

**The nonabelian tensor product of
groups: computations and structural
results**

**The nonabelian tensor product of
groups: computations and structural
results**

Aidan McDermott

Department of Mathematics

Faculty of Arts

National University of Ireland

Galway

February 1998

Supervisor: Dr. Graham Ellis

Contents

Acknowledgements	i
Summary	ii
1 The nonabelian tensor	
product of groups	1
1.1 Introduction	1
1.2 The tensor product	2
1.3 The q -tensor product	22
2 The computation of the nonabelian tensor product	
of finite groups	31
2.1 Introduction	31
2.2 Theoretical background	32
2.3 Algorithms and programs	39
2.4 Examples	40

3	Tensor products of prime–power groups	49
3.1	Introduction	49
3.2	Group-theoretic results	50
3.3	A topological application	60
3.4	Computer computations	66
4	Bounds on nonabelian tensor squares of prime–power groups	69
4.1	Introduction	69
4.2	Preliminaries	70
4.3	Bounds on the order of $\mathbf{G} \otimes \mathbf{G}$	76
4.4	Examples	80
4.5	The \mathbf{p} –tensor product	81
4.6	Placing bounds on the order of the kernel J	88
5	The computation of the Frattini	
	subgroup of prime–power groups	93
5.1	Introduction	93
5.2	Definitions and Notation	93
6	Groups with perfect subgroups	104
6.1	Introduction	104

6.2	Results	104
6.3	Example	107
Appendix A: Functions for computing tensor products		110
Appendix B: Programs for Chapter 2		119
Appendix C: Programs for Chapter 4		145
Appendix D: Programs for Chapter 5		149

Acknowledgements

I would like to thank my supervisor Dr. Graham Ellis for all his help and encouragement while working on this thesis. I would also like to thank my family for their support.

The research in this thesis was funded by Fórbairt.

Summary

Chapter 1 of this thesis recalls definitions and results on nonabelian tensor products needed in subsequent chapters. In Chapter 2 we improve on an algorithm for computing $G \otimes H$ and implement the algorithm in GAP with specifications

- a. finite groups are entered as finite presentations,
- b. actions between groups defined on generators.

In Chapter 3 we give bounds on the order of $G \otimes H$ when G is a p -group and H is a q -group, p and q primes. As a topological application we consider a union of CW-spaces $X = A \cup B$; we give a bound on $\pi_3 X$ in terms of the homotopy groups of A , B and $A \cap B$, and a certain tensor product.

In Chapter 4 we give bounds on the order and exponent of $G \otimes G$ when G is a p -group.

By placing bounds on the kernel J ,

$$J \rightarrow G \otimes G \rightarrow [G, G]$$

we derive an analogue of a result of Berkovitch on the Schur Multiplier, namely:

Theorem: Let G be a group of order p^n and suppose $|J| = p^{n^2-l}$. Then

i) $l = 0$ if and only if G is elementary abelian,

ii) $l \neq 1$ or 3 ,

iii) $l = 2$ if and only if $G \cong C_{p^2}$,

iv) $l = 4$ if and only if G is isomorphic to one of $C_{p^2} \times C_p \times C_p$, $C_{p^2} \times C_p$, or

$p = 2$ and $G \cong \langle a, b, c | b^p = a, c^p = a, a^p = [a, b] = [a, c] = 1, [b, c] = a \rangle$,

v) if $l = 5$ then either $p > 1000$ or $p = 2$ and $G \cong \langle a, b, c | a^2 = b^2 = c^2 = [a, b] =$

$[a, c] = 1, [b, c] = a \rangle$,

where C_m denotes the cyclic group of order m .

In Chapter 5 we give an algorithm for solving the following problem. Given a finite presentation for a group Q , determine all possible $[G, G]G^q$, where $Q = G/Z(G, q)$, q is an integer and $Z(G, q)$ is defined as $\{g \in Z(G) : g^q = 1\}$.

In Chapter 6 we prove the following:

Theorem: Let G be a group and N a perfect normal subgroup of G . The sequence

$$M(N) \longrightarrow J(G) \longrightarrow J(G/N) \longrightarrow 1$$

is exact, where M denotes the Schur multiplier and $J(G)$ denotes the kernel of the canonical homomorphism $G \otimes G \longrightarrow G$.

Chapter 1

The nonabelian tensor product of groups

1.1 Introduction

The nonabelian tensor square $G \otimes G$ of a group G was introduced by R.K. Dennis [14] in a search for “homology functors having a close relationship to K–theory.” It is based on the work of C. Miller [47] who studied what would now be called the exterior square $G \wedge G$ of a group and showed the Schur Multiplier $M(G)$ can be described as

$$M(G) \cong \ker(G \wedge G \rightarrow G).$$

Subsequently R. Brown and J.-L. Loday [9] discovered a topological significance for the tensor square, namely, that the third homotopy group of the suspension of an Eilenberg MacLane space $K(G, 1)$ satisfies

$$\pi_3 SK(G, 1) \cong \ker(G \otimes G \rightarrow G).$$

They also defined the tensor product $G \otimes H$ of two distinct groups acting “com-

patibly” on each other and showed that it arose in a certain “universal crossed square.” The exterior product $G \wedge H$ was also defined and used in an exact homology sequence with coefficients in \mathbb{Z} .

The q -exterior square $G \wedge^q G$ was defined by G. Ellis [22] who showed that $F \wedge^q F \cong [F, F]F^q$ for a free group F . The q -exterior product of two groups N and G was defined in the case N is a normal subgroup of G by G. Ellis and C. Rodríguez-Fernández in [26] and the q -tensor product was defined under the same conditions by R. Brown [7]. Finally, D. Conduché and C. Rodríguez-Fernández defined the q -tensor product of groups, $G \otimes^q H$, when the groups act compatibly [13]. The construction $G \otimes^q H$ reduces to Brown and Loday’s tensor product when $q = 0$.

In this chapter we recall the definitions and results on nonabelian tensor products that are needed for subsequent chapters. The bibliography at the end of the thesis contains a fairly comprehensive list of the papers on the tensor product.

1.2 The tensor product

Let G and H be groups. By an action of G on H we mean a homomorphism

$$\phi : G \longrightarrow \text{Aut}(H)$$

where $\text{Aut}(H)$ is the group of automorphisms of H . If $g \in G$ and $h \in H$ then we write ${}^g h$ for $\phi(g)(h)$. If ϕ is the trivial homomorphism then we say G acts *trivially* on H . If $H = G$ and ${}^g g' = gg'g^{-1}$ for all $g, g' \in G$ we say G acts *on itself by conjugation*. We shall always assume that the action of a group on itself

is by conjugation. In keeping with this assumption we set $[x, y] = xyx^{-1}y^{-1}$ for elements x, y lying in a group G .

1.2.1 Definition [9] Suppose G and H are groups which act on themselves by conjugation and on each other so that

$${}^{(gh)}(g') = g({}^h(g^{-1}g')), \quad {}^{(hg)}(h') = h(g({}^{h^{-1}}h')),$$

for all $g, g' \in G$ and $h, h' \in H$. We call such an action a *compatible* action and say G and H act on each other *compatibly*.

1.2.2 Remarks Throughout this thesis we write ${}^{gh}g'$ to stand for $g({}^hg')$.

1.2.3 Remarks Throughout this chapter we assume that G and H act compatibly on one another.

1.2.4 Definition [9] Suppose G and H act on each other compatibly. We define the *nonabelian tensor product* of G and H to be the group $G \otimes H$ generated by the symbols $g \otimes h$ where $g \in G$ and $h \in H$, subject to the relations

$$gg' \otimes h = ({}^gg' \otimes {}^gh)(g \otimes h) \tag{1.1}$$

$$g \otimes hh' = (g \otimes h)({}^hg \otimes {}^hh') \tag{1.2}$$

for $g, g' \in G, h, h' \in H$.

1.2.5 Definition [9] Let $G, H,$ and K be groups and suppose G and H act on each other. A function $\phi : G \times H \rightarrow K$ is a *crossed pairing* if

$$\phi(gg', h) = \phi({}^gg', {}^gh)\phi(g, h),$$

$$\phi(g, hh') = \phi(g, h)\phi({}^h g, {}^h h'),$$

for all $g, g' \in G$ and $h, h' \in H$.

We note the resemblance between the identities in the definition of a crossed pairing and (1.1) and (1.2). Because of this resemblance each crossed pairing ϕ determines a unique homomorphism $\hat{\phi} : G \otimes H \rightarrow K$ defined on generators by $\hat{\phi}(g \otimes h) = \phi(g, h)$ for $g \in G$ and $h \in H$. In particular if G and H are abelian and act trivially on each other a crossed pairing is simply a bilinear map. In this case we see that $G \otimes H$ is the usual tensor product $G \otimes_{\mathbb{Z}} H$ of abelian groups or \mathbb{Z} modules (see for example [49]).

1.2.6 Proposition [9] *Suppose the groups G and H act on each other compatibly.*

Then G and H act on $G \otimes H$ by

$${}^g(g' \otimes h) = ({}^g g' \otimes {}^g h), \quad {}^h(g \otimes h') = ({}^h h' \otimes {}^h h')$$

for all $g, g' \in G$ and $h, h' \in H$.

Proof: We must show that the actions defined above respect the identities (1.1) and (1.2). Let $g, g' \in G$, $h \in H$ and p be an arbitrary element of G or H . Then

$$\begin{aligned} {}^p(gg' \otimes h) &= {}^p(gg') \otimes {}^p h \\ &= {}^p g^p g' \otimes {}^p h \\ &= ({}^{(p)g}({}^p g') \otimes ({}^{(p)g}({}^p h)))({}^p g \otimes {}^p h) && \text{using compatibility} \\ &= ({}^{pgp^{-1}}({}^p g') \otimes {}^{pgp^{-1}}({}^p h))({}^p g \otimes {}^p h) \\ &= ({}^{p^g} g' \otimes {}^{p^g} h)({}^p g \otimes {}^p h) \end{aligned}$$

$$\begin{aligned}
&= {}^p({}^g g' \otimes {}^g h)({}^p(g \otimes h)) \\
&= {}^p[({}^g g' \otimes {}^g h)(g \otimes h)].
\end{aligned}$$

Thus (1.1) is respected. A similar argument shows that (1.2) is respected. ■

1.2.7 Proposition [9] *There exists an isomorphism $G \otimes H \xrightarrow{\cong} H \otimes G$ such that $g \otimes h \mapsto (h \otimes g)^{-1}$ for all $g \in G$ and $h \in H$.*

Proof: Define the map $\phi : G \times H \rightarrow H \otimes G$ by $(g, h) \mapsto (h \otimes g)^{-1}$. Then ϕ is a crossed pairing since,

$$\begin{aligned}
\phi(gg', h) &= (h \otimes gg')^{-1} \\
&= [(h \otimes g)({}^g(h \otimes g'))]^{-1} \\
&= ({}^g h \otimes {}^g g')^{-1}(h \otimes g)^{-1} \\
&= \phi({}^g g', {}^g h)\phi(g, h),
\end{aligned}$$

and

$$\begin{aligned}
\phi(g, hh') &= (hh' \otimes g)^{-1} \\
&= [({}^h(h' \otimes g))(h \otimes g)]^{-1} \\
&= (h \otimes g)^{-1}({}^h h' \otimes {}^h g)^{-1} \\
&= \phi(g, h)\phi({}^h g, {}^h h').
\end{aligned}$$

Therefore ϕ gives rise to a homomorphism $\phi^* : G \otimes H \rightarrow H \otimes G$ defined on generators by $\phi(g \otimes h) = (h \otimes g)^{-1}$. This map is clearly surjective and by reversing the roles of G and H we get a homomorphism $\psi^* : H \otimes G \rightarrow G \otimes H$ such that $\psi^*(h \otimes g) = (g \otimes h)^{-1}$. The map ψ^* is the inverse of ϕ^* and so ϕ^* is the required isomorphism. ■

1.2.8 Proposition [9] *The following identities hold in $G \otimes H$ for all $g, g' \in G$ and $h, h' \in H$.*

$$1_G \otimes h = 1_{G \otimes H} = g \otimes 1_H \quad (1.3)$$

$$(g \otimes h)^{-1} = {}^g(g^{-1} \otimes h) = {}^h(g \otimes h^{-1}) \quad (1.4)$$

$$(g \otimes h)(g' \otimes h')(g \otimes h)^{-1} = [{}^{g,h}] (g' \otimes h') \quad (1.5)$$

$$(g^h g^{-1}) \otimes h' = (g \otimes h)^{h'} (g \otimes h)^{-1} \quad (1.6)$$

$$g' \otimes ({}^g h h^{-1}) = g' (g \otimes h) (g \otimes h)^{-1} \quad (1.7)$$

$$[g \otimes h, g' \otimes h'] = (g^h g^{-1}) \otimes (g' h' h'^{-1}) \quad (1.8)$$

Proof: We derive the left hand side of (1.3) by taking g and g' to be the identity in (1.1). The right hand side follows similarly by taking h and h' to be the identity in (1.2).

