J. Symbolic Computation (1999) **27**, 387–404
Article No. jsco.1998.0258
Available online at http://www.idealibrary.com on IDEAL

Construction of Finite Groups

HANS ULRICH BESCHE† AND BETTINA EICK‡

†*Lehrstuhl D für Mathematik, RWTH Aachen, 52056 Aachen, Germany*
‡*Mathematisches Institut, Universität Würzburg, 97074 Würzburg, Germany*

We introduce three practical algorithms to construct certain finite groups up to isomorphism. The first one can be used to construct all soluble groups of a given order. This method can be restricted to compute the soluble groups with certain properties such as nilpotent, non-nilpotent or supersoluble groups. The second algorithm can be used to determine the groups of order $p^n \cdot q$ with a normal Sylow subgroup for distinct primes p and q. The third method is a general method to construct finite groups which we use to compute insoluble groups.

1. Introduction

When attempting to determine up to isomorphism the groups of a given order it is often known how to compute a list of groups of this order which contains each isomorphism type at least once. The central problem is to reduce to isomorphism type representatives. In Magnus (1937) one can find the following quote on this problem: *"Die Hauptschwierigkeit besteht dabei nicht in einer Konstruktion aller Gruppen eines bestimmten Typs, sondern in der Angabe eines vollständigen Systems nicht isomorpher Gruppen aus den konstruierten Gruppen".*

For certain classes of groups there exist practical methods to list such groups and to reduce to isomorphism classes. For example, Newman and O'Brien introduced an algorithm to determine up to isomorphism the groups of prime-power order, see O'Brien (1990). For soluble groups it is well known how to list such groups. But the reduction to isomorphism classes has been a comparatively difficult problem. In particular, there was no method to test isomorphism of finite soluble groups known which was sufficiently effective for this type of application.

Here we introduce practical algorithms to construct finite groups up to isomorphism. There are three different methods, as we developed different ways to solve the isomorphism problem for different classes of groups.

Our first method, the *Frattini extension method*, is an algorithm to determine up to isomorphism all finite soluble groups of a given order. Our approach is to compute a list of soluble groups which contains few isomorphic redundancies and then to reduce this list to isomorphism types. We introduce a new method to solve the isomorphism problem for finite soluble groups for this purpose. Moreover, the Frattini extension method can be restricted to compute soluble groups with certain properties only (see Section 4).

0747–7171/99/040387 + 18 $30.00/0

The second algorithm, the *cyclic split extension method*, is a method to construct the groups of order $p^n \cdot q$ for distinct primes p and q with a normal Sylow subgroup. This method is more practical on this class of groups than the Frattini extension method. In particular, the isomorphism problem is reduced to computations within automorphism groups of p-groups and explicit isomorphism testing is avoided (see Section 5).

The third algorithm, the *upwards extension method*, is a general approach to construct finite groups. It is less practical than either of the above two methods and we will only use it to construct insoluble groups. As the number of insoluble groups of a given order is much less than the number of soluble groups, this algorithm has been sufficient for our applications (see Section 6).

We used our algorithms to determine up to isomorphism the groups of order at most 1000 except 512 and 768. For details on implementations and results the reader is referred to Besche and Eick (1998).

2. Motivation

The determination of all groups of a given order up to isomorphism is a very old question in group theory. It was introduced by Cayley who constructed the groups of order 4 and 6 in 1854, see Cayley (1854).

A large number of publications have since studied this problem. For example, Hall and Senior (1964) determined the groups of order 2^n for $n \leq 6$. Neubüser (1967) listed all groups of order at most 100 except for 64 and 96. The groups of order 96 were added by Laue (1982).

Moreover, for certain factorizations of orders the corresponding groups have been classified, e.g. Hölder (1893) determined the groups of order pq^2 and pqr, and James (1980) determined the groups of order p^n for odd primes p and $n \leq 6$.

More recently, algorithms have been used to determine certain groups. For example, O'Brien (1991) determined the 2-groups of order at most 2^8 and the 3-groups of order at most 3^6. Moreover, Betten (1996) developed a method to construct finite soluble groups and used this to construct the soluble groups of order at most 242 at the same time as we worked on our project.

It was probably Hölder who asked for the construction of all groups of order at most 200 as "Preisaufgabe der Fürstlich Jablonowskischen Gesellschaft" of the year 1938, see Crelle (1936). As far as we know, of the groups having order at most 200 those of order 192 had not been enumerated when we started our work on this subject.

3. Descriptions of Finite Soluble Groups

A large part of this paper will be concerned with computations with finite soluble groups and we will need different ways to represent a finite soluble group suitably for our constructions. On the one hand we want to compute with these groups and thus we need a representation which is useful for this purpose. We will use (special) polycyclic generating sequences and their associated presentations (see Sections 3.1 and 3.2). On the other hand, we want to store a large number of finite soluble groups. Thus we need a very space efficient way to represent finite soluble groups. We introduce a representation of a finite soluble group by an integer in Section 3.3.

3.1. POLYCYCLIC GENERATING SEQUENCES

Let G be a finite soluble group and let $G = C_1 \rhd C_2 \rhd \cdots \rhd C_n \rhd C_{n+1} = \{1\}$ be a composition series of G. Then for $1 \leq i \leq n$ the order of the factor C_i/C_{i+1} is a prime which we denote by p_i. If we choose an element $a_i \in C_i \setminus C_{i+1}$ for $1 \leq i \leq n$, then the resulting sequence of elements $(a_1, \ldots, a_n)$ is called a *polycyclic generating sequence*. One can easily see that the subsequence $(a_i, \ldots, a_n)$ forms a polycyclic generating sequence of the group C_i for $1 \leq i \leq n$. Furthermore, one can write G as the set $\{a_1^{e_1} \cdots a_n^{e_n} \mid 0 \leq e_i < p_i \text{ for } 1 \leq i \leq n\}$. A polycyclic generating sequence determines a finite presentation of G with relations of the following form.

$$[a_j, a_i] = a_{i+1}^{e_{i,j,i+1}} \cdots a_n^{e_{i,j,n}} \text{ for } 1 \leq i < j \leq n$$
$$a_i^{p_i} = a_{i+1}^{e_{i,i,i+1}} \cdots a_n^{e_{i,i,n}} \text{ for } 1 \leq i \leq n,$$
$$\text{where } 0 \leq e_{i,j,k} < p_k.$$

This type of presentation is called *power commutator presentation*. The words on the right-hand sides of the relations are denoted by $w_{i,j}$ for $1 \leq i \leq j \leq n$. These presentations can be used to compute with finite soluble groups effectively. See Laue *et al.* (1984) for further information.

3.2. SPECIAL POLYCYCLIC GENERATING SEQUENCES

A *special polycyclic generating sequence* is a polycyclic generating sequence having some additional properties. We will only give a brief outline of the main ideas of a special polycyclic generating sequence. We refer to Cannon and Leedham-Green (1998) for an algorithm to compute a special polycyclic generating sequence from an arbitrary one and some applications of special polycyclic generating sequences. In Eick (1997) one can find a more detailed overview of the properties of a special polycyclic generating sequence.

First, for a special polycyclic generating sequence we choose a composition series of G such that it refines a certain unique, characteristic series of G with elementary abelian factors which is called an LG-series.

