Suppose* $p \in K{<}E_1, \ldots, E_M{>}$ *and*

$$\chi(p) \in Diff(D_1, \ldots, D_N; R)$$

is the corresponding derivation defined by the substitution

$$E_j = \sum_{\mu=1}^{N} a_j^{\mu} D_{\mu}, \quad a_j^{\mu} \in R, \quad 1 \le j \le M.$$

Then

$$\chi(p)(a) = (\psi \circ \phi(p))(a).$$

Proof. We need only prove the lemma when p is a monomial. Let

$$p = E_{\gamma_m} \cdots E_{\gamma_1}$$

We proceed by induction on m. In the case $m = 1$,

$$\phi(E_{\gamma_1}) = \quad {}_{1, E_{\gamma_1}}$$

and

$$\psi(\quad {}_{1, E_{\gamma_1}}) = \sum_{\mu_1=1}^{N} a_{\gamma_1}^{\mu_1} D_{\mu_1},$$

as required.

Assume the lemma holds for $m' = 1, \ldots, m-1$. Then we claim that

$$(\sum_{\mu_m=1}^{N} a_{\gamma_m}^{\mu_m} D_{\mu_m})(\sum_{\mu_{m-1}=1}^{N} a_{\gamma_{m-1}}^{\mu_{m-1}} D_{\mu_{m-1}}) \cdots (\sum_{\mu_1=1}^{N} a_{\gamma_1}^{\mu_1} D_{\mu_1})(a),$$

which is $\chi(p)(a)$, is equal to

$$= \quad (\sum_{\mu_m=1}^{N} a_{\gamma_m}^{\mu_m} D_{\mu_m})(\psi \bigoplus_{h \in LH_{m-1}(E_1, \ldots, E_M)} h(E_{\gamma_1}, \ldots, E_{\gamma_{m-1}}))(a)$$

$$
\begin{aligned}
&= \sum_{\mu_m} a_{\gamma_m}^{\mu_m} D_{\mu_m} \Big(\sum_{h \in LH_{m-1}(E_1, \ldots, E_M)} \psi(h(E_{\gamma_1}, \ldots, E_{\gamma_{m-1}})) \Big)(a) \\
&= \sum_{\mu_m} \sum_{h \in LH_{m-1}(E_1, \ldots, E_M)} a_{\gamma_m}^{\mu_m} D_{\mu_m} (R_h(m-1) \cdots R_h(0))(a) \\
&= \sum_{\mu_m} \sum_{h \in LH_{m-1}(E_1, \ldots, E_M)} a_{\gamma_m}^{\mu_m} (D_{\mu_m} R_h(m-1)) \cdots R_h(0))(a) \\
&+ \sum_{\mu_m} \sum_{h \in LH_{m-1}(E_1, \ldots, E_M)} a_{\gamma_m}^{\mu_m} R_h(m-1) \cdots (D_{\mu_m} R_h(0))(a) \\
&= \sum_{h' \in LH_m(E_1, \ldots, E_M)} R_{h'}(m) \cdots R_{h'}(0)(a) \\
&= \psi(\phi(E_{\gamma_m} \cdots E_{\gamma_1}))(a)
\end{aligned}
$$

Here $h' \in LH_m(E_1, \ldots, E_M)$ is obtained from $h \in LH_{m-1}(E_1, \ldots, E_M)$ by attaching the node labeled m to each node v of h', as in Lemma 3.1. ∎

The map ψ from orchards to $\mathrm{Diff}(D_1, \ldots, D_N; R)$ is not injective. This is because the derivations D_i commute. We conclude this section by describing some elements of the kernel. This requires some additional notation. Let $LT_m(E_1, \ldots, E_M)$ denote the set of all trees consisting of $m+1$ nodes labeled with the symbols $E_1, \ldots, E_M$. As in § 3, let

$$LT(E_1, \ldots, E_M) = \bigcup_{m \geq 0} LT_m(E_1, \ldots, E_M)$$

and let $\mathcal{O}(LT(E_1, \ldots, E_M))$ denote the correpsonding orchard. Let

$$\rho : \mathcal{O}(LH(E_1, \ldots, E_M)) \longrightarrow \mathcal{O}(LT(E_1, \ldots, E_M)),$$

denote the natural map which simply sends a labeled heap-ordered tree to the labeled tree obtained by ignoring the heap-ordering.