Taking $g' = g^{-1}$ in (1.1) we have

$$\begin{aligned} 1_G \otimes h &= (g^{-1} \otimes {}^g h)(g \otimes h) \\ &= {}^g(g^{-1} \otimes h)(g \otimes h). \end{aligned}$$

It follows that

$$(g \otimes h)^{-1} = {}^g(g^{-1} \otimes h).$$

In a similar manner using (1.2) and taking $h' = h^{-1}$ we get

$$(g \otimes h)^{-1} = {}^h(g \otimes h^{-1}).$$

Consider $g g' \otimes h h'$. We can expand this in two ways. First using (1.1),

$$g g' \otimes h h' = ({}^g g' \otimes {}^g(h h'))(g \otimes h h')$$

$$\begin{aligned}
&= {}^g(g' \otimes hh')(g \otimes hh') \\
&= {}^g\{(g' \otimes h)({}^h g' \otimes {}^h h')\}(g \otimes h)({}^h g \otimes {}^h h') \\
&= {}^g(g' \otimes h)g^h(g' \otimes h')(g \otimes h)(g \otimes h').
\end{aligned}$$

Again using (1.2),

$$\begin{aligned}
gg' \otimes hh' &= ({}^g g' \otimes h)({}^h(gg') \otimes hh') \\
&= ({}^g g' \otimes {}^g h)(g \otimes h)({}^h[({}^g g' \otimes {}^g h')(g \otimes h)]) \\
&= {}^g(g' \otimes h)(g \otimes h)g^h(g' \otimes h')g(g \otimes h').
\end{aligned}$$

Equating we have

$$g^h(g' \otimes h')(g \otimes h) = (g \otimes h)g^h(g' \otimes h')$$

which implies

$$g^h(g' \otimes h') = (g \otimes h)g^h(g' \otimes h')(g \otimes h)^{-1}$$

Now if we let $x = {}^h g'$ and $y = {}^h h'$ then we have

$$g^h(g^{-1}h^{-1}x \otimes g^{-1}h^{-1}y) = (g \otimes h)(x \otimes y)(g \otimes h)^{-1}$$

and so

$$[g,h](x \otimes y) = (g \otimes h)(x \otimes y)(g \otimes h)^{-1}.$$

Using (1.1) we have

$$\begin{aligned}
(g^h g^{-1}) \otimes h' &= {}^g({}^h g^{-1} \otimes h')(g \otimes h') \\
&= g^h(g^{-1} \otimes h^{-1}h'h)(g \otimes h') \\
&= {}^g(g^{-1} \otimes h)^{-1}g(g^{-1} \otimes h'h)(g \otimes h') \\
&= (g \otimes h)g(g^{-1} \otimes h')g^h(g^{-1} \otimes h)(g \otimes h')
\end{aligned}$$

$$\begin{aligned}
&= (g \otimes h)(g \otimes h')^{-1gh'}(g^{-1} \otimes h)(g \otimes h') \\
&= (g \otimes h)^{h'g}(g^{-1} \otimes h) \\
&= (g \otimes h)^{h'}(g \otimes h)^{-1}.
\end{aligned}$$

Equation (1.7) is proved similarly using (1.2). Finally starting with (1.6) we get

$$\begin{aligned}
(g^h g^{-1}) \otimes (g' h' h'^{-1}) &= (g \otimes h)^{(g' h' h'^{-1})}(g \otimes h)^{-1} \\
&= (g \otimes h)^{[g', h']}(g \otimes h)^{-1} \\
&= (g \otimes h)[(g' \otimes h')(g \otimes h)(g' \otimes h')^{-1}]^{-1} \\
&= [g \otimes h, g' \otimes h'].
\end{aligned}$$

■

In our discussion of the nonabelian tensor product it will be useful to have the following algebraic concept (introduced by J.H.C. Whitehead [56] in work on homotopy theory).

1.2.9 Definition [56] A *crossed N -module* is a homomorphism of groups $\lambda : M \rightarrow N$ together with a group action $(n, m) \rightarrow {}^n m$ of N on M such that

$$\lambda({}^n m) = n\lambda(m)n^{-1}, \quad (1.9)$$

$$mm'm^{-1} = \lambda({}^m)m', \quad (1.10)$$

for all $m, m' \in M$ and $n \in N$. For q a non-negative integer, a *q -crossed N -module* was defined in [24] to be a crossed module λ for which $a^q = 1$ for all $a \in \ker \lambda$.

When the context makes it clear what the group N is we often use the terms *crossed module* and *q -crossed module*.

1.2.10 Definition Let $\lambda : M \rightarrow N$ and $\lambda' : M' \rightarrow N$ be two (q -)crossed N -modules. A *morphism* of the (q -)crossed N -modules is a homomorphism $\phi : M \rightarrow$

M' such that $\lambda'\phi(m) = \lambda(m)$, and $\phi({}^n m) = {}^n \phi(m)$ for $m \in M$ and $n \in N$.

In other words a morphism is a commutative triangle

$$\begin{array}{ccc} M & \xrightarrow{\phi} & M' \\ & \searrow \lambda & \swarrow \lambda' \\ & & N \end{array}$$

with ϕ a group homomorphism satisfying $\phi({}^n m) = {}^n \phi(m)$.

We let $Crossed(N)$ denote the category whose objects are crossed N -modules, and whose arrows are morphisms of crossed N -modules. We let $q\text{-Crossed}(N)$ denote the category whose objects are q -crossed N -modules, and whose arrows are morphisms of q -crossed N -modules.

1.2.11 Definition (See [46] for example.) An object P in a category \mathcal{C} is said to be *projective* if for any diagram

$$\begin{array}{ccc} & & P \\ & & \downarrow g \\ A & \xrightarrow{f} & B \end{array}$$

in \mathcal{C} with f an epimorphism, there is a commutative diagram

$$\begin{array}{ccc} & & P \\ & \swarrow \dots & \downarrow g \\ A & \xrightarrow{f} & B \end{array}$$

in \mathcal{C} .

1.2.12 Definition The term *projective* crossed N -module (*projective* q -crossed

N -module) is used to mean a projective object in the category $\text{Crossed}(N)$ (respectively $q\text{-Crossed}(N)$).

1.2.13 Proposition [9] *Let G and H act on each other compatibly.*

(a) *There are homomorphisms*

$$\lambda : G \otimes H \rightarrow G,$$

$$\lambda' : G \otimes H \rightarrow H$$

such that $\lambda(g \otimes h) = g^h g^{-1}$, and $\lambda'(g \otimes h) = {}^g h h^{-1}$ for all $g \in G$ and $h \in H$.

(b) *The homomorphisms λ, λ' together with the actions of G and H on $G \otimes H$ given in Proposition 1.2.4 form crossed modules.*

(c) *If $x \in G \otimes H$, $g \in G$, $h \in H$, then*

$$\lambda(x) \otimes h = x^h x^{-1}$$

$$g \otimes \lambda'(x) = {}^g x x^{-1}.$$

(d) *The actions of G on $\text{Ker} \lambda'$, and H on $\text{Ker} \lambda$ are trivial.*

(e) *For all $x, y \in G \otimes H$,*

$$[x, y] = \lambda(x) \otimes \lambda'(y)$$

Proof: (a) Define $\phi : G \times H \rightarrow G$ by $(g, h) \mapsto g^h g^{-1}$. Then ϕ is a crossed pairing, since

$$\phi(gg', h) = gg'^h (gg')^{-1}$$

$$\begin{aligned}
&= gg'^h g'^{-1h} g^{-1} \\
&= gg'g^{-1gh} g'^{-1h} g^{-1} \\
&= gg'g^{-1gh} g'^{-1} g^h g^{-1} \\
&= {}^g g' g^h g'^{-1} g^h g^{-1} \\
&= {}^g g' g^h g^{-1} ({}^g g')^{-1} g^h g^{-1} \\
&= {}^g g'^{gh} ({}^g g')^{-1} g^h g^{-1} \\
&= \phi({}^g g', {}^g h) \phi(g, h),
\end{aligned}$$

and we can show in a similar fashion that

$$\phi(g, hh') = \phi(g, h) \phi({}^h g, {}^h h').$$

So ϕ gives rise to a group homomorphism $\phi^* : G \otimes H \rightarrow G$, defined on generators by $\phi^*(g \otimes h) = \phi(g, h)$. Take λ to be this map ϕ^* .

In a similar manner we can define the crossed pairing $\psi : G \otimes H \rightarrow H$, $(g, h) \mapsto {}^g h h^{-1}$ and take λ' to be the associated homomorphism $\lambda' : G \otimes H \rightarrow H$, $\lambda'(g, h) = {}^g h h^{-1}$.

(b) We verify the crossed module identities (1.9) and (1.10) by considering the special case of generating tensors.

$$\begin{aligned}
\lambda({}^g(g' \otimes h)) &= \lambda({}^g g' \otimes {}^g h) \\
&= {}^g g' {}^g h ({}^g g')^{-1} \\
&= gg'g^{-1} g^h g^{-1} ({}^g g')^{-1} \\
&= gg'g^{-1} g^h g'^{-1} \\
&= gg'g^{-1} g^h g'^{-1} g^{-1}
\end{aligned}$$

$$\begin{aligned}
&= gg'{}^h g'^{-1} g^{-1} \\
&= g\lambda(g' \otimes h)g^{-1}.
\end{aligned}$$

$$\begin{aligned}
\lambda^{(g \otimes h)}(g' \otimes h') &= g^h g^{-1}(g' \otimes h') \\
&= [g, h](g' \otimes h') \\
&= (g \otimes h)(g' \otimes h')(g \otimes h)^{-1}.
\end{aligned}$$

using (1.5).

The general case now follows from the fact that both the action of G on $G \otimes H$ and the map λ are homomorphisms. In the same way we can show that λ' is a crossed module.

(c) Suppose $x = g' \otimes h'$ for some $g' \in G$ and $h' \in H$. Then from (1.6) we have

$$\begin{aligned}
\lambda(x) \otimes h &= (g'^h g'^{-1}) \otimes h \\
&= (g' \otimes h')^h (g' \otimes h')^{-1} \\
&= x^h x^{-1}.
\end{aligned}$$

so (c) holds in this case.

Now suppose $x = x_1 x_2$ and $\lambda(x_1) \otimes h = x_1^h x_1^{-1}$ and $\lambda(x_2) \otimes h = x_2^h x_2^{-1}$ for all $h \in H$. Then

$$\begin{aligned}
\lambda(x_1 x_2) \otimes h &= (\lambda(x_1) \lambda(x_2)) \otimes h \\
&= \lambda^{(x_1)}(\lambda(x_2) \otimes h)(\lambda(x_1) \otimes h) \\
&= \lambda^{(x_1)}(x_2^h x_2^{-1})(x_1^h x_1^{-1}) \\
&= x_1(x_2^h x_2^{-1})x_1^{-1}(x_1^h x_1^{-1}) \\
&= x_1 x_2^h (x_1 x_2)^{-1}.
\end{aligned}$$

The general case now follows from an induction of the length of x when written in the generators of $G \otimes H$. The second equation can be proved similarly.

