Computing Nilpotent Quotients in Finitely Presented Lie Rings

Csaba Schneider

1995 August

Abstract

A nilpotent quotient algorithm for finitely presented Lie rings over $\mathbf{Z}$ is described. The paper studies the graded and non-graded cases separetely. Computationan details are provided and implementation in C is available.

1 Introduction

The nilpotent quotient algorithm for finitely presented Lie rings operates with Lie rings over $\mathbf{Z}$. Its essential goal is to compute a sufficient and useful presentation for a given Lie ring. It actually means to compute a presentation for its abelian group and to compute the structure constants in order to determine the Lie ring structure. By efficient and useful we mean that several important pieces of information can be read off immediately(e.g.: nilpotency, nilpotency class etc). Furthermore, by this presentation the so-called *word problem* is decidable, although it is known to be undecidable in the most general case.

Such algorithms have been existing for several decades for groups and Lie rings as well. Recall, that groups and Lie rings have similar structure if we think the commutator as a second operation in a group. It has a Jacobi-like property and an identity similar to the distributivity and it is anticommutative (see [8]). This allows us to alter the known group algorithms for Lie rings.

The first widely known (and maybe the most successful) algorithm is due to Havas and Newman (see [2]). It has several implementation, in FORTRAN, C, and GAP (see [1]).

From our point of view, probably the most important algorithms are those of Havas, Newman and Vaughan-Lee (see [3]) and Nickel (see [6]). The algorithm described in this paper is a mixture of the latter mentioned ones, however it differs at several points from those.

This paper contains a brief description of the presentation that we want to obtain in the second section, a theoretic description of the algorithm in the third section, also to emphasize the difference in the graded case in the fourth section, the fifth section is devoted to the implementation, while in the very last—the sixth—one shows a concrete computational example.

Note that throughout this paper we deal with finitely presented Lie rings and the fact which allows us to build a nilpotent quotient algorithm is presented in

Lemma 1.1. *In a finitely presented Lie ring each lower central factor is finitely presented.*

The proof can be done by a similar argument that is presented in [8] 5.2.6, but one might want to see [9] for more details.

2 The Nilpotent Presentation of a Lie Ring

Let L denote a Lie ring and L^i the ith term of its lower central series. If L is finitely presented Lemma 1.1 implies that L^i/L^{i+1} is finitely presented for each i. Since at the present stage we are only interested in the nilpotent factor L/L^k, we think L is nilpotent of class $k-1$. In other words we think L/L^k equals to L. In this case L has a so-called *nilpotent presentation*, that is a tuple $(\mathcal{G}, S)$, where $\mathcal{G}$ is a finite set of generators $\{a_1, \ldots, a_r\}$, say, and S is a finite set of relators of the form

$$o_i \cdot a_i = \alpha_{i,i+1} \cdot a_{i+1} + \cdots + \alpha_{i,r} \cdot a_r \quad \text{for some } i, \qquad \text{TR}$$

$$[a_j, a_i] = \beta_{j,i,j+1} \cdot a_{j+1} + \cdots + \beta_{j,i,r} \quad \text{for each } j > i. \qquad \text{PR}$$

Note that the relations of type TR are sometimes referred as *torsion relations* while those of PR are known as *product relations*. A more detailed description and the proof for existence of such presentation can be found in [9], while it is easy to see, that a nilpotent presentation always defines a nilpotent Lie ring. Observe that we demand the existence of TR relations for each pair (j,i), where $j > i$, but only some of those of type PR (even possibly none). Throughout this paper I denotes the set of indices, so that $i \in I$ iff a_i has TR relation.

One easily sees that the relations of type TR determine the underlying abelian group structure for L, while the relations of type PR define its ring structure. The β can be viewed as structure constants for the $\mathbf{Z}$-module defined by $\mathcal{G}$ and the TR relations.

For our purpose this concept will be too general, so we shall keep some restrictions. Define a weight function $\omega : \mathcal{G} \to \mathbf{N}$ as follows. Let ω be an increasing function in i, such that the following holds.

1. $\omega(a_1) = 1$,
2. if the nilpotent presentation contains a relation of the form

   $$[a_j, a_i] = w_{ji},$$

   where w_{ji} is a sum of integer multiples of generators from $\mathcal{G}$, then all generators $a_k \in \mathcal{G}$ such that $\omega(a_k) < \omega(a_j) + \omega(a_i)$ have coefficients zero in w_{ji}.