Secondly, the choice of the elements $a_i \in C_i \setminus C_{i+1}$ is restricted for a special polycyclic generating sequence. For example, the sequence of elements a_i with $p_i \in \pi$ for a set of primes π must form a polycyclic generating sequence of a Hall π-subgroup of G. In particular, this implies that each element a_i is of p_i-power order.

3.3. CODING POWER COMMUTATOR PRESENTATIONS BY INTEGERS

Suppose a power commutator presentation of G is given in the above form. Then the aim is to store sufficient information about the presentation in one integer to be able to reconstruct the presentation from it. Thus we have to store the sequence of primes $p_1, \ldots, p_n$ and the right-hand sides $w_{i,j}$ of the power commutator presentation.

The resulting integer consists of three parts. The first is a code for the primes $p_1, \ldots, p_n$, the second tells which words $w_{i,j}$ are non-trivial and the third contains the code for the non-trivial words $w_{i,j}$. The second part is introduced to reduce the length of the integer.

Each part will be written as integer to a certain base. For the first part, the base is $m := \max\{p_i \mid 1 \leq i \leq n\} - 1$, for the second part it is $b := 2$ and for the third part we choose

$o := |G|$. To code the primes we compute $t_1 := (p_1-2)+(p_2-2)\cdot m+\cdots+(p_n-2)\cdot m^{n-1}$. Then for the second part we calculate $t_2 := \sum_{i=1}^{n}\sum_{j=i}^{n} \overline{w_{i,j}} b^{n(i-1)+j}$, where $\overline{w_{i,j}}$ is equal to 1, whenever the word $w_{i,j}$ is non-trivial, and $\overline{w_{i,j}}$ is equal to 0 otherwise. To code the non-trivial words in the presentation we sum $t_3 := \sum_{w_{i,j}|\overline{w_{i,j}}=1} \widetilde{w_{i,j}} o^l$, where l counts the summands and $\widetilde{w_{i,j}} + 1$ is equal to the position of $w_{i,j}$ in a canonically ordered list of elements of G. Here we assume that the identity is the first element of G yielding $\tilde{1} = 0$. The final code is then derived as $t_1 + m^n \cdot (t_2 + b^{n(n+1)/2+1} \cdot t_3)$.

Hence it is straighforward to compute a code corresponding to a power commutator presentation. For the decoding we need the code and the order of the group. Then it is a simple calcalutation to obtain the power commutator presentation corresponding to the code.

4. The Frattini Extension Method

Gaschütz (1953) suggested an approach to construct finite groups. We modified this to obtain a practical algorithm to construct finite soluble groups up to isomorphism. The Frattini subgroup $\phi(G)$ of a finite soluble group G, i.e. the intersection of all maximal subgroups of G, plays a fundamental role in this approach. As the Frattini subgroup is invariant under isomorphisms, we can split the construction process in two steps.

In the first step we compute up to isomorphism a list of candidates for the Frattini factors $G/\phi(G)$ of the soluble groups G of order o. Gaschütz determined a class of groups to be suitable as candidates and described a method to list them up to isomorphism, see Section 4.1.

In the second step we consider each computed candidate F in turn and compute up to isomorphism the extensions G of F of order o with $G/\phi(G) \cong F$. Extensions G of this type are called *Frattini extensions* and we outline their construction in Section 4.2.

One advantage of this two-step approach is that we obtain the Frattini factors of the desired groups without explicit isomorphism testing. However, in the second step of this algorithm we may need to filter out isomorphic copies from a computed list of groups. In Section 4.3 we introduce a practical method to reduce a list of finite soluble groups to isomorphism type representatives.

We give a summary of the second step of our algorithm in Section 4.4 and in Section 4.6 we describe an example for an application of the Frattini extension method.

Another advantage of this overall approach is that we can restrict the method to extend certain candidates for Frattini factors only. This is useful, because a number of properties of finite soluble groups are inherited from Frattini factors and hence we can use this to determine groups with certain properties only, see Section 4.5.

4.1. Construction of candidates for Frattini factors

First we consider the Frattini factors of the finite groups of order o in general and we restrict to soluble groups later.

Lemma 4.1. (Gaschütz, 1953) *Let G be a finite group. Then $\phi(G/\phi(G)) = \phi(G)/\phi(G)$.*

Thus Frattini factors of finite groups have trivial Frattini subgroup; that is, they are *Frattini free*. In the following lemma we recall a well-known restriction on the order of the Frattini factors of the finite groups of order o.

LEMMA 4.2. *Let F be the Frattini factor of a group of order o. Then $|F|$ divides o and $|F|$ is divisible by each prime divisor of o.*

In the following theorem a construction of Frattini free groups is described. Recall that the socle of a finite group is a direct product of finite simple groups. We say that a group K acts *semisimply* on an abelian group A, if the intersection of all maximal K-normal subgroups of A is trivial.

THEOREM 4.3. (GASCHÜTZ, 1953) *Let S be a direct product of finite simple groups. Write $S = A \times B$ where A is abelian and B is a direct product of non-abelian simple groups. Consider the subgroups of* $\mathrm{Aut}(S)$ *which contain $Inn(S)$ and act semisimply on A and let Γ be a set of conjugacy class representatives of such subgroups. Then the set $\{A \rtimes K \mid K \in \Gamma\}$ is a full set of isomorphism type representatives of Frattini free groups with socle isomorphic to S.*

This yields a method to construct all Frattini free groups with given socle S. As the finite simple groups are classified, we can deduce a method to construct the Frattini free groups of order f.

As candidates for the Frattini factors of the soluble groups of order o we choose the soluble Frattini free groups of order dividing o and divisible by each prime divisor of o. In practice, this list of groups is usually very close to the desired Frattini factors.

By Theorem 4.3 we can determine up to isomorphism the soluble Frattini free groups of order f, if we consider all direct products A of groups of prime order with $a := |A|$ dividing f and compute up to conjugacy the semisimple soluble subgroups K of $\mathrm{Aut}(A)$ with order $k = f/a$.

4.1.1. SEMISIMPLE SOLUBLE GROUPS

Let $p_1, \ldots p_n$ be distinct primes and consider $A \cong C_{p_1}^{e_1} \times \cdots \times C_{p_n}^{e_n}$.

We will use semidirect products to compute up to conjugacy the semisimple soluble subgroups of order k in $\mathrm{Aut}(A) \cong GL(e_1, p_1) \times \cdots \times GL(e_n, p_n)$. Semidirect products have been investigated by Remak (1930), and in Eick (1996) a method to compute them up to conjugacy has been introduced.

If K is a semisimple subgroup of $\mathrm{Aut}(A)$, then K induces a semisimple action on the Sylow subgroups $C_{p_i}^{e_i}$ of A. Thus K is a subdirect product of semisimple subgroups K_i of $\mathrm{Aut}(C_{p_i}^{e_i}) = GL(e_i, p_i)$. Moreover, two semisimple subgroups of $\mathrm{Aut}(A)$ are conjugate, if and only if their subdirect factors in $GL(e_i, p_i)$ are conjugate for all i.

Now we want to determine up to conjugacy the semisimple soluble subgroups of $GL(e, p)$. Here we consider all partitions $d_1, \ldots, d_r$ of e and construct semisimple groups as subdirect products of irreducible soluble subgroups of $GL(d_i, p)$. In this case two subdirect products are conjugate, if their subdirect factors are conjugate.