(d) This follows immediately from part (c).

(e) Equation 1.8 shows that (e) holds when x and y are generators of $G \otimes H$.

Using part (c) and the fact that λ and λ' are crossed modules we get:

$$\begin{aligned} \lambda(x) \otimes \lambda'(y) &= x^{\lambda'(y)} x^{-1} \\ &= xyx^{-1}y^{-1} \\ &= [x, y]. \end{aligned} \quad \blacksquare$$

1.2.14 Proposition [9] *If G and H act trivially on one another, then*

$$G \otimes H \cong G^{ab} \otimes_{\mathbb{Z}} H^{ab}$$

where $\otimes_{\mathbb{Z}}$ denotes the usual tensor product of abelian groups and G^{ab} and H^{ab} are the abelianization of G and H respectively.

Proof: From Proposition 1.1.4 (b) we have $\lambda : G \otimes H \rightarrow G$, $g \otimes h \mapsto g^h g^{-1} = 1$. Therefore $\text{Ker} \lambda = G \otimes H$ and by part (e) of Proposition 1.1.4 we see that G acts trivially on $G \otimes H$. In the same way we find that H acts trivially on $G \otimes H$. It follows from (1.8) that $G \otimes H$ is abelian.

Suppose G' and H' are the derived subgroups of G and H respectively, and consider the mapping $\phi : G \times H \rightarrow G^{ab} \otimes_{\mathbb{Z}} H^{ab}$, defined by $(g, h) \mapsto \bar{g} \otimes_{\mathbb{Z}} \bar{h}$ where $\bar{g} = G'g$ and $\bar{h} = H'h$.

We must show that ϕ is well defined. Suppose $\bar{g}_1 = G'g_1 = \bar{g}$ and $\bar{h}_1 = H'h_1 = \bar{h}$.

Then $g_1 = mg$ and $h_1 = nh$ for some $m \in G'$ and $n \in H'$. Now

$$\begin{aligned}
\overline{g_1} \otimes_{\mathbb{Z}} \overline{h_1} &= \overline{mg} \otimes_{\mathbb{Z}} \overline{nh} \\
&= \overline{m}g \otimes_{\mathbb{Z}} \overline{n}h \\
&= (\overline{m} \otimes_{\mathbb{Z}} \overline{n})(\overline{m} \otimes_{\mathbb{Z}} \overline{h})(\overline{g} \otimes_{\mathbb{Z}} \overline{n})(\overline{g} \otimes_{\mathbb{Z}} \overline{h}) \\
&= \overline{g} \otimes_{\mathbb{Z}} \overline{h}.
\end{aligned}$$

Furthermore, ϕ is a crossed pairing for,

$$\begin{aligned}
\phi(gg', h) &= \overline{gg'} \otimes_{\mathbb{Z}} \overline{h} \\
&= (\overline{g} \otimes_{\mathbb{Z}} \overline{h})(\overline{g'} \otimes_{\mathbb{Z}} \overline{h}) \\
&= (\overline{g}g' \otimes_{\mathbb{Z}} \overline{g}h)(\overline{g} \otimes_{\mathbb{Z}} \overline{h}) \\
&= \phi({}^g g', {}^g h)\phi(g, h),
\end{aligned}$$

and similiary for the second crossed pairing identity.

Therefore there exists a homomorphism $\phi^* : G \otimes H \rightarrow G^{ab} \otimes_{\mathbb{Z}} H^{ab}$, such that

$$\phi^*(g \otimes h) = \overline{g} \otimes_{\mathbb{Z}} \overline{h}.$$

We show that ϕ^* is an isomorphism by exhibiting its inverse. Let $\psi : G^{ab} \otimes_{\mathbb{Z}} H^{ab} \rightarrow G \otimes H$ be the homomorphism defined on generators by $\overline{g} \otimes_{\mathbb{Z}} \overline{h} \mapsto g \otimes h$.

We now show that ψ is well defined by noting first that if $a, b \in G$ and $h \in H$ then since the actions of G and H are trivial,

$$\begin{aligned}
[a, b] \otimes h &= (b^{-1} \otimes h)(a^{-1} \otimes h)(b \otimes h)(a \otimes h) \\
&= 1_{G \otimes H},
\end{aligned}$$

since $G \otimes H$ is abelian. It follows that the canonical homomorphism $G' \otimes H \rightarrow G \otimes H$ is the trivial map. In the same way we can see that $G \otimes H' \rightarrow G \otimes H$ is

trivial.

Once again if $g_1 = mg$ and $h_1 = nh$ for $m \in G'$ and $n \in H'$ then

$$\begin{aligned}
g_1 \otimes h_1 &= mg \otimes nh \\
&= {}^m(g \otimes n) {}^{mn}(g \otimes h) (m \otimes n)^n (m \otimes h) \\
&= (g \otimes n)(g \otimes h)(m \otimes n)(m \otimes h) \\
&= g \otimes h,
\end{aligned}$$

where we make use of the fact that the actions of G and H on $G \otimes H$ are trivial. ■

1.2.15 Definition [9] Suppose G and H are normal subgroups of some group Q , and act by conjugation in Q . We define their *exterior product* $G \wedge H$ to be the quotient of $G \otimes H$ by the normal subgroup generated by all $x \otimes x$ with $x \in G \cap H$.

We denote the coset represented by $g \otimes h$ as $g \wedge h$.

1.2.16 Definition [9] Let G, H , and K be groups with G and H normal subgroups of some group Q . A function $\phi : G \times H \rightarrow K$ is a *exterior pairing* if

$$\begin{aligned}
\phi(gg', h) &= \phi({}^g g', {}^g h) \phi(g, h), \\
\phi(g, hh') &= \phi(g, h) \phi({}^h g, {}^h h'), \\
\phi(m, m) &= 1,
\end{aligned}$$

for all $g, g' \in G$ and $h, h' \in H$ and $m \in G \cap H$.

As with the tensor product and the crossed pairing, an exterior pairing $\phi : G \times H \rightarrow K$ determines a unique homomorphism $\hat{\phi} : G \wedge H \rightarrow K$.

1.2.17 Proposition *If $x, y \in G \cap H$, then*

$$(x \wedge y)^{-1} = y \wedge x$$

Proof: Expanding $xy \wedge xy$ we have

$$\begin{aligned} 1 &= xy \wedge xy = {}^x(y \wedge xy)(x \wedge xy) \\ &= {}^x(y \wedge x)^x(x \wedge y), \end{aligned}$$

which implies $(x \wedge y)(y \wedge x) = 1$. ■

1.2.18 Remarks It must be kept in mind that neither the tensor product nor the exterior product are determined by the isomorphism classes of the groups involved. In the case of the tensor product the actions of the groups come into play, while in the exterior product the intersection of the groups is also involved.

1.2.19 Example Let C_k denote a cyclic group of order k . Suppose A and B are cyclic groups of orders n and m respectively which act on each other trivially.

Then

- i) $A \otimes B \cong C_{hcf(n,m)}$,
- ii) $A \wedge B \cong C_{hcf(n,m)}$ if $A \cap B = 1$,
- iii) $A \wedge B \cong 1$ if $A = B$,
- iv) $A \otimes C_\infty \cong A$,

where $hcf(n, m)$ denotes the highest common factor of n and m .

(i) Let x generate A , y generate B . Then $A \otimes B \cong \langle x^i \otimes y^j, 0 \leq i < n, 0 \leq j < m \mid x^{i+l} \otimes y^{j+k} = (x^i \otimes y^j)(x^l \otimes y^k) \rangle$. It follows that $A \otimes B$ is cyclic with

generator $x \otimes y$. Furthermore, since $(x \otimes y)^n = x^n \otimes y = 1$ and $(x \otimes y)^m = x \otimes y^m = 1$ we have that $(x \otimes y)^{hcf(n,m)} = 1$. Now the map $\phi : A \times B \longrightarrow C_{hcf(n,m)}$ defined by $(x^i, y^j) \mapsto z^{ij}$ where z is a generator of $C_{hcf(n,m)}$ is a crossed pairing. There is then a unique homomorphism $\hat{\phi} : A \otimes B \longrightarrow C_{hcf(n,m)}$ from which (i) follows.

Parts (ii), (iii) and (iv) follow immediately.

1.2.20 Proposition [9] *Let A , B and C be groups with A acting on B and C , B and C acting on A , and, C acting on B such that*

$$(i) \quad {}^{bc}a = {}^{cb}a,$$

(ii) *the action of B on A and C induces the trivial action of B on $A \otimes C$,*

(iii) *the action of C on A and B induces the trivial action of C on $A \otimes B$,*

for all $a \in A$, $b \in B$ and $c \in C$. Then

$$A \otimes (B \times C) \cong (A \otimes B) \times (A \otimes C).$$

Proof: First note that (i) ensures that the action of $B \times C$ on A given by ${}^{(b,c)}a = {}^{bc}a$ is well defined.

Let $\phi : A \otimes (B \times C) \rightarrow (A \otimes B) \times (A \otimes C)$, be defined on the generators $\phi(a \otimes (b, c)) = (a \otimes b, a \otimes c)$.

Now ϕ is a crossed pairing since,

$$\begin{aligned} \phi(aa' \otimes (b, c)) &= (aa' \otimes b, aa' \otimes c) \\ &= ({}^a(a' \otimes b)(a \otimes b), {}^a(a' \otimes c)(a \otimes c)) \end{aligned}$$

$$\begin{aligned}
&= ({}^a a' \otimes {}^a b, {}^a a' \otimes {}^a c)(a \otimes b, a \otimes c) \\
&= \phi({}^a a' \otimes {}^a (b, c))\phi(a \otimes (b, c)) \\
&= \phi({}^a (a' \otimes (b, c)))\phi(a \otimes (b, c)),
\end{aligned}$$

and

$$\begin{aligned}
\phi(a \otimes (b, c)(b', c')) &= \phi(a \otimes (bb', cc')) \\
&= (a \otimes bb', a \otimes cc') \\
&= ((a \otimes b)^b(a \otimes b'), (a \otimes c)^c(a \otimes c')) \\
&= (a \otimes b, a \otimes c)^{(b(a \otimes b'), c(a \otimes c'))} \\
&= \phi(a \otimes (b, c))^{(bc(a \otimes b'), bc(a \otimes c'))} \\
&= \phi(a \otimes (b, c))\phi^{(bc a \otimes ({}^{bc} b', {}^{bc} c'))} \\
&= \phi(a \otimes (b, c))\phi^{(b, c)}(a \otimes (b', c')).
\end{aligned}$$

Therefore ϕ extends to a homomorphism $\phi^* : A \otimes (B \times C) \rightarrow (A \otimes B) \times (A \otimes C)$.