An other restriction, if $a_k \in \mathcal{G}$ and $\omega(a_k) > 1$, we require a_k to have a definition, i.e. a relation of the form

$$[a_j, a_i] = a_k \qquad (2.1)$$

where $\omega(a_k) = \omega(a_j) + \omega(a_i)$ and $\omega(a_i) = 1$. It might easily happen that a nilpotent presentation contains more then one relations of the form (2.1) for some i, in this case we arbitrarily choose one for the definition for a_i.

From now on by a nilpotent presentation we shall always mean weighted nilpotent presentation which satisfies the condition stated above for the existence of definitions.

Let $\ell \in L$, it is easy to see, that ℓ can be written in normal form with respect to the given nilpotent presentation, i.e.

$$\ell = \alpha_1 \cdot a_1 + \cdots + \alpha_r \cdot a_r \tag{2.2}$$

where $0 \leq \alpha_i < o_i$ whenever $i \in I$. An arbitrary ℓ might have more than one normal form of type (2.2). This leads to the following

Definition 2.1. *A nilpotent presentation is said to be consistent if every element of L uniquely has a normal form.*

In general, an equivalent condition of that of Definition 2.1 is that the element 0 uniquely has a normal form.

From the computational point of view it is important to see that all the relations of type PR are not necessary to determine the structure of L. Our first result is devoted to this observation.

Lemma 2.2. *Let L be a Lie ring given by the nilpotent presentation*

$$\langle\, \mathcal{G} \,|\, o_i \cdot a_i = w_i,\ i \in I, [a_j, a_i] = w_{ji},\ 1 \leq i < j \leq r \,\rangle.$$

Then L is already determined by the presentation

$$\langle\, \mathcal{G} \,|\, i_i \cdot a_i = w_i,\ i \in I,\ [a_j, a_i] = w_{ji},\ 1 \leq i < j \leq r,\ \omega(a_i) = 1 \,\rangle. \tag{2.3}$$

Proof. Essentially we want to prove that all PR relations can be expressed by those listed in (2.3). We proceed by an induction argument on $\omega(a_i)$. If $\omega(a_i) = 1$ then we are done. Suppose now,that $\omega(a_i) = k > 1$. Using the definition for a_i and the identities of a Lie ring one has: $[a_j, a_i] = [a_j, [a_k, a_l]] = -[a_k, a_l, a_j] = [a_l, a_j, a_k] + [a_j, a_k, a_l] = [a_j, a_k, a_l] - [a_j, a_l, a_k]$. And the latter ones are known by the induction hypothesis. In the first equality we substituted a_i by its definition.

3 Computing the Nilpotent Presentation

The aim of this section is to show how it is possible to compute a consistent nilpotent presentation for a nilpotent factor of a a given finitely presented Lie ring L and determine an epimorphism from L onto that nilpotent factor.

The basic ideas—besides being simple—are similar to those used in [3] and [6]. In fact, it is based on an induction argument. Suppose, we are given a consistent nilpotent presentation for L/L^i and an epimorphism from L onto L/L^i, we shall show how to extend them to a consistent presentation and an epimorphism for L/L^{i+1}.

As computational tools, matrices over $\mathbf{Z}$ play an important rôle. Computing the *row Hermite normal form* is a crucial point of the algorithm. Since its details can be found in [6], [9], [10], we do not emphasize them further in this paper.

3.1 Computing the Abelian Factor

Suppose that we are given a Lie ring L with finite presentation $\langle \mathcal{X} \mid \mathcal{R} \rangle$, where $\mathcal{X}$ is a finite set of generators, while $\mathcal{R}$ is a finite set of relators. For simplicity suppose $\mathcal{X} = \{x_1, \ldots, x_n\}$. Set $\mathcal{G} = \{a_1, \ldots, a_n\}$. Construct a map $\phi : \mathcal{X} \to \mathcal{G}$, by simply saying

$$x_i\phi = g_i.$$

Furthermore we set all PR relations to be trivial and $I = \emptyset$. We call the presentation constructed above *trivial nilpotent presentation* and it is clear that it represents the free abelian Lie ring $\mathcal{F}_n^{ab}$ on n generators. It is straightforward to see that it contains L/L^2 as a factor ring. Indeed, let $\mathcal{I}$ be the ideal generated by the set $\mathcal{R}\phi$ where we evaluate ϕ as if it was a homomorphism elementwise on $\mathcal{R}$. Then we have

Lemma 3.1. *L/L^2 is isomorphic to $\mathcal{F}_n^{ab}/\mathcal{I}$.*

Proof. Both Lie rings are generated by n elements and satisfy exactly the same relations. It checks the isomorphic property.