Thus it remains to determine up to conjugacy the irreducible soluble subgroups of $GL(d, p)$ of order dividing k. Short (1992) has developed an algorithm to compute up to conjugacy all soluble irreducible subgroups of a general linear group. However, there are some important special cases in which these groups can be obtained more easily.

- $d = 1$: This is the trivial case, as $GL(d, p)$ is cyclic and every subgroup of $GL(d, p)$ is soluble and acts irreducibly.
- $p^d < 256$: Short used his method to obtain a catalog of the soluble irreducible subgroups of $GL(d, p)$ for $p^d < 256$ which we can use here.
- "small" k: As $k < o$ we may assume by induction that the soluble groups of order dividing k are known. Then for each such group L we determine up to equivalence the irreducible $\mathbb{F}_p L$-representations of dimension d, see Plesken (1987) for a practical method. From this list of representations we discard the non-faithful ones. Now it remains to reduce the list of modules obtained to conjugacy class representatives in $GL(d, p)$. This approach is practical, if there are only few groups L to process. The following lemma shows that we may reduce the computation to groups L with $O_p(L) = \{1\}$.

LEMMA 4.4. (GASCHÜTZ, 1954) *Let L be a finite group. Then there exists an irreducible faithful $\mathbb{F}_p L$-module if and only if $O_p(L) = \{1\}$ and there exists a conjugacy class of elements of L which generates the socle of L.*

4.2. CONSTRUCTION OF FRATTINI EXTENSIONS

Now we assume that we have a soluble Frattini free group F of order f dividing o and divisible by each prime divisor of o. We want to construct up to isomorphism all groups G of order o with $G/\phi(G) \cong F$. To begin, we need a more detailed definition of Frattini extensions.

DEFINITION 4.5. Let G, H and M be finite groups.

- G is an *extension* of H by M, if there exists $N \trianglelefteq G$ with $N \cong M$ and $G/N \cong H$.
- G is a *Frattini extension* of H by M, if G is an extension of H by M and $G/\phi(G) \cong H/\phi(H)$.
- G is a *minimal Frattini extension* of H by M, if G is a Frattini extension of H by M and there exists a minimal normal subgroup $N \trianglelefteq G$ with $N \cong M$ and $G/N \cong H$.

Suppose G is a Frattini extension of H and H is a Frattini extension of F. Then G is an extension of F as well and $G/\phi(G) \cong H/\phi(H) \cong F$. Thus G is a Frattini extension of F. Therefore our overall approach to construct Frattini extensions is to iterate the construction of minimal Frattini extensions until we obtain groups of order o.

LEMMA 4.6. *Let G be an extension of H by M. Then G is a minimal Frattini extension of H, if and only if there exists a minimal, non-complemented normal subgroup N of G with $N \cong M$ and $G/N \cong H$.*

PROOF. Let N be a minimal normal subgroup of G. If N is not complemented, then $N \leq \phi(G)$ and thus $\phi(H) \cong \phi(G/N) \cong \phi(G)/N$. Hence $H/\phi(H) \cong G/N/\phi(G/N) \cong G/\phi(G)$ and G is a minimal Frattini extension of H.

On the other hand, if N is a complemented minimal normal subgroup of G, then $N \cap \phi(G) = \{1\}$ and $\phi(G)N/N \leq \phi(G/N)$. Thus $H/\phi(H) \cong G/N/\phi(G/N)$ is a proper factor group of $G/\phi(G)$. Hence G is not a Frattini extension of H.□

Thus the minimal Frattini extensions of H are exactly the non-split extensions of H by an irreducible H-module M.

To construct minimal Frattini extensions of H we have to consider all suitable irreducible H-modules M. Clearly, M can be viewed as an irreducible $\mathbb{F}_pH$-module for some prime p. Moreover, we may assume that we construct Frattini extensions G of F along a central series of the nilpotent group $\phi(G)$. Thus it is enough to consider up to equivalence all irreducible $\mathbb{F}_pH$-modules which are centralized by $\phi(H)$; that is, irreducible $\mathbb{F}_pF$-modules. A practical method to compute such modules up to equivalence is described in Plesken (1987).

Then we have to determine non-split extensions of H by M. For this purpose we compute the second cohomomology group $H^2(H, M)$, see Wegner (1992) for a practical method. The non-trivial elements of $H^2(H, M)$ correspond one-to-one to the equivalence classes of non-split extensions of H by M. As equivalent extensions are isomorphic, the non-trivial elements of $H^2(H, M)$ lead to a set of extensions of H by M which contains each isomorphism type of minimal Frattini extension at least once.

In Section 4.2.1 we introduce a method to compute certain subsets of $H^2(H, M)$ which lead to isomorphic extensions. Using this algorithm the number of computed extensions can be reduced substantially.

By iteration of minimal Frattini extension steps, we obtain a list of Frattini extensions G of F of order o which contains each isomorphism type at least once. However, the list can contain isomorphic groups. Thus we have to reduce the computed list of extensions to isomorphism class representatives. For this purpose we use the method described in Section 4.3. Clearly, we reduce the list of extensions after each minimal Frattini extension step to avoid the extension of isomorphic groups.

4.2.1. CONSTRUCTION OF EXTENSIONS UP TO STRONG ISOMORPHISM

Let H be a finite group and M an $\mathbb{F}_pH$-module and consider two extensions L_1 and L_2 of H by M via two cocycles ψ_1 and ψ_2. Let $M_i \trianglelefteq L_i$ be the subgroups corresponding to M for $i = 1, 2$. Then L_1 and L_2 are *strongly isomorphic*, if there exists an isomorphism $\iota : L_1 \to L_2$ with $M_1^\iota = M_2$. If $\psi_1 \equiv \psi_2 \bmod B^2(H, M)$, then L_1 and L_2 are strongly isomorphic. But this condition does not characterize strong isomorphism.

In this section we introduce a method to find a subset of $Z^2(H, M)$ which leads to representatives of strong isomorphism classes of extensions of H by M. Let $A := \mathrm{Aut}(H) \times \mathrm{Aut}(M)$ where M is considered as an elementary abelian p-group and let $H \to \mathrm{Aut}(M) : h \mapsto \overline{h}$ be the operation homomorphism to the operation of H on M. We say that $(\alpha, \nu) \in A$ is a *compatible pair*, if $\overline{h^\alpha} = (\overline{h})^\nu$ holds for all $h \in H$. Let C be the subgroup of all compatible pairs in A.

We want to introduce an action of C on $H^2(H, M)$. Suppose that $(\alpha, \nu) \in C$ and let $\psi \in Z^2(H, M)$ be a cocycle. Then we define $\psi^{(\alpha,\nu)} : H \times H \to M : (x, y) \mapsto ((x^\alpha, y^\alpha)^\psi)^{\nu^{-1}}$. It is straightforward, but technical, to prove that this defines an action of C on $Z^2(H, M)$ and that $B^2(H, M)$ is setwise invariant under this action, see Robinson (1981). Thus we obtain an induced action of C on $H^2(H, M)$ which is linear, because ν is linear. The next theorem shows that the orbits of this action on $H^2(H, M)$ correspond one-to-one to the strong isomorphism classes of extensions of H with M.