Consider the two maps

$$\begin{aligned}
\psi_1 : A \otimes B &\rightarrow A \otimes (B \times C) & \psi_2 : A \otimes C &\rightarrow A \otimes (B \times C) \\
(a \otimes b) &\mapsto a \otimes (b, 1_C) & (a \otimes c) &\mapsto a \otimes (1_B, c).
\end{aligned}$$

Now,

$$\begin{aligned}
\psi_1(aa' \otimes b) &= aa' \otimes (b, 1_C) \\
&= ({}^a a' \otimes {}^a (b, 1_C))(a \otimes (b, 1_C)) \\
&= \psi_1({}^a a' \otimes {}^a b)\psi_1(a \otimes b) \\
&= \psi_1({}^a (a' \otimes b))\psi_1(a \otimes b),
\end{aligned}$$

and,

$$\psi_1(a \otimes bb') = a \otimes (bb', 1_C)$$

$$\begin{aligned}
&= a \otimes (b, 1_C)(b', 1_C) \\
&= (a \otimes (b, 1_C))^{(b, 1_C)}(a \otimes (b', 1_C)) \\
&= (a \otimes (b, 1_C))({}^b a \otimes {}^b(b', 1_C)) \\
&= \psi_1(a \otimes b)\psi_1({}^b a \otimes {}^b b') \\
&= \psi_1(a \otimes b)\psi_1({}^b(a \otimes b')).
\end{aligned}$$

So ψ_1 is a crossed pairing and extends to a homomorphism $\psi_1^* : A \otimes B \rightarrow A \otimes (B \times C)$ ■

Let K be a normal subgroup of both G and H . The crossed pairing $K \times H \rightarrow G \otimes H$, $(k, h) \mapsto k \otimes h$ induces a homomorphism $\iota_1 : K \otimes H \rightarrow G \otimes H$. Equation 1.5 ensures that the image of ι_1 is a normal subgroup of $G \otimes H$. Let K_1 denote this image. In a similar manner we can produce a homomorphism $\iota_2 : G \otimes K \rightarrow G \otimes H$, and we denote its normal image by K_2 .

1.2.21 Proposition [21]

$$\text{i) } G/K \otimes H/K \cong G \otimes H/K_1 K_2$$

$$\text{ii) } G/K \wedge H/K \cong G \wedge H/K_1 K_2$$

Proof: We define the map $\phi : G/K \times H/K \rightarrow G \otimes H/K_1 K_2$ by $(gK, hK) \mapsto g \otimes h/K_1 K_2$ for $g \in G$ and $h \in H$. This map is well defined, for if $g' = gk_1$ and $h' = hk_2$ for some $k_1, k_2 \in K$, then

$$\begin{aligned}
g' \otimes h' K_1 K_2 &= gk_1 \otimes hk_2 K_1 K_2 \\
&= {}^g(k_1 \otimes hk_2)(g \otimes h)^h(g \otimes k_2)K_1 K_2
\end{aligned}$$

$$= (g \otimes h)K_1K_2.$$

The normality of K_1K_2 in $G \otimes H$ together with the relations of the tensor product in that group imply that ϕ is a crossed pairing and so there is a homomorphism $\hat{\phi} : G/K \otimes H/K \rightarrow G \otimes H/K_1K_2$. Now if $x \in \ker \hat{\phi}$ with $x = (g_1K \otimes h_1K)(g_2K \otimes h_2K) \dots (g_nK \otimes h_nK)$ then considered as an element of $G \otimes H$, $(g_1 \otimes h_1)(g_2 \otimes h_2) \dots (g_n \otimes h_n) \in K_1K_2$. Then this element can be rewritten as a product of elements of the form $g \otimes h$ where either $g \in K$ or $h \in K$. In either case $gK \otimes hK$ is the identity of $G/K \otimes H/K$ and $\ker \hat{\phi} = 1$. This establishes (i). The proof of (ii) is similar. ■

1.2.22 Theorem [47][9] *Let F be a free group. Then $F \wedge F \cong [F, F]$, $x \wedge y \mapsto [x, y]$.*

Proof: This algebraic proof is based on [47]. An alternative topological proof can be found in [9].

Suppose firstly that F is finitely generated. We use induction on n the minimum number of generators of F . For $n = 1$ as in the example above we see that $F \wedge F = 1 = [F, F]$. Suppose $n > 1$ and F is generated by the elements f_1, f_2, \dots, f_n . Let F_1 denote the subgroup of F generated by f_1 and F_2 denote the subgroup generated by $f_2, f_3 \dots f_n$. Then F_1 and F_2 are free and Proposition 1.2.11 provides homomorphisms $\lambda_k : F_k \wedge F_k \rightarrow [F_k, F_k]$, $k = 1, 2$, and $\lambda : F \wedge F \rightarrow [F, F]$. Now Corollary 1.6 of [28] states that if P_1, \dots, P_n are groups then there is an isomorphism

$$J_2(P_1 * P_2 * \dots * P_n) \cong J_2(P_1) \times J_2(P_2) \times \dots \times J_2(P_n) \times \prod_{1 \leq i < j \leq n} P_i^{ab} \otimes P_j^{ab},$$

where $J_2(G) = \ker(G \otimes G \rightarrow G)$, from which it follows that $\ker \lambda = \ker \lambda_1 \times$

$\ker \lambda_1$. The induction hypothesis now implies that $\ker \lambda_1 = 1 = \ker \lambda_2$.

To deal with the general case let F be a free group and $\lambda : F \rightarrow F$ be the homomorphism provided by Proposition 1.2.11. Suppose $g \in \ker \lambda$, $g = (a_1 \wedge b_1)(a_2 \wedge b_2) \dots (a_m \wedge b_m)$, where $a_i, b_i \in F$, $1 \leq i \leq m$. Let E be the subgroup of F generated by $a_1, b_1, a_2, b_2 \dots a_m, b_m$. The map $E \wedge E \rightarrow F \wedge F$ together with the result in the finitely generated case produces the commutative diagram

$$\begin{array}{ccccc} E \wedge E & \longrightarrow & F \wedge F & \xrightarrow{\lambda} & F \\ & \searrow \phi & & \nearrow & \\ & & E & & \end{array}$$

Now $g = (a_1 \wedge b_1)(a_2 \wedge b_2) \dots (a_m \wedge b_m)$ may be regarded as an element of $E \wedge E$ and by the injectivity of ϕ , $g = 1$. ■

Recall from for instance [44] that Hopf's formula for the second integral homology of a group G is

$$H_2(G) \cong \frac{R \cap [F, F]}{[R, F]}$$

where F/R is any free presentation of G . Thus we have an immediate corollary to Theorem 1.2.20.

1.2.23 Corollary [9] *For any group G there is an isomorphism*

$$H_2(G) \cong \ker(G \wedge G \longrightarrow G).$$

Proof: Let F/R be a free presentation of G and let ι and ν be the canonical homomorphisms $\iota : R \wedge F \longrightarrow F \wedge F$ and $\nu : F \wedge R \longrightarrow F \wedge F$. In view of Proposition 1.2.15, we note that $\iota(R \wedge F) = \nu(F \wedge R)$. By Proposition 1.2.19 (ii)

we have

$$F/R \wedge F/R \cong F \wedge F / \iota(R \wedge F) \nu(F \wedge R) \cong F \wedge F / \iota(R \wedge F)$$

and so $G \wedge G \cong F \wedge F / \iota(R \wedge F)$.

By Theorem 1.2.20 there is an isomorphism $F \wedge F \cong [F, F]$ with $x \wedge y \mapsto [x, y]$

which induces the isomorphism

$$G \wedge G \cong \frac{F \wedge F}{\iota(R \wedge F)} \cong \frac{[F, F]}{[R, F]}.$$

It follows that $\ker(G \wedge G \xrightarrow{\delta} G) \cong \ker\left(\frac{[F, F]}{[R, F]} \xrightarrow{\delta'} F/R\right)$. Now suppose we have $x = [f_1, f'_1][f_2, f'_2] \dots [f_n, f'_n][R, F]$ in $\ker\left(\frac{[F, F]}{[R, F]} \xrightarrow{\delta'} F/R\right)$. Since $\delta'(x) = 1$ we have $[f_1, f'_1][f_2, f'_2] \dots [f_n, f'_n] \in R$ and so $x \in \frac{R \cap [F, F]}{[R, F]}$. Furthermore since $\frac{R \cap [F, F]}{[R, F]} \leq \ker \delta'$ the results follows. ■

1.3 The q -tensor product

Let q be a non-negative integer. We recall here the definition and some results concerning the q -tensor product of crossed modules.

Let $\mu : G \rightarrow L$ and $\nu : H \rightarrow L$ be two crossed L -modules and let $K = G \times_L H = \{(g, h) | g \in G, h \in H, \mu(g) = \nu(h)\}$. Then we have the following diagram,

$$\begin{array}{ccc} G \times_L H & \xrightarrow{\pi_2} & H \\ \pi_1 \downarrow & & \downarrow \nu \\ G & \xrightarrow{\mu} & L \end{array}$$

where π_1 and π_2 are the projections $\pi_1(g, h) \mapsto g$, $\pi_2(g, h) \mapsto h$. We can define an

action of G on H by utilizing the action of L on H , i.e., ${}^g h = \mu^{(g)}h$. The group G acts on $G \times_L H$ by ${}^{g'}(g, h) = ({}^{g'}g, {}^{g'}h)$. Similar actions of H on G and $G \times_L H$ can be defined.

1.3.1 Definition [13] For $q \geq 1$ we define the q -tensor product of the crossed L -modules μ and ν to be the group $G \otimes^q H$ generated by the symbols $g \otimes^q h$ and $\{m\}$ with $g \in G$, $h \in H$ and $m \in K$, subject to the relations

$$gg' \otimes^q h = ({}^g g' \otimes^q {}^g h)(g \otimes^q h), \quad (1.11)$$

$$g \otimes^q hh' = (g \otimes^q h)({}^h g \otimes^q {}^h h'), \quad (1.12)$$

$$\{m\}(g \otimes^q h)\{m\}^{-1} = m^q g \otimes^q m^q h, \quad (1.13)$$

$$\{mm'\} = \{m\} \prod_{i=1}^{q-1} (m^{-1} \otimes^q (m^{1-q+i} m')^i) \{m'\}, \quad (1.14)$$

$$[\{m\}, \{m'\}] = m^q \otimes^q m'^q, \quad (1.15)$$

$$\{[m, m']\} = (m \otimes^q m')^q, \quad (1.16)$$

for all $g, g' \in G$, $h, h' \in H$ and all $m, m' \in K$.

For $q = 0$ we define the q -tensor product, $G \otimes^0 H$ to be the group generated by the symbols $g \otimes^0 h$ and subject to the relations (1.11) and (1.12) only. We usually write $G \otimes H$ and $g \otimes h$ in this case.

1.3.2 Remarks (a) Just as the motivation for the relations (1.1) and (1.2) is the commutator relations in groups, the identities (1.11) ... (1.16) can be viewed as abstractions of the identities occurring between commutators and powers in groups.

(b) When G and H are normal subgroups of the group L , then the inclusions

$G \hookrightarrow L$, and $H \hookrightarrow L$, are natural crossed L -modules with the actions taken to be conjugation in L . In this case $K = G \times_L H \cong G \cap H$.

(c) There is a natural homomorphism $G \otimes H \longrightarrow G \otimes^q H$ of the nonabelian tensor product into the q -tensor product, namely the map defined on generators by $g \otimes h \mapsto g \otimes^q h$.

1.3.3 Definition The q -exterior product of two crossed L -modules $\mu : G \rightarrow L$ and $\nu : H \rightarrow L$ is the quotient of $G \otimes^q H$ factored by the normal subgroup generated by $\pi_1 k \otimes^q \pi_2 k$ for all $k \in G \times_L H$. We denote this group by $G \wedge^q H$ and the coset represented by $g \otimes^q h$ by $g \wedge^q h$.

1.3.4 Lemma [13] *There are homomorphisms*

$$\lambda : G \otimes^q H \rightarrow G,$$

$$\lambda' : G \otimes^q H \rightarrow H$$

such that

$$\lambda(g \otimes^q h) = g^h m^{-1}, \quad \lambda(\{k\}) = \pi_1 k^q,$$

$$\lambda'(g \otimes^q h) = {}^g h h^{-1}, \quad \lambda'(\{k\}) = \pi_2 k^q,$$

for all $g \in G$, $h \in H$ and $k \in K$.

Proof: The proof here is an extension of that of Proposition 1.1.8. The chief idea is that the relations (1.13) ... (1.16) model the relationships between powers and commutators in groups. ■

As in Proposition 1.1.8 the homomorphisms λ and λ' are crossed modules, but in order to see this we require the following lemma.