What we do in practice is the following. Evaluate the relations in $\mathcal{F}_n^{ab}$ as described above and put the images in a matrix M. Then compute the row Hermite normal form M^H for M. It is well known that the rows of M generate the same subgroup in the free abelian group, as those of M^H. If the first non-zero element, the ith say, of a row of M^H is 1, i.e. we have a row of the form

$$(0, \ldots, 0, 1, m_{i+1}, \ldots, m_n)$$

we simply throw a_i out of $\mathcal{G}$ and modify the map ϕ

$$x_i\phi = -m_{i+1} \cdot a_{i+1} - \cdots - m_n \cdot a_n.$$

Otherwise, if the leading element is greater then one, i.e. we have a row of the form

$$(0, \ldots, 0, m_i > 1, m_{i+1}, \ldots, m_n),$$

we introduce a TR relation

$$m_i \cdot a_i = -m_{i+1} \cdot a_{i+1} - \cdots - m_n \cdot a_n.$$

and put i into I.

Proceeding each row as described above we obtain a presentation for L/L^2. We set the surviving generators of weight one and it is straightforward to see, that the presentation obtained so far is indeed a presentation for the abelian factor and ϕ extends to an epimorhism. We summarize this result in

Proposition 3.2. *The so obtained presentation is a consistent nilpotent presentation for L/L^2 and ϕ extends to a epimorphism from L onto L/L^2.*

The untouched images under epimorphism will be referred as definitions of generators of weight one. This terminology will be important later on.

3.2 Extending the Presentation

Now we shall see how the induction argument works in details. Suppose, we have a nilpotent presentation for the factor ring L/L^i, an epimorphism from L onto L/L^i. Extending the presentation will consist of three steps, i.e.

1. extending the epimorphism ϕ,
2. modifying the TR relations,
3. modifying the PR relations.

Extending the epimorphism means introducing a new generator for all $x_i \in \mathcal{X}$ where $x_i\phi$ is not a definition. In other words, if $x_i\phi = w_i$, we modify $x_i\phi$ so that $x_i\phi = w_i + t_i$.

The torsion relations will change, so that if $i \in I$, $o_i \cdot a_i = w_{ii}$ then we alter $o_i \cdot a_i = w_{ii} + t_{ii}$.

Those of the PR relations that are not definitions are modified in a similar way. If $[a_j, a_i] = w_{ji}$ $j > i$ and $[a_j, a_i]$ is not a definition, let the new relation be $[a_j, a_i] = w_{ji} + t_{ji}$.

Note, that all newly introduced generators t_i, t_{ii}, t_{ji} are different from one another. We introduce PR relations so that all of them are central. At this stage we do not alter I. We prove the following

Proposition 3.3. *The so extended presentation contains L/L^{i+1} as a factor ring.*

Proof. Since we did not alter the definitions, every relation holding in the extended presentation is satisfied in L/L^{i+1}.

In practice we use Lemma 2.2. This lemma essentially says, that we do not need to introduce new generators for all PR relations, only for those of the form $[a_j, a_i] = w_{ji}$ for $j > i$ and a_i is of weight one. The remaining ones can be computed as we have seen in the proof of this lemma. This computation is usually referred to as *computing the tails*.

3.3 Enforcing Consistency

Recall, we called a nilpotent presentation consistent if each element in L uniquely had a normal form. The extended presentation investigated in the previous subsection might happen to be not consistent as the case is in general.

In order to investigate the uniqueness property we introduce an operation on the nilpotent quotient. First of all let A be an abelian group generated by $\mathcal{G}$ and the TR relations of the extended presentation. Introduce a map ψ such that,

$$\psi : \mathcal{G} \times \mathcal{G} \to A \qquad \psi(a_j, a_i) = \begin{cases} w_{ji} & \text{if } j > i, \\ -w_{ji} & \text{if } j < i, \\ 0 & \text{if } j = i \end{cases}$$

provided that the nilpotent presentation possesses a PR relation of the form $[a_j, a_i] = w_{ji}$ for $j > i$. Then the following is true.