Note that in Robinson (1981) similar methods to determine the automorphism group of the extension of H by M via a cocycle ψ have been used. Furthermore in Laue (1982) the classification of strong isomorphism classes of extensions of H has been con-

sidered. While we use the intermediate step of computing the subgroup of compatible pairs and consider one module M at a time only, Laue determined these classes in a single step approach which involves the difficulty that the module considered is not fixed.

THEOREM 4.7. *Let L_i be an extension of a finite group H by an $\mathbb{F}_pH$-module M via the cocycle ψ_i for $i = 1, 2$. Then L_1 is strongly isomorphic to L_2, if and only if there exists an element $(\alpha, \nu) \in C$ such that $\psi_1^{(\alpha,\nu)} \equiv \psi_2 \bmod B^2(H, M)$.*

PROOF. $\Rightarrow$ Let $\iota : L_1 \to L_2$ be a strong isomorphism and denote by $\tau_i : H \to L_i$ for $i = 1, 2$ the canonical transversals to the extensions. Then ι induces automorphisms $\iota\ |_{L_1/M_1} =: \alpha^{-1} \in \mathrm{Aut}(H)$ and $\iota\ |_{M_1} =: \nu^{-1} \in \mathrm{Aut}(M)$. Furthermore, there exists a function $\eta : H \to M : h \mapsto m_h$ such that

$$((h^\alpha)^{\tau_1})^\iota = h^{\tau_2} \cdot m_h.$$

Let $\gamma : H \times H \to M : (g, h) \mapsto m_{gh} \cdot m_g^{-\overline{h}} \cdot m_h^{-1}$ be the coboundary corresponding to the function η. We have to show that $\psi_1^{(\alpha,\nu)} \cdot \gamma = \psi_2$. Let $g, h \in H$. Then

$$\begin{aligned}((g^\alpha)^{\tau_1} \cdot (h^\alpha)^{\tau_1})^\iota &= ((g^\alpha)^{\tau_1})^\iota \cdot ((h^\alpha)^{\tau_1})^\iota \\ &= g^{\tau_2} \cdot m_g \cdot h^{\tau_2} \cdot m_h \\ &= g^{\tau_2} \cdot h^{\tau_2} \cdot m_g^{\overline{h}} \cdot m_h.\end{aligned}$$

On the other hand, we have

$$\begin{aligned}((g^\alpha)^{\tau_1} \cdot (h^\alpha)^{\tau_1})^\iota &= ((g^\alpha \cdot h^\alpha)^{\tau_1} \cdot (g^\alpha, h^\alpha)^{\psi_1})^\iota \\ &= ((g^\alpha \cdot h^\alpha)^{\tau_1})^\iota \cdot ((g^\alpha, h^\alpha)^{\psi_1})^\iota \\ &= (gh)^{\tau_2} \cdot m_{gh} \cdot ((g^\alpha, h^\alpha)^{\psi_1})^{\nu^{-1}}.\end{aligned}$$

Combining the results of the two equations we obtain

$$\begin{aligned}(g, h)^{\psi_2} &= ((gh)^{\tau_2})^{-1} \cdot g^{\tau_2} \cdot h^{\tau_2} \\ &= m_{gh} \cdot ((g^\alpha, h^\alpha)^{\psi_1})^{\nu^{-1}} \cdot m_g^{-\overline{h}} \cdot m_h^{-1} \\ &= (g, h)^{\psi_1^{(\alpha,\nu)}} \cdot m_{gh} \cdot m_g^{-\overline{h}} \cdot m_h^{-1} \\ &= (g, h)^{\psi_1^{(\alpha,\nu)}} \cdot (g, h)^\gamma\end{aligned}$$

which yields the first part of the equivalence.

$\Leftarrow$ Let $(\alpha, \nu) \in C$ and $\gamma \in B^2(H, M)$ such that $\psi_1^{(\alpha,\nu)} \cdot \gamma = \psi_2$. By definition there exists a function $\eta : H \to M : h \mapsto m_h$ corresponding to the coboundary γ. Define a mapping $\iota : L_1 \to L_2 : h^{\tau_1} m \mapsto (h^{\alpha^{-1}})^{\tau_2} \cdot m_{h^{\alpha^{-1}}} \cdot m^{\nu^{-1}}$. Now we have to show that ι is a strong isomorphism from L_1 to L_2.

First, one has to show that ι is in fact an homomorphism. This is done by computations similar to above. Then it is easy to see that ι is injective and thus, because $|L_1| = |L_2|$, it has to be surjective as well. Finally, one can observe from the definition of ι that $M_1^\iota = M_2$ and thus ι is a strong isomorphism from L_1 to L_2.□

Thus, by Theorem 4.7, we have to compute orbit representatives under action of C on $H^2(H, M)$ to obtain strong isomorphism classes of extensions. For a method to compute

C the reader is referred to Smith (1995). Note that the action of C on $H^2(H, M)$ is linear. Therefore we can compute a matrix representation of the action of C on $H^2(H, M)$ and then it remains to compute orbits of vectors under action of a matrix group.

4.3. solving the isomorphism problem

In this section we introduce a method to reduce a given list of finite soluble groups to isomorphism class representatives. Our method proceeds in three steps.

In the first step we compute invariants of the given groups and use them to split the list of groups up in sublists. As the given list of groups may be large, we need invariants which are efficient to compute, see Section 4.3.1.

In the second step we consider each sublist separately and search for isomorphisms among the groups in a sublist using a probabilistic method. We describe a method to detect isomorphisms in a list of finite soluble groups at random in Section 4.3.2. The output of this algorithm is a sublist of the input list in which only isomorphic copies have been removed.

Finally, in step three, we have to verify that the lists of groups obtained do not contain isomorphic copies (or remove copies). The lists of groups should be small now; in fact, many of them should have a length of one. For the lists of length more than one we can continue to compute invariants of groups. As we are dealing with small lists of groups now we can afford to use invariants which are less efficient to compute here. Another possibility is to use a general, deterministic isomorphism test to verify the result, see Hulpke (1996).

4.3.1. invariants of groups

To distinguish groups by invariants it is important to choose a suitable set of invariants for the given groups; they should be fast to compute and nevertheless helpful. Therefore, the choice of invariants depends on the number of groups in the list as well as on the structure of the groups and their representation.

In our applications G is a finite soluble group given by a power-commutator presentation. Then, for example, the order of G or the isomorphism types of the factors in the derived series of G will be very efficient to compute, but they will not help very much in distinguishing groups.

We have chosen a set of invariants which is based on the set of conjugacy classes $\mathcal{C}$ of G. We compute a partition of $\mathcal{C}$ into subsets $\mathcal{C}_1, \ldots, \mathcal{C}_l$ which is respected by the automorphism group of G. The aim is to obtain as many subsets as possible, but without explicitly computing the automorphism group.

We initialize a partition of $\mathcal{C}$ by the subsets of $\mathcal{C}$ containing the classes of a certain length and a certain representative order. Then we apply the following operations to split this initial partition into smaller subsets. For every subset $\mathcal{C}_i$ of a partition we derive a three-tuple consisting of the number of classes in $\mathcal{C}_i$, their length and their representative order and we combine these three-tuples to obtain an invariant of the group.

- *Fixed points under power maps:* Let $\mathcal{C}_i$ be a partition containing classes with representative order o and let p be coprime to o with $p < o$. Then we consider the power map $f_p : x \mapsto x^p$ and separate the classes in $\mathcal{C}_i$ which are fixed under f_p from the non-fixed classes.