Lemma 5.2 *Let $h, h' \in LH_m(E_1, \ldots, E_M)$ be two labeled heaps with the property $\rho(h) = \rho(h')$. Then $\psi(h) = \psi(h')$.*

Proof. We use the notation introduced at the beginning of the section and write

$$
\begin{aligned}
\psi(h(E_{\gamma_1}, \ldots, E_{\gamma_m})) &= \sum R(m; \mu_1, \ldots, \mu_m) \ldots R(0; \mu_1, \ldots, \mu_m) \\
\psi(h'(E_{\eta_1}, \ldots, E_{\eta_m})) &= \sum R(m; \nu_1, \ldots, \nu_m) \ldots R(0; \nu_1, \ldots, \nu_m).
\end{aligned}
$$

Recall that if a node k has children $l, \ldots, l'$, then

$$R(k; \mu_1, \ldots, \mu_m) = D_{\mu_l} \cdots D_{\mu_{l'}} a^{\mu_k}_{\gamma_k}.$$

Observe that the heap names of the nodes simply provide dummy indicies—$\mu_1, \ldots, \mu_m$ for h and $\nu_1, \ldots, \nu_m$ for h'—which are used to write out the differential operators in $\mathrm{Diff}(D_1, \ldots, D_N; R)$ corresponding to the heaps h and h'. Therefore if the underlying labeled trees are the same, then the differential operators in $\mathrm{Diff}(D_1, \ldots, D_N; R)$ will be equal; that is $\rho(h) = \rho(h')$ implies $\psi(h) = \psi(h')$ as asserted. ∎

6 Symmetries of orchards

Let $\sigma \in \mathcal{O}(LH_m(E_1, \ldots, E_M))$ be an orchard on labeled heaps which are homogeneous of degree m. We say that σ has a *symmetry decomposition* in case

1. $\sigma = \sigma_0 + \sigma_1$;
2. $\rho(\sigma_0) = 0$.

In this section, we prove our main theorem by showing how symmetry decompositions can be used to reduce the operation counts of computations involving higher order derivations.

Consider an expression involving higher order derivations p in $K<E_1, \ldots, E_M>$. The substitution (1) produces a differential operator $\chi(p)$ in $\mathrm{Diff}(D_1, \ldots, D_N; R)$. Because of the symmetry of the expression p and the commuting of the D_μ's, the naive computation of $\chi(p)$ often involves some cancellation. It makes sense, therefore, to ask whether there is an algorithm to compute a differential operator $L \in \mathrm{Diff}(D_1, \ldots, D_N; R)$, which agrees with $\chi(p)$ and which does not require the computation of terms which cancel in the end. We will show that this is the case under the assumption that there is a symmetry decomposition of the corresponding orchard.

To state the theorem requires a final definition. If the orchard σ can be written

$$\sigma = \sum_h \sigma(h) h, \quad \text{where } \sigma(h) \in K, \quad \text{and } h \in LH_m(E_1, \ldots, E_M),$$

then let $|\sigma|$ denote the number of heaps h such that the corresponding coefficient $\sigma(h)$ is nonzero.

Theorem 6.1 *Fix an expression $p \in K<E_1,\ldots,E_M>$ which is homogeneous of degree m and let*

$$\sigma = \phi(p) \in \mathcal{O}(LH_m(E_1,\ldots,E_M))$$

be the corresponding orchard. Let $\chi(p) \in Diff(D_1,\ldots,D_N;R)$ be the differential operator obtained by the substitution

$$E_j = \sum_{\mu=1}^{N} a_j^{\mu} D_{\mu}, \quad a_j^{\mu} \in R, \quad 1 \leq j \leq M.$$

Assume that $\sigma = \sigma_0 + \sigma_1$ is a symmetry decomposition of σ and let $L = \psi(\sigma_1)$. Then (i) L involves

$$|\sigma_0|\, N^m$$

fewer occurrences of terms containing $D_1,\ldots,D_N$ than does $\chi(p)$ and (ii) the higher order derivations $\chi(p)$ and L agree.

Proof. All the work has already been done. Since

$$\sigma = \sigma_0 + \sigma_1$$

is a symmetry decomposition, we have $\rho(\sigma_0) = 0$. Hence by Lemma 5.2, $\psi(\sigma_0) = 0$, and there will be $|\sigma_0|\, N^m$ fewer occurrences of terms containing $D_1,\ldots,D_N$. By Lemma 5.1 $\chi(p)$ and L agree. This proves the theorem. ∎

Theorem 1.1 is an immediate corollary of this theorem.