Proposition 3.4. *ψ can be extended to a binary operation on A iff*

$$o_j \cdot [a_j, a_i] = \sum_{k=j+1}^{n} \alpha_{jk}[a_k, a_i] \qquad C1$$

where $j \in I$ and the relation corresponding to a_j is of the form

$$o_j \cdot a_j = \alpha_{j,j+1} a_{j+1} + \cdots + \alpha_{in} \cdot a_n.$$

Proof. It is immediate that the condition is necessary. To see that sufficient as well, observe if two expressions, w and w' say, equal, it means that they can be transformed to each other by using TR relations. The condition of the proposition claims that such a transition respects the equality $\psi(w, v) = \psi(w', v)$ for any v. Similar result is obtained in the second variable of ψ by swapping the two variables.

Equipped A by that operation A happens to be a non-associative ring. We prove the following

Lemma 3.5. *The nilpotent presentation of L is consistent iff A is a Lie ring.*

Proof. Observe, A contains L as a factor ring in a non-associative ring, furthermore $A = L$ iff the Jacobi identity holds in A, in other words $A = L$ iff A is a Lie ring.

One can easily check the Jacobi identity, since it is a multilinear identity, thus it is enough to check it on the triples formed by the abelian group generators. However we can restrict ourselves even more by the following

Lemma 3.6. *A nilpotent presentation $(\mathcal{G}, S)$ of a Lie ring L is consistent iff the Jacobi identity holds for each triple $(a_i, a_j, a_k) \in \mathcal{G} \times \mathcal{G} \times \mathcal{G}$ where $1 \leq i < j < k \leq n$ and a_i is of weight one. In other words*

$$[a_i, a_j, a_k] + [a_j, a_k, a_i] + [a_k, a_i, a_j] = 0 \qquad C2$$

for $1 \leq i < j < k \leq n$ and a_i is of weight one.

For the proof we refer [3].

3.4 Enforcing the Defining Relations

One easily observes that the presentation obtained so far is a presentation of the freest Lie ring that happens to contain L/L^{i+1} as a factor ring in it. What is still needed is to assure that the images of the elements of $\mathcal{R}$ under the epimorphism are 0. It simply means taking the factor ring over the ideal generated by the epimorphic images. By the induction hypothesis those images vanish in L/L^i, so they must lie in L/L^{i+1} and we are allowed to use the Hermite normal form method again.

In practice we proceed as follows. Put all elements obtained from the consistency relations (C1), (C2) and the defining relations together in a matrix M. Compute the row Hermite normal form M^H for M. Using those rows of M^H whose leading element equals to 1 we can eliminate some of the generators from the nilpotent presentation. The other rows express linear dependence among the generators over $\mathbf{Z}$. They can be viewed as TR relations and added to the presentation. Extend the weight function to the newly introduced generators by saying that they are of weight i and add the the indices of generators that were provided with TR relations to I. The following lemma makes sure that generators of weight i have definitions.

Lemma 3.7. *If L is a Lie ring and suppose that L/L^2 is additively generated as*

$$L/L^2 = \langle \ell_1 + L^2, \ldots, \ell_k + L^2 \rangle$$

and L^{i-1}/L^i is additively generated as

$$L^{i-1}/L^i = \langle {\ell_1}' + L^i, \ldots, \ell^{l'} + L^i \rangle$$

then L^i/L^{i+1} is additively generated as

$$L^i/L^{i+1} = \langle [\ell_i', \ell_j] + L^{i+1} \,|\, 1 \le i \le l,\, 1 \le j \le k \,\rangle.$$

The proof of the above lemma can be done essentially in the same way as in the group case. That proof can be found in [10] Proposition 2.6. But see also [9] for detailed study of the Lie ring case.

4 Computing Graded Lie Rings

Recall, a Lie ring is called graded if it is a *direct* sum $L = \oplus L^i$ such that $[L^i, L^j] \le L^{i+j}$. Since a Lie ring of that kind possesses a relatively easy structure we are allowed to simplify our algorithm in this case.

First of all we need not extend the TR relations any more since if $a_i \in L^l - L^{l+1}$ for some $i \in I$ implies $o_i \cdot a_i \in L^l - L^{l+1}$, so the right hand side of the relation corresponding to a_i cannot contain generators of weight greater than l.

Similar thing happens when we extend the PR relations. At the lth step we only alter the relatoins of the form

$$[a_j, a_i] = w_{ji} \qquad \text{for } i > j \quad i + j = l$$

We regard these facts when we introduce new generators and compute the tails.