- *Powers of classes:* Let $\mathcal{C}_i$ be a partition containing classes with representative order o and let p be a prime dividing o. Again we consider the map $f_p : x \mapsto x^p$. If we apply f_p to an element in a class of $\mathcal{C}_i$, then we obtain an element in a class of a different partition $\mathcal{C}_j$, because the order of this element is smaller than o. Thus to each class in $\mathcal{C}_i$ we can assign the partition containing the powers of this class. Now we separate the classes in $\mathcal{C}_i$ according to their power partition.
- *Roots of classes:* We use the same approach as above together with the examination of pth roots instead of pth powers.
- *Mapped words:* We generalize the above approaches by applying functions $f_w : (a, b) \mapsto w(a, b)$ to tuples of partitions $\mathcal{C}_i \times \mathcal{C}_j$ where $w(a, b)$ is a word in a and b.

The first operation is very efficient, while the second and third are slower. The last approach might be too slow to apply it to large lists of groups. Moreover, it may be useful to iterate the last three operations until no further splitting of partitions occurs.

4.3.2. REMOVING ISOMORPHIC COPIES

Our general approach is to loop over a given list of finite soluble groups and compute a random special polycyclic generating sequence and its associated power commutator presentation for each group in turn. Then each computed new presentation will be coded as an integer and stored. If we find the same presentation for two different groups, they must be isomorphic. Hence we can discard one of them.

We loop over the list (possibly several times) until either only one group remains or until the list has remained constant after a number of applications of the procedure described here; that is, until for each of the groups we computed n times a known presentation for some given integer n. Thus n controls the probability to find all isomorphisms among groups in the list. In particular, if $n = 0$, then the algorithm will return the input list of groups. Note that we can never guarantee that the algorithm will find all isomorphisms. An outline of the algorithm is given in Figure 1.

This procedure will perform better when many groups in the list *groups* are isomorphic copies of another group, as in this case only for a few groups the limit n for the number of identical presentations has to be reached. Thus the preceding computation of invariants as outlined in Section 4.3.1 is essential.

It remains to describe a procedure to compute a random special polycyclic generating sequence for a finite soluble group G. We assume that G is given by a power commutator presentation associated to a special polycyclic generating sequence. Such a polycyclic generating sequence goes through the LG-series $G = L_1 \trianglerighteq L_2 \trianglerighteq \cdots \trianglerighteq L_m \trianglerighteq \{1\}$ of G and exhibits a Sylow system $\mathcal{S} = \{S_p \mid p \text{ prime}\}$ of G.

In the first step we want to determine a random polycyclic generating sequence which goes through the LG-series and exhibits a Sylow system of G. As G is soluble, all Sylow systems of G are conjugate. Hence each polycyclic generating sequence exhibiting a Sylow system of G is conjugate to a polycyclic generating sequence exhibiting $\mathcal{S}$. However, conjugation of a polycyclic generating sequence does not change the corresponding power commutator presentation. Hence we may compute a random polycyclic generating sequence which exhibits the given Sylow system $\mathcal{S}$.

Consider a factor group $L_j/L_{j+1} \cong C_p^d$. First we choose a random basis $b_1, \ldots, b_d$ of this elementary abelian group. Each basis element b_i corresponds to a word $w(b_i)$ in the elements of the given special polycyclic generating sequence of G contained in $L_j \setminus L_{j+1}$.

```
RemoveIsomorphicCopies(groups, n)
let l be the length of the list groups
initialize codes as list of length l consisting of empty lists
initialize hit as list of length l consisting of zeros
while Minimum(hit) ≤ n and there is more than one group in groups unmarked do
    for i = 1, 2, ..., l do
        if groups[i] is unmarked
          compute a new special polycyclic generating sequence of groups[i] at random
          compute the associated power commutator presentation and code it as integer c
          if c ∈ codes[i] then
            hit[i] := hit[i] + 1
          elif c ∈ codes[j] for some j ≠ i then
            mark groups[i]
            concatenate codes[i] to codes[j] and reset codes[i] := []
            reset hit[j] := 0 and reset hit[i] := n + 1
          else
            add c to codes[i]
          fi
        fi
    od
od
return the list of unmarked members in groups
```

Figure 1. The algorithm *RemoveIsomorphicCopies*.

Note that $w(b_i)$ is an element of the exhibited Sylow subgroup S_p. Then we compute a random element $f_i \in L_j \cap S_p$. This amounts to the computation of a random exponent vector, as L_{j+1} as well as S_p are exhibited by the special polycyclic generating sequence of G. We assign $a_i := w(b_i) \cdot f_i$. Then the concatenation of the sequences $(a_1, \ldots, a_d)$ for each of the factors L_j/L_{j+1} yields a random polycyclic generating sequence of G which goes through the LG-series and exhibits $\mathcal{S}$.

Thus we constructed a random polycyclic generating sequence of G which is "nearly" special and it remains to compute a unique special polycyclic generating sequence and the associated power commutator presentation from it.

Clearly, it is useful to restrict the choice of the random elements a_i as far as possible. For example, if $L_j \cap S_p$ is small enough to list all its elements, then we allow elements of smallest possible order only.

4.4. A SUMMARY OF THE ALGORITHM

Let F be a soluble Frattini free group and o an integer. Suppose that the order of F divides o and is divisible by each prime divisor of o. In Figure 2 we summarize the determination up to isomorphism of the soluble groups of order o with Frattini factor isomorphic to F.

4.5. RESTRICTION TO CLASSES OF GROUPS

A number of properties of a finite soluble group can be read off its Frattini factor; for example, a finite soluble group is nilpotent, if and only if its Frattini factor is nilpotent. The same holds for supersoluble groups and groups with a normal Sylow p-subgroup. In fact, this is possible for all properties such that the groups with this property form a "saturated formation", see Gaschütz (1963) for details.

```
FrattiniExtensions(F, o)
for each prime p dividing o/|F| do
    compute the list mods[p] of irreducible 𝔽_pF-modules
od
initialize groups as empty list
initialize extend as list containing F
set h := |F|
while h < o do
    let p be the smallest prime dividing o/h
    for H in extend do
        for each module M in mods[p] with |M| dividing o/h do
            consider M as 𝔽_pH-module
            compute H^2(H, M)
            if H^2(H, M) has sufficiently large order then
              compute the compatible pairs C ≤ Aut(H) × Aut(M)
              compute orbits of C on H^2(H, M)
            fi
            for each non-trivial representative of cocycles ψ do
              add the extension G of H by M via ψ to groups
            od
        od
    od
    reset h to the minimal order of groups in groups (or h := o, if groups is empty)
    reset extend := [] and move the groups of order h from groups to extend
    reduce extend to representatives of isomorphism classes
od
return the list extend
```

Figure 2. The algorithm *FrattiniExtensions*.

Hence we construct the soluble groups with or without such a property by restricting to the corresponding Frattini factor candidates. In the next lemma we describe the structure of such Frattini free groups for three important properties.

LEMMA 4.8. *Let F be a soluble Frattini free group with socle A and socle complement K. Then the following equivalences hold.*

(i) F is nilpotent if and only if K is trivial.
(ii) F has a normal Sylow p-subgroup if and only if p does not divide the order of K.
(iii) F is supersoluble if and only if A is the direct sum of irreducible K-modules of dimension 1.

