KATSUYUKI SHIBATA

Micro-computer prolog as a handy tool for formal algebraic computations

Astérisque, tome 192 (1990), p. 69-78

<http://www.numdam.org/item?id=AST_1990__192__69_0>

MICRO-COMPUTER PROLOG AS A HANDY TOOL FOR FORMAL ALGEBRAIC COMPUTATIONS

Katsuyuki Shibata

Prolog is a logic programming language, and its grammer is based on the first order predicate logic. It has in itself an inference mechanism which runs automatically. Since logic and mathematics are near in a naive sense, it is not surprising that sometimes the translation from mathematical formulas to a Prolog program is straightforward and that they look very similar. I will shortly show it by explicit examples.

From this point of view, Prolog seems to be a very good language for those mathematicians who are not specialists of computer sciences and who are not so much interested in learning the details of computer mechanisms.

But let me first explain about the Gelfand-Fuks cohomology of a smooth manifold. My experience on micro-computer Prolog was to compute a part of that cohomology in the sphere case.

§ 1. Gelfand-Fuks cohomology

Let M be a paracompact Hausdorff smooth manifold and $\mathcal{L}_M$ be the Lie algebra of smooth tangent vector fields on M equipped with C^∞-topology. ($\mathcal{L}_M$ is also denoted as $\mathfrak{X}(M)$ or as Vect(M).) And let $C_c^\bullet(\mathcal{L}_M) = \bigoplus_{q>0} C_c^q(\mathcal{L}_M)$ be the Koszul complex of continuous cochains of the topological Lie algebra $\mathcal{L}_M$. Namely the graded vector space $C_c^q(\mathcal{L}_M)$ is defined to be the set of all the continuous, alternating q-linear forms

S.M.F.
Astérisque 192 (1990)

$$f : \mathcal{L}_M \times \ldots \times \mathcal{L}_M \dashrightarrow \mathbf{R} .$$
$$(\text{ q times })$$

And the differential $d : C_c^q(\mathcal{L}_M) \longrightarrow C_c^{q+1}(\mathcal{L}_M)$ is defined by the formula

$$(df)(X_1, X_2, \ldots X_{q+1}) = \sum_{1 < i < j < q+1} (-1)^{i+j} f([X_i, X_j], X_1, \ldots \overset{i}{\vee} \ldots \overset{j}{\vee} \ldots, X_{q+1}).$$

By the Jacobi identity of Lie algebra, $d \circ d \equiv 0$ holds and we can take the cohomology of $C_c^{\bullet}(\mathcal{L}_M)$ with respect to d.

DEFINITION. <u>The Gelfand-Fuks cohomology</u> $H_c^{\bullet}(\mathcal{L}_M)$ of the manifold M is defined to be the cohomology $H^{\bullet}(C_c^{\bullet}(\mathcal{L}_M);d)$.

The Gelfand-Fuks cohomology is related to the theory of exotic characteristic classes of foliations and is interesting in various aspects.

Gelfand and Fuks proved, among other things, the following finiteness theorem for the additive structure of $H_c^{\bullet}(\mathcal{L}_M)$.

THEOREM (Gelfand-Fuks [1]).

<u>If</u> $\dim_{\mathbf{R}}(H^{\bullet}(M;\mathbf{R})) < \infty$, <u>then</u> $\dim_{\mathbf{R}}(H_c^q(\mathcal{L}_M)) < \infty$ <u>for every</u> q.

In contrast to this, we have proved the following theorem concerning the multiplicative structure of $H_c^{\bullet}(\mathcal{L}_M)$.

THEOREM (Shibata [4]).

<u>If</u> $\dim_{\mathbf{R}}(H^{\bullet}(M;\mathbf{R})) < \infty$, <u>then</u> $H_c^{\bullet}(\mathcal{L}_M)$ <u>is finitely generated as an</u> **R**-<u>al-gebra if and only if either of the following two conditions holds</u>;

(1) $\dim M \leq 1$ (<u>ie</u>. M = <u>a finite union of</u> {pt}, S^1, <u>and</u> $\mathbf{R}^1$), <u>or</u>

(2) $\bar{H}^{\bullet}(M;\mathbf{Q}) \equiv 0$ (<u>ie</u>. M <u>is rationally acyclic</u>).