Now it should be clear that enforcing the consistency simply means enforcing the consistency relations (C2) for the triples $(a_i, a_j, a_k) \in \mathcal{G} \times \mathcal{G} \times \mathcal{G}$

where $1 \leq i < j < k \leq n$ $\omega(a_i) + \omega(a_j) + \omega(a_k) = l$ and $\omega(a_i) = 1$. And the relations of type (C1) are checked for only pairs $(a_i, a_j) \in \mathcal{G} \times \mathcal{G}$ where $\omega(a_i)+\omega(a_j) = l$. Note that l still denotes the nilpotency class of the current factor.

We are allowed to deal even more simplification, namely we need to enforce only those relations of $\mathcal{R}$ which have weight l.

The natural question arises. How can one recognise a graded Lie ring by looking at its finite presentation? We only mention here the well known fact that a Lie ring defined by homogeneous relations is always graded.

5 Performance

The implementation for the LieNQ algorithm has been written in C programming language. The currently available version is the Version 1.3, which provides time measure, computations for graded and non-graded cases separately.

The program takes a finite presentation $(\mathcal{X}, \mathcal{R})$ for a Lie ring L and the nilpotency class of required factor as its input. It outputs the TR, PR relations and the images of the elements of $\mathcal{X}$ under the epimorphism. The nilpotency class can be omitted, in this case the program runs until the lower central series stabilizes or the algorithm goes beyond the capacity of the computer.

Throughout the computation the data of the nilpotent presentation is stored in normal word form. At previous stages they were stored as coefficient vectors, but it turned out to be more efficient in an array consisting of structures, where the first element of a structure contains the number of a generator and the second one its coefficient (the idea is due to [7]). For instance, the Lie ring element

$$0 \cdot a_1 + 0 \cdot a_2 + 2 \cdot a_3 - 4 \cdot a_4 + 0 \cdot a_5$$

is stored as

$$((3, 2), (4, -4), (0, 0))$$

The first component of the last pair indicates an `EOW` (end of word) sign, while the second one, in fact, is arbitrary.

The information of the nilpotent presentation is stored in several arrays as follows:

`Coefficients[i]`	o_i if $i \in I$, 0 otherwise,
`Power[i]`	w_i, right hand side of the TR relation $o_i \cdot a_i$,
`Product[j][i]`	w_{ji}, right hand side of the PR relation $[a_j, a_i]$,
`Epimorphism[i]`	$x_i\phi$, the ith epimorphic image,
`Weight[i]`	$\omega(a_i)$, the weight of a_i,
`Dimension[i]`	number of generators of weight i,
`Definition[i]`	the definition of a_i.

Note that `Definition[i]` is a structure. It either contains the number of the two generators if the definition of a_i is like $a_i = [a_k, a_l]$, or contains the number of the generator from $\mathcal{X}$ in the first component and 0 in the second if a_i is defined as an epimorphic image.

The defining relations are stored as expression trees. See [6] for its description which emphasizes its advantages in details.

During the computation graded and non-graded cases are distinguished according to whether the user switched on the `-G` option or not. There are different tails routines, that is `Tails` and `GradedTails` and consistency routines like `Consistency` and `GradedConsistency` built in the code. Of course, introducing new generators is also a different task.

It is, however, important to introduce the new generators in a right order. Lemma 3.7 makes possible to express all tails in terms of generators a_k of weight l where $a_k = [a_j, a_i]$ for some a_j of weight $l-1$ and a_i of weight 1. In order to use this fact in practice, we introduce those generators the last and when we compute the row Hermite normal form for the matrix they will be, in fact, the only surviving generators and all the others disappear. In this way we ensure that the new generators indeed have definitions, so the restrictions we kept are reasonable.

The routine for matrix computation and the basics of the parser program are due to W. Nickel. The first mentioned one uses the GNU MP package to deal with long integers.

6 Some Sample Computation

For showing some examples, several coputations have been done with the positive components of simple Lie algebras of finite dimension (L^+ denotes it for an arbitrary L) over the integers. By using them it was possible to check whether the free ranks of the computed nilpotent presentation correspond to the known dimensions of those algebras.

Those algebras are finitely presented, the finite presentation for them was computed by using a GAP program written by W. Nickel [5].