PROOF. Part (i) is trivial.

Consider part (ii). In Gaschütz (1953) it is observed that $\mathrm{Soc}(F) = \mathrm{Fit}(F)$ holds for a soluble Frattini free group F. Thus if F is a soluble Frattini free group with normal Sylow p-subgroup T, then $T \leq \mathrm{Fit}(F) = \mathrm{Soc}(F) = A$ is obtained. Therefore p cannot divide $|K|$. Conversely, if p does not divide $|K|$, then the Sylow p-subgroup of A is also the Sylow p-subgroup of F and must be normal.

It remains to prove part (iii). If A is a direct sum of irreducible K-modules of dimension 1, then K is an iterated subdirect product of abelian groups and thus K is abelian. This yields that there exists a chief series with prime factors for F and thus F is supersoluble. On the other hand, if F is supersoluble, then there exists a chief series of F through A with prime factors. Thus each irreducible K-submodule of A is of dimension 1. As F is Frattini free, we know that A is the direct sum of irreducible K-modules.□

Thus the construction of nilpotent Frattini free groups is trivial. The supersoluble Frattini free groups can be determined as subdirect products of irreducible linear groups of dimension 1. Hence it is trivial to obtain the irreducible subdirect factors for these groups. To obtain the soluble Frattini free groups with a normal Sylow p-subgroup we can restrict the order of K and hence have to consider only certain possible socles A.

4.6. EXAMPLE

We describe the calculation of all groups of order $192 = 2^6 \cdot 3$ without a normal Sylow subgroup.

The first step in the construction is the determination of the candidates for the Frattini factors. As explained in Section 4.1 we use Frattini free groups of order f where $2 \cdot 3 \mid f$ and $f \mid 2^6 \cdot 3$ as candidates for the Frattini factors of all groups of order 192. To restrict the construction to groups without a normal Sylow subgroup, we restrict the computation of Frattini free groups to groups without normal Sylow subgroups. Thus, as shown in Lemma 4.8, we have to list the Frattini free groups of the form $A \rtimes K$ such that $A \cong C_2^e$ for $1 \leq e \leq 5$ and $2 \cdot 3 \mid |K|$.

Thus we have to start with the irreducible soluble subgroups of $GL(e, 2)$ for $1 \leq e \leq 5$ of order dividing $2^{6-e} \cdot 3$. Here $GL(1, 2)$ and $GL(2, 2)$ are the only two groups with these properties.

The next step is to compute all subdirect products up to conjugacy of these two groups up to dimension 5 with suitable order. There are three categories of subdirect products fulfilling this dimension-bound: first the direct product of up to 5 copies of $GL(1, 2)$, secondly the direct product of up to 3 copies of $GL(1, 2)$ and 1 copy of $GL(2, 2)$ and thirdly the direct product of up to one copy of $GL(1, 2)$ with any subdirect product $GL(2, 2) \wedge GL(2, 2)$. The groups in the first category all have order 1 and thus do not fulfil the order requirements. The groups in the second category all have order 6 and thus fulfil the order requirements. It remains to consider the groups in the third category. Up to conjugacy in $GL(4, 2)$ there are three subdirect products of $GL(2, 2)$ with $GL(2, 2)$ of orders 36, 18 and 6. We can only use the one of order 6 here which we denote by $\overline{GL(2, 2)} \leq GL(4, 2)$. We obtain the following list of possible socle complements:

$$K_{1,j} := GL(1, 2)^j \times GL(2, 2) \text{ for } 0 \leq j \leq 3,$$
$$K_{2,j} := GL(1, 2)^j \times \overline{GL(2, 2)} \text{ for } 0 \leq j \leq 1.$$

This yields the following list of candidates for the Frattini factors:

$$F_{1,j} := C_2^{2+j} \rtimes K_{1,j} \text{ for } 0 \leq j \leq 3,$$
$$F_{2,j} := C_2^{4+j} \rtimes K_{2,j} \text{ for } 0 \leq j \leq 1.$$

Now we have to determine Frattini extensions of order 192 of each of these groups. We compute iterated minimal Frattini extensions for this purpose. In Table 1 we list the number of Frattini extensions of order dividing 192 for each of the groups $F_{i,j}$.

Note that each of the Frattini free groups appears as the Frattini factor of a group of order 192.

5. The Cyclic Split Extension Method

Let p and q be distinct primes throughout this section. Here we want to outline a method to construct up to isomorphism all groups G of order $p^n \cdot q$ which have a normal

Table 1.

	$F_{1,0}$	$F_{1,1}$	$F_{1,2}$	$F_{2,0}$	$F_{1,3}$	$F_{2,1}$
24	1					
48	3	1				
96	4	11	1	1		
192	8	48	21	7	1	1

Sylow subgroup, say S. Clearly, S is complemented in G by a subgroup K and K has to be a Sylow subgroup for the other prime. Hence the structure of G is completely defined by the structure of S and K and the operation of K on S.

The idea is to construct operations of K on S where S or K runs over all groups of order p^n and the other group is isomorphic to C_q. We assume that we have a list of isomorphism types of representatives of the groups of order p^n. If such a list is not known, it may be determined using the p-group generation method, see O'Brien (1990). Thus we have to find a set of operations of K on S which leads to a list of isomorphism type representatives of such groups.

THEOREM 5.1. (TAUNT, 1955) *Let S and K be finite soluble groups with $(|S|, |K|) = 1$. Furthermore let $\psi_i : K \to \mathrm{Aut}(S)$ for $i = 1, 2$ be two homomorphisms from K into $\mathrm{Aut}(S)$. Define $G_i := S \rtimes_{\psi_i} K$ for $i = 1, 2$.*

Then $G_1 \cong G_2$, if and only if there exist automorphisms $\alpha \in \mathrm{Aut}(S)$ and $\beta \in \mathrm{Aut}(K)$ such that $(k^\beta)^{\psi_2} = (k^{\psi_1})^\alpha$ for all $k \in K$.

5.1. THE CASE $S \cong C_q$ AND K OF ORDER p^n

Here $\mathrm{Aut}(S) \cong C_{q-1}$. In particular, $\mathrm{Aut}(S)$ is abelian. Thus, according to Theorem 5.1, two homomorphisms ψ_1 and ψ_2 lead to isomorphic split extensions, if and only if there exists an automorphism $\beta \in \mathrm{Aut}(K)$ with $k^{\psi_1} = (k^\beta)^{\psi_2}$ for all $k \in K$; that is, $\psi_1 = \beta \cdot \psi_2$. Furthermore, if β exists, then we obtain $K^{\psi_1} = K^{\psi_2}$ and $Ker(\psi_1)^\beta = Ker(\psi_2)$.

Suppose we have two homomorphisms ψ_1 and ψ_2 from K into $\mathrm{Aut}(S)$ such that $K^{\psi_1} = K^{\psi_2}$ and such that there exists an automorphism $\delta \in \mathrm{Aut}(K)$ with $Ker(\psi_1)^\delta = Ker(\psi_2)$. We define $\psi_2' := \delta \cdot \psi_2$. Clearly, there exists an automorphism $\beta \in \mathrm{Aut}(K)$ with $\psi_1 = \beta \cdot \psi_2$, if and only if there exists an automorphism $\beta' \in \mathrm{Aut}(K)$ with $\psi_1 = \beta' \cdot \psi_2'$. Moreover, ψ_1 and ψ_2' have the same image, say H, and the same kernel, say $N \trianglelefteq K$. Therefore, if β' exists, then it has to stabilize N.