1.3.5 Lemma [13] For $g \in G$, $h \in H$ and $k \in K$ we have

$$\begin{aligned} {}^g\{k\} &= (g \otimes^q \pi_2 k^1)\{k\}, \\ {}^h\{k\} &= \{k\}(\pi_1 k^{-q} \otimes^q h). \end{aligned}$$

Proof: We prove the first of these equations, the proof of the second being similar. Firstly define the homomorphism $\delta : G \otimes^q H \rightarrow G \times_L H$ by $\delta(x) = (\lambda(x), \lambda'(x))$.

Now the equation given in Proposition 1.2.11 (c) above holds for the nonabelian tensor product and so holds for λ restricted to the subgroup of $G \otimes^q H$ generated by all $g \otimes^q h$. So for x in this subgroup and any integers i and t we have

$$\begin{aligned} \lambda x^{-1} \otimes^q (x^{1-t+i} h^i) &= x^{-1} (x^{1-t+i} h^i x) \\ &= x^{-t+i} (h^i x) x^{-1+t-i} \end{aligned}$$

Let $k' = {}^m k k^{-1}$ so that ${}^g\{k\} = k' k$. Then $\pi_1 k' = g \pi_1 k g^{-1} \pi_1 k^{-1} = g^k g^{-1} = g^{\pi_2 k} g^{-1}$ and $\pi_2 k' = {}^g \pi_2 k \pi_2 k^{-1}$ so that $k' = \delta(g \otimes^q \pi_2 k)$. We now have

$$\begin{aligned} \{k' k\} &= \{k'\} \prod_{i=1}^{q-1} (\pi_1 k'^{-1} \otimes^q (k'^{1-t+i} \pi_2 k)^i) \{k\} \\ &= \{g k k^{-1}\} (g \otimes^q \pi_2 k)^{-q+1} \left(\prod_{i=1}^{q-1} (k^i (g \otimes^q \pi_2 k)) \right) \{k\} \\ &= (g \otimes^q \pi_2 k)^q (g \otimes^q \pi_2 k)^{-q-1} \left(\prod_{i=1}^{q-1} (k^i (g \otimes^q \pi_2 k)) \right) \{k\} \quad \text{using (1.16)} \\ &= \left(\prod_{i=0}^{q-1} (k^i (g \otimes^q \pi_2 k)) \right) \{k\} \\ &= (g \otimes^q \pi_2 k^q) \{k\}, \end{aligned}$$

where on the last step we apply (1.11) $q - 1$ times. ■

1.3.6 Proposition [13] *The homomorphisms*

$$\lambda : G \otimes^q H \rightarrow G,$$

$$\lambda' : G \otimes^q H \rightarrow H$$

are crossed modules.

Proof: We need to check the crossed module identities (1.9) and (1.10). Working with generators we have for the first of these

$$\begin{aligned} \lambda(g'(g \otimes^q h)) &= \lambda(g'g \otimes^q g'h) \\ &= g'g^{g'h}(g'g)^{-1} \\ &= g\lambda(g \otimes^q h)g^{-1}; \end{aligned}$$

$$\begin{aligned} \lambda(g'\{k\}) &= \lambda(\{g'k\}) \\ &= g'(\pi_1 k^q) \\ &= g'\lambda(\{k\})g'^{-1}. \end{aligned}$$

The identities $\lambda^{(g \otimes^q h)}(g' \otimes^q h') = (g \otimes^q h)(g' \otimes^q h')(g \otimes^q h)^{-1}$ and $\lambda^{\lambda(\{k\})}(g' \otimes^q h') = \pi_1 k^q (g' \otimes^q h') = \{k\}(g' \otimes^q h')\{k'\}$ follow from (1.5) and (1.13) respectively. The remaining identities require the preceding lemma.

$$\begin{aligned} \lambda^{(g \otimes^q h)}\{k\} &= g^h g^{-1}\{k\} \\ &= (g^h g^{-1} \otimes^q \pi_2 k^q)\{k\} \\ &= (g \otimes^q h)^{\pi_2 k^q} (g \otimes^q h)^{-1}\{k\} \\ &= (g \otimes^q h)\{k\}(g \otimes^q h)^{-1} \end{aligned}$$

using (1.5) and (1.6).

$$\begin{aligned}
\lambda(\{k\})\{k'\} &= \pi_1 k^q \{k'\} \\
&= (\pi_1 k^q \otimes^1 \pi_2 k'^q)\{k'\} \\
&= [\{k\}, \{k'\}]\{k'\} \\
&= \{k\}\{k'\}\{k\}^{-1}
\end{aligned}$$

■

Let M be a normal subgroup of $G \times_L H$, ι_1 be the canonical homomorphism $\iota_1 : \pi_1 M \otimes^q H \rightarrow G \otimes^q H$ and ι_2 be the cononical homomorphism $\iota_2 : G \otimes^q \pi_2 M \rightarrow G \otimes^q H$. Let M_1 and M_2 denote the images of ι_1 and ι_2 respectively. The subgroups M_1 and M_2 are normal for the preceding proposition provides crossed modules $\mu : \pi_1 M \otimes^q H \rightarrow H$ and $\nu : G \otimes^q H \rightarrow H$ and since ι_1 is a morphism of crossed modules. It follows that ι_1 is a crossed module and so its image is normal. The following analogue to Proposition 1.2.19 is new.

1.3.7 Proposition

- i) $G/\pi_1 M \otimes^q H/\pi_2 M \cong G \otimes^q H/M_1 M_2$
- ii) $G/\pi_1 M \wedge^q H/\pi_2 M \cong G \wedge^q H/M_1 M_2$

Proof: As with Proposition 1.2.19 we can show that the map $\phi : G/\pi_1 M \otimes^q H/\pi_2 M \rightarrow G \otimes^q H$ defined on generators by $g\pi_1 M \otimes^q h\pi_2 M \mapsto g \otimes^q hM_1 M_2$, $\{(g\pi_1 M, h\pi_2 M)\} \mapsto \{(g, h)\}M_1 M_2$ is a well-defined homomorphism with trivial kernel. The normality of $M_1 M_2$ follows from the fact that $\iota : \pi_1 M \otimes^q H \rightarrow G \otimes^q H$ is a morphism of the crossed modules $\mu : \pi_1 M \otimes^q H \rightarrow H$ and $\lambda : G \otimes^q H \rightarrow H$

$$\begin{array}{ccc}
\pi_1 M \otimes^q H & \xrightarrow{\iota} & G \otimes^q H \\
& \searrow \mu & \downarrow \lambda \\
& & H
\end{array}$$

and so is itself a crossed module. The image of a crossed module being normal it follows that $M_1 M_2$ is normal in $G \otimes^q H$. ■

1.3.8 Proposition [22][26] *If F is a free group then $F \wedge^q F \cong [F, F]F^q$.*

Proof: By Theorem 1.2.20 above there exists an isomorphism $\alpha : F \wedge F \rightarrow [F, F]$ and by the previous proposition a crossed module $\beta : F \wedge^q F \rightarrow [F, F]F^q$ so that we have the commutative diagram

$$\begin{array}{ccccc}
F \wedge F & \longrightarrow & F \wedge^q F & \longrightarrow & F \wedge^q F / \text{Im}(\iota) \\
\alpha \downarrow & & \beta \downarrow & & \downarrow \gamma \\
[F, F] & \longrightarrow & [F, F]F^q & \longrightarrow & (F^{ab})^q
\end{array}$$

where γ is induced by β and ι is the canonical homomorphism $\iota : F \wedge F \rightarrow F \wedge^q F$. Now $(F^{ab})^q$ is a free abelian group, $F \wedge^q F / \text{Im}(\iota)$ is abelian because of (1.15) and γ is surjective. It follows that γ has a splitting and because of (1.14) this splitting is surjective. Therefore γ is an isomorphism and since the rows are both short exact sequences, β is an isomorphism. ■

1.3.9 Proposition [22][26] *Let F/R be a free presentation for the group G . Then $[F, F]F^q / [F, R]R^q \cong G \wedge^q G$.*

Proof: Firstly note that because of Proposition 1.3.8, under the canonical

homomorphisms $\iota_1 : F \wedge^q R \rightarrow F \wedge^q F$ and $\iota_2 : R \wedge^q F \rightarrow F \wedge^q F$, $Image(\iota_1) = Image(\iota_2)$. By Proposition 1.3.7, $F \wedge^q F / F \wedge^q R = F \wedge^q F / Image(\iota_1) Image(\iota_2) = F / R \wedge^q F / R = G \wedge^q G$. The last proposition then provides the following diagram

$$\begin{array}{ccc}
 F \wedge^q R & \xrightarrow{\phi} & [F, R]R^q \\
 \downarrow & & \downarrow \\
 F \wedge^q F & \xleftarrow{\cong} & [F, F]F^q \\
 \downarrow & & \downarrow \\
 \frac{F \wedge^q F}{F \wedge^q R} & \longrightarrow & \frac{[F, F]F^q}{[F, R]R^q}
 \end{array}$$

Since the mapping $[F, R]R^q \rightarrow [F, F]F^q$ is injective the map $F \wedge^q R \rightarrow F \wedge^q F$ is injective and so the kernels $[F, R]R^q$ and $F \wedge^q R$ are isomorphic. ■

Hopf's formula can be extended to a formula for the second homology of a group with coefficients in the trivial module \mathbb{Z}_q , namely

$$H_2(G, \mathbb{Z}_q) \cong \frac{R \cap [F, F]F^q}{[R, F]R^q},$$

where F/R is a free presentation of G . Then the following is an immediate corollary of Proposition 1.3.9.

1.3.10 Corollary *For any group G there is an isomorphism*

$$H_2(G, \mathbb{Z}_q) \cong \ker(G \wedge^q G \rightarrow G).$$

1.3.11 Proposition *Let $\delta : M \rightarrow N$ be a projective q -crossed N -module and let $R \rightarrow F \xrightarrow{\lambda} N$ be a free presentation of N . Then there is an isomorphism*

$$[M, M]M^q \xrightarrow{\cong} [F, F]F^q / F \#^q R$$

given by $[m, m'] \mapsto [f, f']F\#^q R$ and $m^q \mapsto f^q F\#^q R$, where $\delta m = \lambda f$, $\delta m' = \lambda f'$.

Proof: The map λ is not itself a q -crossed N -module, but since $F\#^q R$ is a normal subgroup of F and contained in R we can factor to produce the q -crossed N -module $\lambda' : F/F\#^q R \rightarrow N$. Using the fact that δ is projective we can complete the triangle in the following diagram.

$$\begin{array}{ccc} & & M \\ & \exists \mu \swarrow \cdots & \downarrow \delta \\ F & & N \\ \frac{F}{F\#^q R} & \xrightarrow{\lambda'} & \end{array}$$

Let $\hat{\mu}$ be the restriction of μ to $M\#^q M$. Since both δ and λ' are onto $\hat{\mu}$ is onto $F\#^q F/F\#^q R$.

Now F is free and so projective and $\delta : M \rightarrow N$ so we can complete the following triangle.

$$\begin{array}{ccc} & & F \\ & \exists \phi \swarrow \cdots & \downarrow \lambda \\ M & \xrightarrow{\delta} & N \end{array}$$

Furthermore if $r \in R$, then $\phi(r)m\phi(r)^{-1} = {}^{\delta\phi(r)}m = m$ for all $m \in M$ and so $[F, R] \leq \ker \phi$. Similarly, $\lambda(r) = 1$ so $\phi(r) \in \ker \delta$ so $\phi(r)^q = 1$ and $r^q \in \ker \phi$. It follows that $F\#^q R \leq \ker \phi$ and so we can factor ϕ to form $\hat{\phi} : F/F\#^q R \rightarrow M$. This map is onto $M\#^q M$ because $\hat{\mu}$ is onto $F\#^q F/F\#^q R$ and forms an inverse to $\hat{\mu}$.

$$\begin{array}{ccc} N & \longleftarrow & M\#^q M \\ \uparrow & \swarrow \cdots & \downarrow \\ \frac{F\#^q F}{F\#^q R} & \longrightarrow & N \end{array}$$

■

Chapter 2

The computation of the nonabelian tensor product of finite groups

2.1 Introduction

For finite groups G and H the nonabelian tensor product of groups is finite [25] and may thus in principle be calculated directly from its defining presentation. However even for groups of moderate size the number of generators and relations can be prohibitively large. For instance the 2-generator Burnside group $B(2, 4)$ of exponent 4 would involve 2^{24} generators and 2^{37} relations. Problems when dealing with such large presentations remain even when presentation reduction algorithms such as the Tietze transformations are applied.