The computations showed that G_2^+/P[1] is nilpotent and torsion-free as a Lie ring where

$$P = \langle\ [e_2, e_1, e_2, e_2, e_1, e_2]\ \rangle$$

is an ideal generated by an element of order 2. P was simply computed by adding more relators to the finite presentation, as far as it turned out to be nilpotent. Similar result has been obtained for F_4^+/P where

$$\begin{aligned} P = \langle \quad & [e_3, e_2, e_1, e_2], [e_4, e_3, e_2, e_3, e_2], [e_4, e_3, e_2, e_3, e_3], \\ & [e_3, e_2, e_3, e_1, e_2, e_3], [e_4, e_3, e_2, e_3, e_4, e_3], \\ & [e_4, e_3, e_2, e_3, e_1, e_2, e_1], [e_4, e_3, e_2, e_3, e_4, e_1, e_2, e_3, e_4], \\ & [e_4, e_3, e_2, e_3, e_1, e_2, e_3, e_4, e_3, e_2, e_3]\ \rangle \end{aligned}$$

is an ideal consisting only of elements of finite order. I found a similar result in E_6^+ where the torsion ideal is

$$\begin{aligned} P = \langle \quad & [e_4, e_3, e_1, e_3], [e_4, e_3, e_2, e_4], [e_5, e_4, e_2, e_4], [e_5, e_4, e_3, e_4], \\ & [e_6, e_5, e_4, e_5], [e_5, e_4, e_3, e_2, e_4, e_2], [e_5, e_4, e_3, e_2, e_4, e_3], \\ & [e_5, e_4, e_3, e_2, e_4, e_5], [e_6, e_5, e_4, e_2, e_3, e_4, e_2], \\ & [e_6, e_5, e_4, e_2, e_3, e_4, e_3]\ \rangle. \end{aligned}$$

Further computations suggested the conjecture that A_n^+/P_n is nilpotent and torsion-free for $n \geq 3$ where

$$P_n = \langle\, [e_k, e_{k-1}, e_{k-2}, e_{k-1}] \,|\, n \geq k \geq 3\rangle.$$

The computations show that P_n is an ideal in A_n^+ consisting only of elements of order 2. Since the above mentioned elements are linearly independent and of weight 4, they form a minimal generating set for P_n. The examples A_3^+, A_4^+, A_5^+, A_6^+, A_7^+, A_{10}^+ support this claim.

The calculations in algebras of type B_n^+ showed that B_n^+/P_n is nilpotent and torsion-free where

$$P_n = \langle\, [e_k, e_{k-1}, e_{k-2}, e_{k-1}], [e_3, e_2, e_1, e_1, e_2, e_1] \ |\ n \leq k \leq 3\,\rangle$$

is a torsion ideal. The calculations were done for B_3^+, B_4^+ and B_5^+. B_2^+ was found to be nilpotent and torsion-free.

[1] We use the standard notation. See [4] for details.

References

[1] F. Celler, M.F. Newman, W. Nickel, A. Niemayer. An algorithm for computing quotients of prime-power order for finitely presented groups and its implementation in GAP. Research Report SMS-127-93, School of Mathematical Sciences, Australian National University, 1993.

[2] M.F. Newman G. Havas. Applications of computers to questions like those of Burnside. In *Burnside Groups*, volume 806 of *Lecture Notes in Math.*, pages 211–230. Bielefeld, Springer-Verlag, 1977.

[3] G. Havas, M.F. Newman, M.R. Vaughan-Lee. A nilpotent quotient algorithm for graded Lie rings. *J. Symbolic Comput.*, 9:653–664, 1990.

[4] N. Jacobson. *Lie Algebras*. Interscience Publishers, New York and London, 1962.

[5] M. Schönert et al. GAP—Groups, Algorithms and Programming. Lehrstuhl D für Mathematik, RWTH Aachen, 1994.

[6] W. Nickel. Computing nilpotent quotients in finitely presented groups. In *Geometric and Computational Prospectives of Finite Groups*, DIMACS series, 1995. To appear.

[7] W. Nickel. personal communication, 1995.

[8] D.J.S. Robinson. *A Course in the Theory of Groups*, volume 80 of *Graduate Texts in Mathematics*. Springer-Verlag New York, Heidelberg, Berlin, 1982.

[9] Cs. Schneider. Computations in finitely presented Lie rings. Master's thesis, University of Arts and Sciences Debrecen, Hungary, 1996. In preparation.

[10] C.C. Sims. *Computations in Finitely Presented Groups*. Cambridge University Press, 1994.