The homomorphisms $K \to \mathrm{Aut}(S)$ with kernel N and image H are in one-to-one correspondence with the isomorphisms $K/N \to H$. They may be obtained by choosing a fixed isomorphism $\iota : K/N \to H$ and computing $\gamma \cdot \iota$ for all $\gamma \in \mathrm{Aut}(K/N)$. Then two isomorphisms $\gamma_1 \cdot \iota$ and $\gamma_2 \cdot \iota$ lead to isomorphic split extensions, if and only if there exists β' in the stabilizer $\mathrm{Aut}(K)_N$ of N in $\mathrm{Aut}(K)$ with $\gamma_1 \cdot \iota = \overline{\beta'} \cdot \gamma_2 \cdot \iota$ where $\overline{\beta'}$ is the automorphism of K/N induced by β'. That means that γ_1 and γ_2 are in the same left coset of the subgroup $\overline{\mathrm{Aut}(K)_N}$ of $\mathrm{Aut}(K/N)$ induced by the stabilizer of N in $\mathrm{Aut}(K)$.

This leads to a practical method to construct a set of operation homomorphisms corresponding to isomorphism type representatives of extensions of K by S. First, we compute up to conjugacy in $\mathrm{Aut}(K)$ all possible kernels of operation homomorphisms; that is, all normal subgroups N with cyclic factor group of order dividing $q-1$. As $\mathrm{Aut}(S)$ is cyclic, there exists exactly one subgroup H of $\mathrm{Aut}(S)$ isomorphic to K/N for each N. Next we

have to compute the subgroup $\overline{\mathrm{Aut}(K)_N}$ of $\mathrm{Aut}(K/N)$ induced by the stabilizer of N in $\mathrm{Aut}(K)$. Note that K/N is cyclic and hence $\mathrm{Aut}(K/N)$ is trivial to obtain. Then we compute a set of coset representatives of $\overline{\mathrm{Aut}(K)_N}$ in $\mathrm{Aut}(K/N)$. This set corresponds to a set of operation homomorphisms from $K \to \mathrm{Aut}(S)$ with kernel N and image H. This set in turn leads to isomorphism type representatives of those split extensions which correspond to an operation homomorphism with image H and kernel conjugate to N under $\mathrm{Aut}(K)$.

In Figure 3 we summarize the algorithm to construct all groups of the form $S \rtimes K$ up to isomorphism.

5.2. THE CASE $K \cong C_q$ AND S OF ORDER p^n

Let x be a generator of the cyclic group K. Then, according to Theorem 5.1, two homomorphisms ψ_1 and ψ_2 from K to $\mathrm{Aut}(S)$ lead to isomorphic split extensions, if and only if there exist two automorphisms $\alpha \in \mathrm{Aut}(S)$ and $\beta \in \mathrm{Aut}(K)$ with $(x^{\psi_1})^\alpha = (x^\beta)^{\psi_2}$.

The automorphisms of K permute the generators of the cyclic group K. Thus to check whether two homomorphisms ψ_1 and ψ_2 lead to isomorphic split extensions it is therefore enough to check whether there exists an automorphism $\alpha \in \mathrm{Aut}(S)$ which conjugates a generator of K^{ψ_1} onto a generator of K^{ψ_2}. Thus we have to check whether K^{ψ_1} is conjugate to K^{ψ_2}, because the two subgroups of $\mathrm{Aut}(S)$ are cyclic.

The homomorphic images of K in $\mathrm{Aut}(S)$ have order 1 or q. Hence to determine a set of operation homomorphisms as desired we have to compute the conjugacy classes of cyclic subgroups of order q in $\mathrm{Aut}(S)$. The algorithm is outlined in Figure 4.

The cyclic split extension method is the combination of both algorithms. It is straightforward to restrict this method to construct non-nilpotent groups only, as the only nilpotent group constructed by the two above algorithms is the split extension corresponding to the trivial homomorphism.

5.3. EXAMPLE

We give an example of the algorithm $CyclicSplitExtensionsDown(S, K)$ for $S := C_5$ and $K := C_{16} \rtimes \mathrm{Aut}(C_{16})$. As $\mathrm{Aut}(C_5) \cong C_4$, we have to compute the normal subgroups of K with cyclic factor group of order 2 or 4. We obtain 7 subgroups of index 2 and 4 subgroups of index 4.

The 7 subgroups of index 2 are characteristic in K. As the automorphism group $\mathrm{Aut}(K/N)$ for N of index 2 is trivial, we obtain one split extension $S \rtimes K$ for each of the 7 subgroups.

The 4 subgroups with factor isomorphic to C_4 fall into two orbits of length 2 under action of $\mathrm{Aut}(K)$. For each of these subgroups N we have that $\mathrm{Aut}(K)_N$ induces the trivial automorphism group on K/N. As $\mathrm{Aut}(K/N)$ has order two, we obtain for each of the orbit representatives two non-isomorphic split extensions $S \rtimes K$.

Altogether we determine 11 non-isomorphic non-nilpotent split extensions $S \rtimes K$.

6. Upwards Extension Method

In this section we describe a general method to construct the finite groups of given order o. Our aim is to use this method for the determination of insoluble groups. Note

```
CyclicSplitExtensionsDown(S, K)
initialize result as empty list
compute Aut(K)
determine a list norms of subgroups N ⊴ K with factor cyclic of order dividing q − 1
compute orbits and stabilisers in norms under action of Aut(K)
for each orbit representative N do
    determine the cyclic subgroup H of Aut(S) of order |K/N|
    choose an isomorphism ι : K/N → H
    compute the subgroup Aut(K)_N of Aut(K/N) induced by Aut(K)_N
    for each coset representative γ of Aut(K)_N in Aut(K/N) do
        compute the isomorphism ψ := γ · ι
        compute the corresponding homomorphism ψ : K → Aut(S)
        compute the split extension G := S ⋊_ψ K
        add G to the list of split extensions result
    od
od
return result
```

Figure 3. The algorithm *CyclicSplitExtensionsDown*

```
CyclicSplitExtensionsUp(S, K)
initialize result as empty list
compute Aut(S)
compute conjugacy class representatives of cyclic subgroups of order q in Aut(S)
for each representative U do
    choose an isomorphism ψ : K → U
    compute G := S ⋊_ψ K
    add G to the list result
od
add the direct product S × K to the list result
return result
```

Figure 4. The algorithm *CyclicSplitExtensionsUp*.

that a similar approach has been proposed in Laue (1982) and used in Betten (1996) to construct finite soluble groups.

Let G be a finite group and let G_S be the soluble residuum of G; that is, G_S is the smallest normal subgroup of G with soluble factor group G/G_S. Then G_S is a perfect subgroup of G which is invariant under isomorphisms. Moreover, as G/G_S is soluble, there exists a subnormal series

$$G := C_1 \rhd C_2 \rhd \cdots \rhd C_n \rhd C_{n+1} = G_S$$

from G down to G_S with factors of prime order, say $[C_i : C_{i+1}] = p_i$.