In this chapter we fine tune an alternative algorithm given in [27] for calculating the nonabelian tensor and exterior product of groups. The alternative method, when applied to $B(2, 4)$, requires only 4 generators and as few as 126 relators.

We implement this algorithm in GAP [54] in the case where G and H are prime-power groups and used to calculate the nonabelian tensor product for all pairs of groups in the normal subgroup lattice of Q_{32} , the Quaternion group of order 32. As a second example we calculate the nonabelian tensor product under all actions between Q_8 the Quaternion group of order 8 and D_4 the Dihedral group of order 8.

Appendix A contains the GAP functions to compute the tensor product and the q -tensor product of two groups. It also contains allied functions required to implement the action of one group on another. Similar functions which compute the exterior product, the tensor and exterior square, and the “ q ” versions of these are available on the diskette accompanying this thesis.

Appendix B contains the GAP programs on which the tables and some of the other calculations are based. These programs are referenced when results from their outputs are mentioned. For example (App B, actions.g) refers to the program actions.g in Appendix B.

2.2 Theoretical background

Let G and H be groups with compatible actions and let $G * H$ denote their free product. The conjugation action $\rho : G \rightarrow \text{Aut}(G)$, $g \mapsto g(\)g^{-1}$ and the action $\kappa : H \rightarrow \text{Aut}(G)$ combine to form an action $\rho * \kappa : G * H \rightarrow \text{Aut}(G)$ of $G * H$ on G . For $z \in G * H$ and $g \in G$ we write ${}^z g = \rho * \kappa(z).g$. Similarly the conjugation action $H \rightarrow \text{Aut}(H)$ and the action $G \rightarrow \text{Aut}(H)$ combine to form an action

$(z, h) \mapsto {}^z h$ of $G * H$ on H . We let J denote the subgroup of $G * H$ normally generated by the elements

$$W(z, g, h) = z[g, h]z^{-1}[{}^z g, {}^z h]^{-1}$$

for $z \in G * H$, $g \in G$ and $h \in H$. We note that in this instance ${}^z g \in G$ and ${}^z h \in H$ denote the action of $G * H$ on G and H respectively and not conjugation in the free product $G * H$.

There are natural inclusions of G and H into $G * H$ and it is shown in [27] that the composite maps $G \rightarrow G * H/J$ and $H \rightarrow G * H/J$ are injective. For this reason we can identify G and H with their images in $G * H/J$.

The algorithm presented here is based on the following results.

2.2.1 Theorem [29][27] *The inclusions $G \hookrightarrow G * H/J$, $H \hookrightarrow G * H/J$ induce an isomorphism*

$$((G \otimes H) \rtimes H) \rtimes G \cong G * H/J$$

where \rtimes denotes the semi-direct product with respect to the actions of H on $G \otimes H$ and of G on $(G \otimes H) \rtimes H$, the latter being given by ${}^g(\tau, h) = ({}^g\tau(g \otimes h), h)$ for $\tau \in G \otimes H$, $g \in G$ and $h \in H$.

2.2.2 Corollary [29][27] *The above isomorphism restricts to an isomorphism*

$$G \otimes H \cong \bar{G} \cap \bar{H}$$

where \bar{G} and \bar{H} are the normal closures of G and H in $G * H/J$.

The efficiency of the algorithm depends on finding a small number of normal

generators for the subgroup J (cf. Theorem 2.2.8). To this end let \mathfrak{G}_G be a generating set for G and \mathfrak{G}_H a generating set for H . Let ${}_H G$ denote those elements of the centre of G which act trivially on H and likewise let ${}_G H$ denote those elements of the centre of H which act trivially on G . Let $T_{{}_H G}$ denote the union of a choice of generating set for ${}_H G$ with a transversal for ${}_H G$ in G . Similarly, let $T_{{}_G H}$ denote the union of a choice of generating set for ${}_G H$ with a transversal for ${}_G H$ in H . It is claimed in [27] that

2.2.3 Theorem [27] *The subgroup J is normally generated by those elements of the form $W(z, g, h)$ where $z \in T_{{}_H G} \cup T_{{}_G H}$, $g \in \mathfrak{G}_G$ and $h \in \mathfrak{G}_H$.*

This result is not proved in [27]. We supply a proof of a slightly more general result (namely Theorem 2.2.8) here. Theorem 2.2.3 depends on the following crucial lemma whose statement and proof was omitted in [27].

2.2.4 Lemma *Let $J \trianglelefteq G * H$ be normally generated by all $W(z, a, b)$ where $z \in G * H$, $a \in \mathfrak{G}_G$ and $b \in \mathfrak{G}_H$ and let $K \trianglelefteq G * H$ be normally generated by all $W(x, a, b)$ where $x \in G \cup H$, $a \in \mathfrak{G}_G$ and $b \in \mathfrak{G}_H$. Then $J = K$.*

Proof: Since $K \leq J$ we need only show that each generator of J is in K . Let $z \in G * H$. We use induction on $|z|$ the length of z written as a word in the generating sets \mathfrak{G}_G and \mathfrak{G}_H . If $|z| = 1$ then $z \in G$ or $z \in H$ and $W(z, a, b) \in K$. Suppose now $z = xy$ where $x \in \mathfrak{G}_G$ or $x \in \mathfrak{G}_H$ and $|y| < |z|$. Then working mod K we have

$$W(xy, a, b) = xy[a, b](xy)^{-1} [{}^{xy}a, {}^{xy}b]^{-1}$$

$$\begin{aligned}
&= xy[a, b](xy)^{-1} y[y^{-1}xy a, y^{-1}xy b]^{-1} \\
&= xy[a, b](xy)^{-1} y[y^{-1}x a, y^{-1}x b]^{-1} \\
&= xy[a, b](xy)^{-1} y \left(y^{-1}x[a, b]^{-1}y^{-1}x^{-1}W(y^{-1}x, a, b) \right) \\
&= xy[a, b](xy)^{-1} y \left(y^{-1}x[a, b]^{-1}y^{-1}x^{-1} \right) \\
&= xy[a, b]y^{-1}x^{-1}x[y a, y b]^{-1}x^{-1} \\
&= xy[a, b]y^{-1}[y a, y b]x^{-1} \\
&= xW(y, a, b)x^{-1} \\
&= 1
\end{aligned}$$

mod K by the induction hypothesis. ■

We now show that we can reduce the generating set for J further than in Lemma 2.2.4. In order to do this we need the following.

2.2.5 Lemma *For all $x, y \in G * H$, $g \in G$, $h \in H$,*

$$W(xy, g, h) = xW(y, g, h)x^{-1}W(x, {}^y g, {}^y h)$$

Proof:

$$\begin{aligned}
W(xy, g, h) &= xy[g, h](xy)^{-1}[{}^{xy}g, {}^{xy}h]^{-1} \\
&= xW(y, g, h)[{}^y g, {}^y h]x^{-1}[{}^{xy}g, {}^{xy}h]^{-1} \\
&= xW(y, g, h)x^{-1}x[{}^y g, {}^y h]x^{-1}x[{}^{xy}g, {}^{xy}h]^{-1} \\
&= xW(y, g, h)x^{-1}W(x, {}^y g, {}^y h)
\end{aligned}$$
■

2.2.6 Lemma *Suppose $K \trianglelefteq G * H$, $g \in G$ and $h \in H$, and if $s, t \in G$ or $s, t \in H$,*

$W(t, g, h) = 1 \pmod K$ and $W([t^{-1}, s], g, h) = 1 \pmod K$, then $W({}^s t, g, h) = 1 \pmod K$.

Proof:

$$\begin{aligned}
W({}^s t, g, h) &= {}^s t[g, h]{}^s t^{-1} [{}^s t g, {}^s t h]^{-1} \\
&= {}^s t[g, h]{}^s t^{-1} [{}^{sts^{-1}} g, {}^{sts^{-1}} h] \\
&= {}^s t[g, h]{}^s t^{-1} {}^t [{}^{t^{-1}sts^{-1}} g, {}^{t^{-1}sts^{-1}} h]^{-1} \\
&= {}^s t[g, h]{}^s t^{-1} {}^t \left([{}^{t^{-1}, s}][g, h]^{-1} [{}^{t^{-1}, s}]^{-1} W([{}^{t^{-1}, s}], g, h) \right) \\
&= {}^s t[g, h]{}^s t^{-1} s^{-1} {}^t [{}^{t^{-1}, s}] {}^t [g, h]^{-1} {}^t [{}^{t^{-1}, s}]^{-1} \\
&= {}^s t[g, h]{}^s t^{-1} s^{-1} t t^{-1} s t s^{-1} t^{-1} [{}^t g, {}^t h]^{-1} t \left(t^{-1} s t s^{-1} \right)^{-1} t^{-1} \\
&= [s, t] t [g, h] t^{-1} [{}^t g, {}^t h]^{-1} t s t^{-1} s^{-1} \\
&= [s, t] W(t, g, h) [s, t]^{-1}
\end{aligned}$$

■

2.2.7 Lemma *Suppose $M \trianglelefteq L \leq G$ and \mathcal{T}_M is a (left) transversal for M in L . Let $R \trianglelefteq G * H$ be normally generated by all elements of the form $W(x, a, b)$ where $x \in L$, $a \in \mathcal{G}_G$ and $b \in \mathcal{G}_H$. Let $S \trianglelefteq G * H$ be normally generated by all elements of the form $W(x, a, b)$ where $x \in M \cup \mathcal{T}_M$, $a \in \mathcal{G}_G$ and $b \in \mathcal{G}_H$. Then $R = S$.*

Proof: To begin with $S \leq R$ so we need only show that all generators of R are in S . To this end let $g \in L$. Then $g = tm$ for some $t \in \mathcal{T}_M$ and $m \in M$. Since $M \trianglelefteq L$, we have $[t^{-1}, m^{-1}] \in M$ and so by Lemma 2.2.6 $W({}^{m^{-1}} t, a, b) = 1 \pmod S$. Now starting with Lemma 2.2.5 and working mod S we have

$$W(tm, a, b) = {}^t W(m, a, b) t^{-1} W(t, {}^m a, {}^m b)$$

$$\begin{aligned}
&= W(t, {}^m a, {}^m b) \\
&= t[{}^m a, {}^m b]t^{-1}[{}^{tm} a, {}^{tm} b]^{-1} \\
&= t[{}^m a, {}^m b]t^{-1} m[{}^{m^{-1}tm} a, {}^{m^{-1}tm} b]^{-1} \\
&= t[{}^m a, {}^m b]t^{-1} m[{}^{m^{-1}t} a, {}^{m^{-1}t} b]^{-1} \\
&= t[{}^m a, {}^m b]t^{-1} m \left(m^{-1}t[a, b]^{-1}m^{-1}t^{-1}W(m^{-1}t, a, b) \right) \\
&= t[{}^m a, {}^m b]t^{-1} m \left(m^{-1}t[a, b]^{-1}m^{-1}t^{-1} \right) \\
&= t[{}^m a, {}^m b]t^{-1}t[{}^m a, {}^m b]^{-1}t^{-1} \\
&= 1
\end{aligned}$$

■

Suppose now that J is the subgroup of $G * H$ normally generated by $W(x, a, b)$ where $x \in G \cup H$, $a \in \mathcal{G}_G$, $b \in \mathcal{G}_H$. Let $1 = G_0 \trianglelefteq G_1 \trianglelefteq \dots \trianglelefteq G_n$ be a subnormal series for G and $1 = H_0 \trianglelefteq H_1 \trianglelefteq \dots \trianglelefteq H_M$ be a subnormal series for H . Let \mathcal{T}_{G_k} denote a transversal for G_{k-1} in G_k and $\mathcal{H}_{G_k/G_{k-1}}$ a lift of a generating set for G_k/G_{k-1} to \mathcal{T}_{G_k} . Similarly \mathcal{T}_{H_k} denotes a transversal for H_{k-1} in H_k and $\mathcal{H}_{H_k/H_{k-1}}$ denotes a lift of a generating set for H_k/H_{k-1} to \mathcal{T}_{H_k} . We let

$$\mathcal{L}_{G_i} = \begin{cases} \mathcal{H}_{G_i/G_{i-1}}, & \text{if } G_i/G_{i-1} \text{ abelian} \\ \mathcal{T}_{G_i} & \text{otherwise} \end{cases}$$

$$\mathcal{L}_{H_i} = \begin{cases} \mathcal{H}_{H_i/H_{i-1}}, & \text{if } H_i/H_{i-1} \text{ abelian} \\ \mathcal{T}_{H_i} & \text{otherwise} \end{cases}$$

The following generalizes Theorem 2.2.3.