For the proof of this theorem, we computed Haefliger's model for $C_c^{\bullet}($

$\mathcal{L}_M$) constructed by using Sullivan's minimal model theory in rational homotopy theory. This computation was done by hand, but later we became interested in using a computer for computations in explicit examples.

In case M is the n-dimensional sphere S^n, Haefliger's model reduces to the Koszul cochain complex $C^\cdot(H^\cdot(S^n;\mathbf{R})\otimes L(V_n))$ of Lie algebra $H^\cdot(S^n;\mathbf{R})\otimes L(V_n)$, where V_n is a certain finite dimensional graded vector space depending on n, $L(V_n)$ is the free graded Lie algebra over V_n, and the Lie product in the above tensor product is defined as

$$[x\otimes\ell, x'\otimes\ell'] = \pm xx'\otimes[\ell,\ell'].$$

The ordinary (not necessarily continuous but all cochains) cohomology of this Lie algebra is isomorphic to $H_c^\cdot(\mathcal{L}_{S^n})$ (Haefliger [2]).

We now know that this cohomology algebra is not finitely generated if $n \geqq 2$, but our knowlege is far from complete. There are too many multiplicative generators. Therefore we are interested in computing the cohomology of the Lie algebra $H^\cdot(S^n;\mathbf{R})\otimes L(V_n)$.

To begin with, we must know in detail the product structure of a free Lie algebra. To avoid tedeous sign calculations, I neglect the odd degree elements of Lie algebra in the explanations of the following sections.

§ 2. Hall basis criteria program

Let V be a vector space (over **Q** or $\mathbf{R}$) and $B = \{x_1, x_2, \ldots\}$ be a well-ordered basis of V.

DEFINITION. A well-ordered subset $H \subset L(V)$ is <u>a Hall set relative to</u> B if

(H-1). $H = \bigcup_{n\geqq 1} H_n$, where H_n consists of elements of length n, $H_1 = B$, and the ordering in H satisfies the condition

$x < y$ if (length x) $<$ (length y),

(H-2). $H_2 = \{[x,y];\ x,y \in B,\ x < y\}$, and

(H-3). $\bigcup_{n\geq 3} H_n = \{[Y,[X,Z]];\ X,Y,Z,[X,Z] \in H,\ Y \geq X,\ Y < [X,Y]\}$.

It is known that a Hall set is an additive basis for $L(V)$.

Given a Lie product element in $L(V)$, we want to know whether it belongs to H or not. I am going to write down a Prolog program for that.

For simplicity's sake, I treat only the case where $\dim V = 2$. The case for $\dim V = n > 2$ is completely analogous. So let us assume $B = \{x,y\}$ with $x < y$.

```
          /* Hall basis criteria program */
hall_basis(x).
hall_basis(y).
hall_basis([x,y]).
hall_basis([Y,[X,Z]]):-
     hall_basis(X), hall_basis(Y), hall_basis(Z), hall_basis([X,Z]),
     (Y=X ; smaller(X,Y)), smaller(Y,[X,Z]).
smaller(x,y).
smaller(X,Y):- lie_length(X,M), lie_length(Y,N),
     (M<N ;
       (M=N, X=[U,V], Y=[W,Z], (smaller(V,Z) ; (V=Z, smaller(U,W))))).
lie_length(x,1).
lie_length(y,1).
lie_length([X,Y],N):- lie_length(X,L), lie_length(Y,M), N is L+M.
```

In Prolog, each <u>logical line</u> ends with a full stop. A logical line may be written in several <u>phisical lines</u> if it is long. Each logical line is called <u>a Horn clause</u>. (Horn is the name of a logician.) There are two kinds of Horn clauses; those containing the symbol ":-" and those without it. The symbol ":-" means the logical "if", and "A:-B" means "<u>statement</u> A <u>is true if statement</u> B <u>is true.</u>" This type of Horn