7 Example

In this section we present a simple example of a computation of a third order derivation in terms of orchards. Fix three derivations

$$E_j = \sum_{\mu=1}^{N} a_j^{\mu} D_{\mu}, \quad \text{where } a_j^{\mu} \in R, \quad 1 \leq j \leq M, \quad j = 1,2,3.$$

Consider a higher order derivation of the form

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

The naive computation of $\chi(p)$ requires computing $24N^3$ terms of the form described in Table 2.

No. of terms	Form of terms
$8N^3$	coeff. D_{μ_1}
$12N^3$	coeff. $D_{\mu_2}D_{\mu_1}$
$4N^3$	coeff. $D_{\mu_3}D_{\mu_2}D_{\mu_1}$

Table 2: Naive computation of $\chi(p)$.

No. of terms	Form of terms
$2N^3$	coeff. D_{μ_1}
$12N^3$	coeff. $D_{\mu_2}D_{\mu_1}$
$4N^3$	coeff. $D_{\mu_3}D_{\mu_2}D_{\mu_1}$

Table 3: Terms in the computation of $\chi(p)$ which cancel.

The orchard $\sigma = \phi(p)$ contains 24 labeled heaps, six for each of the four terms of p. For example, the six labeled heaps corresponding to the first term are given in Figure 2. The orchard σ has the symmetry decomposition $\sigma_0 + \sigma_1$, where the labeled heaps corresponding to σ_1 are given in Figure 3. The orchard σ_0 therefore contains the remaining 18 labeled heaps of σ. An example of the cancellation of labeled heaps is given in Figure 4. The differential operator $L = \psi(\sigma_1)$ is equal to

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

and contains $18N^3$ fewer terms of the form indicated in Table 3 than does the naive computation of $\chi(p)$.

References

[1] R. Grossman, "Evaluation of expressions involving higher order derivations," *Center For Pure and Applied Mathematics, University of California at Berkeley*, PAM-367, 1987.

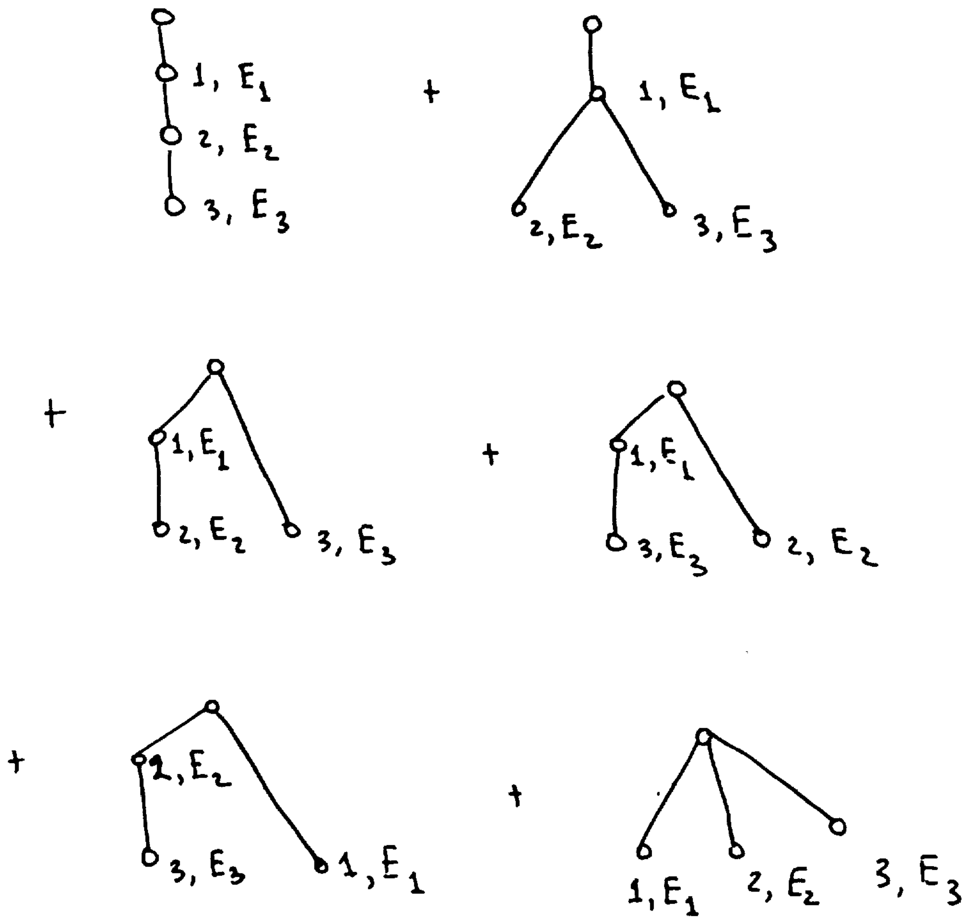

Figure 2: The labeled heaps $\phi(E_3E_2E_1)$.