2.2.8 Theorem *Let K be the subgroup of $G * H$ normally generated by $W(x, a, b)$*

where x is in $(\bigcup_{i=1}^n \mathcal{L}_{G_i}) \cup (\bigcup_{j=1}^m \mathcal{L}_{H_j})$, $a \in \mathcal{G}_G$ and $b \in \mathcal{G}_H$. Then $J = K$.

Proof: By repeated applications of Lemma 2.2.7 we see that J is normally generated by $W(x, a, b)$ where $x \in \bigcup_{i=1}^n \mathcal{T}_{G_i} \bigcup_{j=1}^m \mathcal{T}_{H_j}$. It remains to show that we can replace \mathcal{T}_{G_i} by $\mathcal{H}_{G_i/G_{i-1}}$ in the case G_i/G_{i-1} is abelian and \mathcal{T}_{H_i} by $\mathcal{H}_{H_i/H_{i-1}}$ in the case H_i/H_{i-1} is abelian.

Suppose then that G_i/G_{i-1} is abelian and $R \trianglelefteq G * H$ is normally generated by $W(x, a, b)$ where $x \in \mathcal{T}_{G_i} \cup G_{i-1}$, $a \in \mathcal{G}_G$ and $b \in \mathcal{G}_H$ and $S \trianglelefteq G * H$ is normally generated by $W(x, a, b)$ where $x \in \mathcal{H}_{G_i/G_{i-1}} \cup G_{i-1}$, $a \in \mathcal{G}_G$ and $b \in \mathcal{G}_H$. Clearly $S \leq R$. We show that each generator of R is in S and so $R = S$.

Suppose $x \in \mathcal{T}_{G_i}$. Then x may be written in the form $s_1 s_2 \dots s_l y$ where $s_j \in \mathcal{H}_{G_i/G_{i-1}}$ for all $1 \leq j \leq l$ and $y \in G_{i-1}$. We define the length of x , $|x|$ to be the least l for which this is possible. If $|x| = 0$ then $x \in \mathcal{H}_{G_i/G_{i-1}}$ and $W(x, a, b) = 1 \pmod{S}$.

Suppose then that $W(y, a, b) = 1 \pmod{S}$ for all $|y| \leq |x|$. Let $x = sy$ where $s \in \mathcal{H}_{G_i/G_{i-1}}$ and $|y| \leq |x|$. Since G_i/G_{i-1} is abelian, $[s^{-1}, y^{-1}] \in G_{i-1}$ and

$$\begin{aligned}
W(sy, a, b) &= sy[a, b]y^{-1}s^{-1}[{}^{sy}a, {}^{sy}b]^{-1} \\
&= sy[a, b]y^{-1}s^{-1}y[{}^{y^{-1}s}a, {}^{y^{-1}s}b]^{-1} \\
&= sy[a, b]y^{-1}s^{-1}y\left(y^{-1}s[a, b]^{-1}y^{-1}s^{-1}W\left(y^{-1}s, a, b\right)\right) \\
&= sy[a, b]y^{-1}[{}^y a, {}^y b]^{-1}s^{-1} \\
&= sW(y, a, b)s^{-1} \\
&= 1
\end{aligned}$$

by the induction hypothesis. ■

2.3 Algorithms and programs

Figure 2.1 gives an algorithm for computing the nonabelian tensor product of two groups given finite presentations for the groups. The action of group G on H is given using an array whose $(i, j)^{th}$ entry is $g_i h_j$ where g_i is the i^{th} generator in the presentation of G and h_j is the j^{th} generator in the presentation of H . The action of H on G is similarly given. This algorithm is used in the case when the two groups are prime-powered groups; a power-commutator presentation is calculated for each group using the **ANUPQ** package supplied with **GAP** and their upper central series is used to calculate the words $W(x, a, b)$. This algorithm together with similar algorithms for computing the exterior product, and the tensor and exterior squares were implemented in **GAP** and form the basis of the functions listed in Appendix A.

Function:	TensorProduct																							
Dependencies:	<ul style="list-style-type: none"> • ANUPQ Package, function Pq • xtz, the Tietze wrapper function • GactsonH, implements action of one group on another 																							
Input:	<table border="1"> <thead> <tr> <th>Object</th> <th>Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>G, H</td> <td>finitely presented groups</td> <td>Groups for which $G \otimes H$ is to be computed. G is generated by g_1, g_2, \dots, g_n and H is generated by h_1, h_2, \dots, h_m.</td> </tr> <tr> <td><i>tietze</i></td> <td>logical</td> <td>If true indicates that Tietze transformations should be applied to $G * H/J$.</td> </tr> <tr> <td><i>prime</i></td> <td>integer</td> <td>Prime passed to ANU's Pq function</td> </tr> <tr> <td><i>bound</i></td> <td>integer</td> <td>Class bound passed to ANU's Pq function</td> </tr> <tr> <td><i>GactH</i></td> <td>array of group elements</td> <td>$n \times m$ array, the $(i, j)^{th}$ entry being $g_i h_j$. Used to compute action of G on H.</td> </tr> <tr> <td><i>HactG</i></td> <td>array of group elements</td> <td>$m \times n$ array, the $(i, j)^{th}$ entry being $h_i g_j$. Used to compute action of H on G.</td> </tr> </tbody> </table>	Object	Type	Notes	G, H	finitely presented groups	Groups for which $G \otimes H$ is to be computed. G is generated by g_1, g_2, \dots, g_n and H is generated by h_1, h_2, \dots, h_m .	<i>tietze</i>	logical	If true indicates that Tietze transformations should be applied to $G * H/J$.	<i>prime</i>	integer	Prime passed to ANU's Pq function	<i>bound</i>	integer	Class bound passed to ANU's Pq function	<i>GactH</i>	array of group elements	$n \times m$ array, the $(i, j)^{th}$ entry being $g_i h_j$. Used to compute action of G on H .	<i>HactG</i>	array of group elements	$m \times n$ array, the $(i, j)^{th}$ entry being $h_i g_j$. Used to compute action of H on G .		
Object	Type	Notes																						
G, H	finitely presented groups	Groups for which $G \otimes H$ is to be computed. G is generated by g_1, g_2, \dots, g_n and H is generated by h_1, h_2, \dots, h_m .																						
<i>tietze</i>	logical	If true indicates that Tietze transformations should be applied to $G * H/J$.																						
<i>prime</i>	integer	Prime passed to ANU's Pq function																						
<i>bound</i>	integer	Class bound passed to ANU's Pq function																						
<i>GactH</i>	array of group elements	$n \times m$ array, the $(i, j)^{th}$ entry being $g_i h_j$. Used to compute action of G on H .																						
<i>HactG</i>	array of group elements	$m \times n$ array, the $(i, j)^{th}$ entry being $h_i g_j$. Used to compute action of H on G .																						
Method:	<ul style="list-style-type: none"> • Use ANU's Pq function to compute power-commutator presentations for G and H. • Form the free group F on $n + m$ generators. • Compute the Upper Central Series of G and H and identify the generators of its members. Create lists $\bigcup \mathcal{L}_{G_i}$ and $\bigcup \mathcal{L}_{H_j}$. • Calculate a list of $W(x, a, b)$ as words in F. • If <i>tietze</i> = true apply Tietze transformation function <i>xtz</i> to presentation for F/J. • Compute the normal closures of the first n and last m generators of F/J and take their intersection A. 																							
Return:	A																							

Figure 2.1 Algorithm for computing $G \otimes H$

2.4 Examples

In this section we provide two examples of the use of the GAP functions mentioned above. In the first of these we compute the nonabelian tensor product and exterior product for every pair of normal subgroups of the quaternion group $Q_{32} = \langle a, b \mid a^{16} = b^8 a^2 = ab^{-1}ab = 1 \rangle$. The normal subgroup lattice for this group is given in Figure 2.2. Using conjugation in Q_{32} as the basis for all actions, Table 2.1 presents the different actions which arise between each pair of subgroups by exhibiting the images of the generators of H under the actions of the generators of G .

Table 2.2 (cf. App B, Q3.g) lists $G \otimes H$ for all pairs of normal subgroups G and H in Q_{32} together with the kernel of the homomorphism $G \otimes H \rightarrow G \cap H$. Table 2.3 (cf. App B, Q3w.g) lists the groups $G \wedge H$ and the kernel of the homomorphism $G \wedge H \rightarrow G \cap H$.

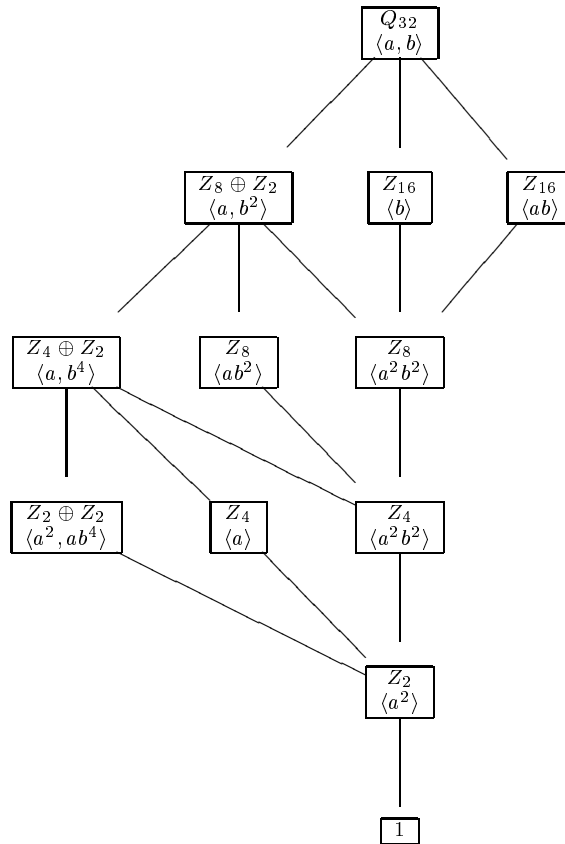


Figure 2.2: The normal subgroup lattice of Q_{32} .