We want to determine up to isomorphism all finite groups G of order o having a given perfect group P as soluble residuum G_S. For this purpose we will use iterated cyclic extensions upwards along a subnormal series with factors of prime order. Note that there are catalogs containing the perfect groups of order less than 10^4, see Sandlöbes (1981) and Holt and Plesken (1989), which can be used to obtain the perfect group P.

In Section 6.1 we describe a method to construct cyclic upwards extensions of a finite group N; that is, groups G of order $|N| \cdot p$ for a prime p which have a normal subgroup isomorphic to N. By iterated use of this method we obtain a set of groups of a given order o with soluble residuum P which contains each isomorphism type of extension at least once. However, non-isomorphic groups may have isomorphic cyclic upwards extensions

and thus we may obtain isomorphic groups by this approach. Hence we have to reduce the computed list of groups to isomorphism types as final step in this algorithm. For a method to test isomorphism of finite groups see Hulpke (1996).

6.1. UPWARDS CYCLIC EXTENSIONS

Suppose we have a group N and a prime p given and we want to construct all upwards cyclic extensions of N of order $|N| \cdot p$. For any such extension G there exists $g \in G$ with $G = \langle g, N\rangle$ and g induces an automorphism α of N by its conjugation action. To describe G uniquely it suffices to know N, the automorphism $\alpha \in \mathrm{Aut}(N)$ and the element $n \in N$ with $g^p = n$, because with this knowledge one can extend a finite presentation of N to a finite presentation of G.

Thus we want to determine a set of tuples (α, n) describing the upwards extension of N by C_p. It is necessary and sufficient for all such tuples that conjugation by n induces the inner automorphism α^p and $n^\alpha = n$ holds.

To determine all tuples with these two conditions, we may first compute the elements α of $\mathrm{Aut}(N)$ with $\alpha^p \in Inn(N)$, see Hulpke (1996) for a method to compute $\mathrm{Aut}(N)$. Then it is straightforward to obtain an element $m \in N$ inducing the inner automorphism α^p. Thus the elements in the coset $mZ(N)$ are exactly the elements of N inducing α^p and it remains to compute the fixed points n in $mZ(N)$ under action of α.

As we are only interested in isomorphism type representatives of upwards cyclic extensions, we do not need to compute all tuples (α, n) with the above conditions. In particular, it is not necessary to examine all elements $\alpha \in \mathrm{Aut}(N)$ with $\alpha^p \in Inn(N)$ in the first step, but it is enough for our purposes to consider a generator of each conjugacy class representative of cyclic subgroups of order p in the factor $\mathrm{Aut}(N)/Inn(N)$.

References

Besche, H. U., Eick, B. (1998). The groups of order at most 1000 except 512 and 768. *J. Symb. Comput.*, **27**, 403–411.

Betten, A. (1996). Parallel construction of finite soluble groups. In *Proceedings Euro PVM '96, Munich*, LNCS **1156**, pp. 126–133.

Cannon, J., Leedham-Green, C. R. (1998). Special presentations of finite soluble groups. In preparation.

Cayley, A. (1854). On the theory of groups, as depending on the symbolic equation $\theta^n = 1$. *Phil. Mag.*, **7**, 40–47.

Crelle (1936). Preisaufgabe der Fürstlich Jablonowskischen Gesellschaft für das Jahr 1938. *J. Reine Angew. Math.*, **175**, 252.

Eick, B. (1996). Charakterisierung und Konstruktion von Frattinigruppen und Anwendungen in der Konstruktion endlicher Gruppen. Ph.D. thesis, RWTH Aachen. Also available as: Aachener Beiträge zur Mathematik 17.

Eick, B. (1997). Special presentations of finite soluble groups and computing (pre-) frattini subgroups. In Finkelstein, L. and Kantor, W., eds, *DIMACS series 'Groups and Computation' 1995*, pp. 101–112. Providence, RI, American Mathematical Society.

Gaschütz, W. (1953). Über die Φ-Untergruppe endlicher Gruppen. *Math. Z.*, **58**, 160–170.

Gaschütz, W. (1954). Endliche Gruppen mit treuen absolut irreduziblen Darstellungen. *Math. Nachr.*, **12**, 253–255.

Gaschütz, W. (1963). Zur Theorie der endlichen auflösbaren Gruppen. *Math. Z.*, **80**, 300–305.

Hall, Jr, M., Senior, J. K. (1964). *The Groups of Order* 2^n *(*$n \leq 6$*)*. New York, The Macmillan Company.

Hölder, O. (1893). Die Gruppen der Ordnungen p^3, pq^2, pqr, p^4. *Math. Ann.*, **43**, 301–412.

Holt, D. F., Plesken, W. (1989). *Perfect Groups*. Oxford, Clarendon Press.

Hulpke, A. (1996). Konstrukton transitiver Pemutationsgruppen. Ph.D. thesis, RWTH Aachen. Also available as: Aachener Beiträge zur Mathematik 18.

James, R. (1980). The groups of order p^6 (p an odd prime). *Math. Comp.*, **34**, 613–637.

Laue, R. (1982). Zur Konstruktion und Klassifikation endlicher auflösbarer Gruppen. Bayreuther Mathematische Schriften 9.

Laue, R., Neubüser, J., Schoenwaelder, U. (1984). Algorithms for finite soluble groups and the sogos system. In Atkinson, M. D., ed., *Proc. LMS Symposium on Computational Group Theory, Durham 1982*, pp. 105–135. New York, Academic Press.

Magnus, W. (1937). Neuere Ergebnisse über auflösbare Gruppen. *Jahresberich der DMV*, **47**, 69.

Neubüser, J. (1967). Die Untergruppenverbände der Gruppen der Ordnung ≤ 100 mit Ausnahme der Ordnungen 64 und 96. Habilitationsschrift an der Universität Kiel.

O'Brien, E. A. (1990). The p-group generation algorithm. *J. Symb. Comput.*, **9**, 677–698.

O'Brien, E. A. (1991). The groups of order 256. *J. Alg.*, **143**, 219–235.

Plesken, W. (1987). Towards a soluble quotient algorithm. *J. Symb. Comput.*, **4**, 111–122.

Remak, R. (1930). Über die Darstellung endlicher Gruppen als Untergruppen direkter Produkte. *J. Reine Angew. Math.*, **163**, 1–44.

Robinson, D. J. S. (1981). Applications of cohomology to the theory of groups. In Campbell, C. M. and Robertson, E. F., eds, *Groups - St. Andrews 1981*, LMS Lecture Note Series 71, pp. 46–80. Cambridge, Cambridge University Press.

Sandlöbes, G. (1981). Perfect groups of order less than 10^4. *Commun. Alg.*, **9**, 477–490.

Short, M. W. (1992). In *The primitive soluble permutation groups*. LNM 1519. Heidelberg, Springer.

Smith, M. J. (1995). Computing automorphisms of finite soluble groups. Ph.D. thesis, Australian National University, Canberra.

Taunt, D. R. (1955). Remarks on the isomorphism problem in theories of construction of finite groups. *Proc. Cambridge Phil. Soc.*, **51**, 16–24.

Wegner, A. (1992). The construction of finite soluble factor groups of finitely presented groups and its application. Ph.D. thesis, University of St. Andrews.

Originally Received 11 March 1997
Accepted 23 November 1998