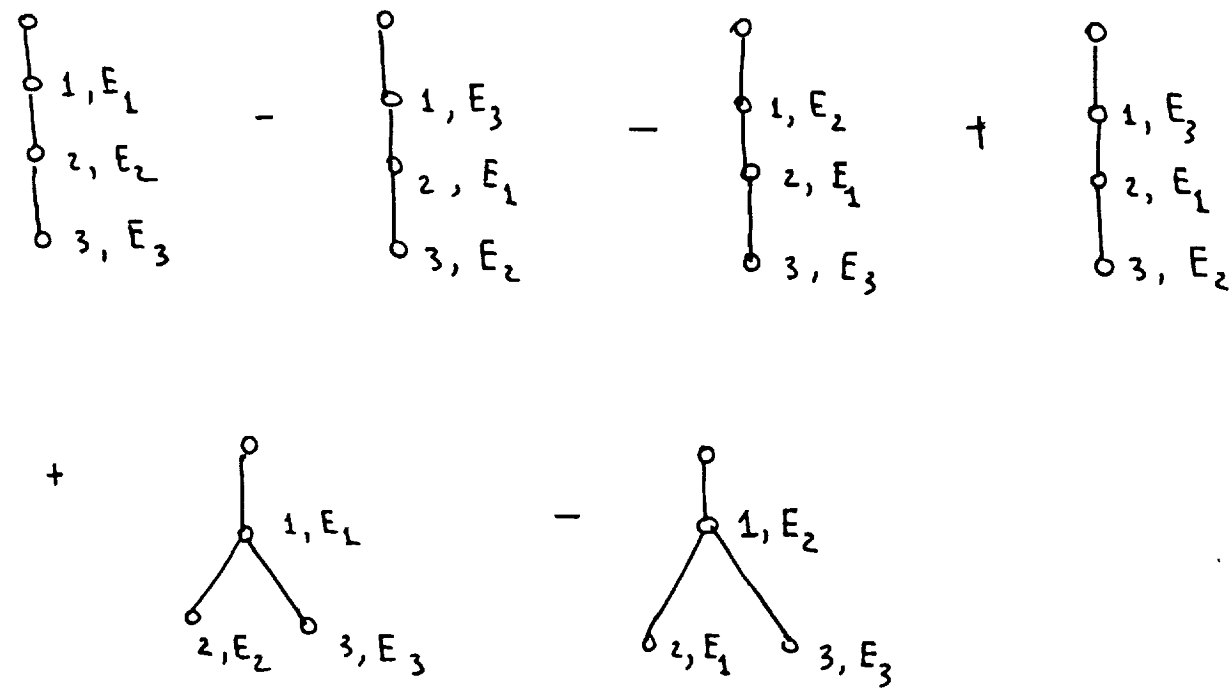

Figure 3: The labeled heaps of σ_1.

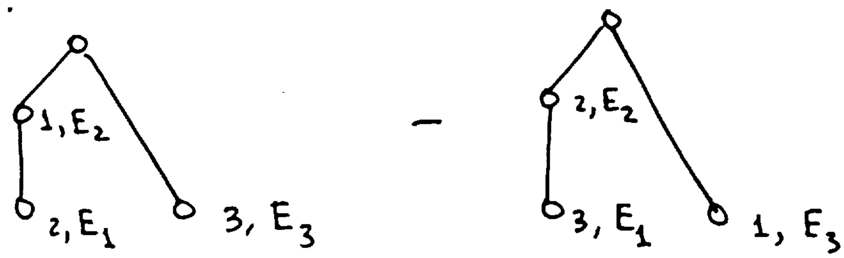

Figure 4: The term $E_3E_1E_2$ contributes the first labeled heap and the term $E_1E_2E_3$ contributes the second, which cancel.

[2] G. W. Haynes and H. Hermes, "Nonlinear controllability via Lie theory," *SIAM J. Control Optimization.*, 8, 1970, pp. 450–460.

[3] R. Hermann and A. J. Krener, "Nonlinear controllability and observability," *IEEE Trans. Automatic Control,* AC-22, pp. 728–740.

[4] H. Hermes, "Nilpotent approximations of control systems and distributions," *SIAM J. Control Optimization,* 24 1986, pp. 731-736.

[5] R. E. Tarjan, *Data Structures and Network Algorithms,* SIAM, Philadelphia, 1983.