Table 2.1: The Conjugation actions of subgroups G on subgroups H

		G															
		1	a^2	a	b^4	a^2	ab^4	a	b^4	a^2b^2	ab^2	a	b^2	b	ab	a	b
H	$x = a^2$	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	$x = a$	x	x	x	x	x	x	x	x	x	x	x	x	x^3	x^3	x	x^3
	$x = b^4$	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	$x = a^2$	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	$y = ab^4$	y	y	y	y	y	y	y	y	y	y	y	y	xy	xy	y	xy
	$x = a$	x	x	x	x	x	x	x	x	x	x	x	x	x^3	x^3	x	x^3
	$y = b^4$	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
	$x = a^2b^2$	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	$x = ab^2$	x	x	x	x	x	x	x	x	x	x	x	x	x^5	x^5	x	x^5
	$x = a$	x	x	x	x	x	x	x	x	x	x	x	x	x^3	x^3	x	x^3
	$y = b^2$	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
	$x = b$	x	x	x^9	x	x	x^9	x^9	x	x	x^9	x^9	x	x	x^9	x^9	x
$x = ab$	x	x	x^9	x	x	x^9	x^9	x	x	x^9	x^9	x	x^9	x	x^9	x^9	
$x = a$	x	x	x	x	x	x	x	x	x	x	x	x	x^3	x^3	x	x^3	
$y = b$	y	y	x^2y	y	y	x^2y	x^2y	y	y	x^2y	x^2y	y	y	x^2y	x^2y	y	

As a second example consider the Quaternion group $Q_8 = \langle x, y \mid x^4 = y^2x^2 = xy^{-1}xy = 1 \rangle$ and the Dihedral group $D_4 = \langle a, b \mid a^2 = b^4 = ba^{-1}ba = 1 \rangle$. Both groups are of order 8. The automorphism group of D_4 is isomorphic to D_4 itself while the automorphism group of Q_8 is the symmetric group S_4 (cf. App B, Auto.g). The program actions.g in Appendix B computes all the possible pairs of actions between D_4 and Q_8 by finding all possible images of their generators in the respective automorphism groups. The program compatible_actions.g tests each pair of actions to see if they are compatible. There are 28 possible actions of Q_8 on D_4 and 76 possible actions of D_4 on Q_8 giving altogether 2128 distinct pairs of actions between D_4 and Q_8 . Of these 292 are compatible pairs.

Rather than deal with all these actions we introduce the following new definition to provide an equivalence relation on the set of actions between pairs of groups.

2.4.1 Definition Let (G, H) be a pair of groups each acting on the other. Let (G', H') be another such pair. We say the pair of actions between G and H is *isomorphic* to the pair of actions between G' and H' if there exists isomorphisms $\alpha : G \rightarrow G'$ and $\beta : H \rightarrow H'$ such that ${}^{\alpha(g)}\beta(h) = \beta(gh)$ and ${}^{\beta(h)}\alpha(g) = \alpha(hg)$ for all $g \in G, h \in H$.

The following is an immediate consequence of this definition.

2.4.2 Corollary *Let the pair of actions between G and H be isomorphic to the pair of actions between G' and H' .*

- i) *If the actions between G and H are compatible, then the actions between G'*

and H' are compatible.

ii) $G \otimes H \cong G' \otimes H'$.

■

The 2128 actions between Q_8 and D_4 break into 109 equivalence classes (cf. App B, actions.g) of which 33 represent compatible actions (cf. App B, compatible_classes.g). The compatible actions are listed up to isomorphism in Table 2.4 together with the size of the equivalence class C they represent and the tensor product formed under this action (cf. App B, actions.g, TProduct_D4Q8.g).

We now turn our attention to the non-compatible pairs of actions between D_4 and Q_8 . There are two existing definitions for the nonabelian tensor product for groups with non-compatible actions. Firstly we may remove the compatibility restriction on the actions between G and H in Definition 1.1.2. We shall denote the tensor product in this case by $G \otimes_{BL} H$ where the BL stands for Brown and Loday. Although the tensor product $G \otimes_{BL} H$ is then defined it may lack certain properties present in the compatible case. For example with non-compatible actions it is not known if $G \otimes H$ is finite when G and H are finite. The second definition available to us was introduced by N. Inassaridze [37][38].

Suppose G and H are groups which act on each other, and on themselves by conjugation. The *nonabelian tensor product* $G \otimes_I H$ is the group generated by the symbols $g \otimes_I h$ where $g \in G$ and $h \in H$, subject to the relations

$$gg' \otimes_I h = ({}^g g' \otimes_I {}^g h)(g \otimes_I h) \quad (2.1)$$

$$g \otimes_I hh' = (g \otimes_I h)({}^h g \otimes_I {}^h h') \quad (2.2)$$

$$(g \otimes_I h)(g' \otimes_I h') = ([g, h]g' \otimes_I [g, h]h') (g \otimes_I h) \quad (2.3)$$

$$(g' \otimes_I h')(g \otimes_I h) = (g \otimes_I h) ([h, g]g' \otimes_I [h, g]h') \quad (2.4)$$

for $g, g' \in G, h, h' \in H$ and $[g, h] = ghg^{-1}h^{-1} \in G * H$.

We computed both $D_4 \otimes_{BL} Q_8$ (cf. App B, TProduct_D4Q8_BL2.g) and $D_4 \otimes_I Q_8$ (cf. App B, TProduct_D4Q8_In2.g) for each pair of isomorphic non-compatible actions between D_4 and Q_8 . The programs here take the presentations for D_4 and Q_8 as input and use the definitions of the two tensor products to produce presentations for $D_4 \otimes_{BL} Q_8$ and $D_4 \otimes_I Q_8$. For each pair of such actions we find $D_4 \otimes_{BL} Q_8 \cong D_4 \otimes_I Q_8$. Table 2.5 gives the frequency distribution of tensor products (both $D_4 \otimes_{BL} Q_8$ and $D_4 \otimes_I Q_8$ for distinct isomorphism classes of non-compatible actions and distinct non-compatible actions. In this table the number of isomorphism classes of non-compatible actions is denoted by N and the number of non-compatible actions by $(\sum |\mathcal{C}|)$.

We note that $\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$ predominates the list of nonabelian tensor products in the compatible cases while the nonabelian tensor product tends to be smaller in the non-compatible cases.

Table 2.4: The isomorphic classes of compatible actions between $G=D_4$ and $H=Q_8$ together with the size of class and the tensor product.

Action of H on G				Action of G on H				$ C $	$G \otimes H$
$^x a$	$^x b$	$^y a$	$^y b$	$^a x$	$^a y$	$^b x$	$^b y$		
a	b	a	b	x	y	x	y	1	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	b	x	y	x	$x^2 y$	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	b	x	$x^2 y$	x	y	3	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	b	x	$x^2 y$	x^3	y	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	aba	x	y	x	y	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	aba	x	y	x	$x^2 y$	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	aba	x	y	x^3	y	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	aba	x	$x^2 y$	x	y	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	aba	x	$x^2 y$	x	$x^2 y$	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	aba	x	$x^2 y$	x^3	y	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	aba	x^3	y	x	y	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	aba	x^3	y	x	xy	24	$\mathbb{Z}_2 \oplus \mathbb{Z}_4$
a	b	a	aba	x^3	y	x	$x^2 y$	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_4$
a	b	a	aba	x^3	y	x^3	y	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	a	aba	x^3	y	x^3	$x^2 y$	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	ab^2	b	x	y	x	y	3	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	ab^2	b	x	y	x	$x^2 y$	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	ab^2	b	x	y	x^3	y	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	ab^2	b	x	$x^2 y$	x	y	3	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	ab^2	b	x	$x^2 y$	x^3	y	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	ab^2	b	x^3	y	x	y	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	b	ab^2	b	x^3	y	x	$x^2 y$	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	aba	ab^2	b	x	y	x	y	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	aba	ab^2	b	x	y	x	$x^2 y$	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	aba	ab^2	b	x	y	x^3	y	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	aba	ab^2	b	x	y	x^3	$x^2 y$	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	aba	ab^2	b	x	$x^2 y$	x	y	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	aba	ab^2	b	x	$x^2 y$	x^3	y	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	aba	ab^2	b	x	$x^2 y$	x^3	$x^2 y$	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	aba	ab^2	b	x^3	y	x	y	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	aba	ab^2	b	x^3	y	x^3	$x^2 y$	12	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$
a	aba	ab^2	b	x^3	$x^2 y$	x^3	y	6	$\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_4$
ab	aba	ba	aba	y	x	x^3	$x^2 y$	12	\mathbb{Z}_8

Table 2.5: The isomorphic classes of compatible actions between $G=D_4$ and $H=Q_8$ together with the size of class and the tensor product.

$G \otimes H$	N	$\sum \mathcal{C} $
\mathbb{Z}_2	16	672
$\mathbb{Z}_2 \otimes \mathbb{Z}_2$	48	1008
\mathbb{Z}_4	3	36
$\mathbb{Z}_2 \otimes \mathbb{Z}_2 \otimes \mathbb{Z}_2$	3	48
$\mathbb{Z}_2 \otimes \mathbb{Z}_4$	6	72

Chapter 3

Tensor products of prime–power groups

3.1 Introduction

Throughout this chapter we let G be a finite p -group and H a finite q -group where the primes p and q are not necessarily distinct. We suppose G and H act on each other compatibly.

It was shown in [25] that $G \otimes H$ is a finite pq -group. In this chapter we give a bound for the order of $G \otimes H$. This bound is then compared with the order of $G \otimes H$ using the computer program of the last chapter for all normal subgroups G , H of the quaternion group of order 32. As a topological application we consider a union of CW -spaces $X = A \cup B$; we give a bound on $\pi_3 X$ in terms of the homotopy groups of A , B and $A \cap B$, and a certain tensor product.

3.2 Group-theoretic results

If there exists a group E containing both G and H as normal subgroups, then conjugation in E gives rise to compatible actions. In this case we let GH denote the subgroup of E generated by G and H , and we define

$${}_H G = \{g \in G : \text{there exists an } h \in H \text{ such that } gh^{-1} \in Z(GH)\}$$

Note that ${}_H G$ is a normal subgroup of G , and that ${}_H G = G$ if $G \subseteq H$.

We let ΦG denote the Frattini subgroup $[G, G]G^p$ of G . Thus G is a d -generator group if and only if $|G/\Phi G| = p^d$.

We will need the following lemmas.

3.2.1 Lemma *Suppose $N \leq [G, H]$. Then the image of ${}_H G \otimes N$ under the canonical mapping ${}_H G \otimes N \longrightarrow G \otimes H$ lies in the image of $N \otimes H$ under the canonical map $N \otimes H \longrightarrow G \otimes H$.*

Proof: Let $g \in {}_H G$ and $\tau = [g_1, h_1]^{\epsilon_1} [g_2, h_2]^{\epsilon_2} \dots [g_t, h_t]^{\epsilon_t} \in N$ where $\epsilon_i = \pm 1$. Let $\tilde{\tau} = (g_1 \otimes h_1)^{\epsilon_1} \dots (g_t \otimes h_t)^{\epsilon_t}$. Now by the definition of ${}_H G$ there exists a $h \in H$ such that ${}^g g_i = {}^h g_i$ and ${}^g h_i = {}^h h_i$ for all $1 \leq i \leq t$. It follows that ${}^g \tilde{\tau} = {}^h \tilde{\tau}$. But then by Proposition 1.2.11 we have

$$g \otimes \tau = {}^g \tilde{\tau} \tilde{\tau}^{-1} = {}^h \tilde{\tau} \tilde{\tau}^{-1} = (\tau \otimes h)^{-1}. \quad \blacksquare$$

Proposition 9 in [8] extends to yield the following.

3.2.2 Lemma *Any central subgroup Z of GH which lies in the intersection $G \cap H$*

gives rise to an exact sequence

$$(G \otimes Z) \times (Z \otimes H) \xrightarrow{\rho} G \otimes H \xrightarrow{\alpha} (G/Z) \otimes (H/Z) \rightarrow 1.$$

Proof: Firstly, by Equation 1.8 we have for all $g \in G$, $h \in H$ and $z_1, z_2 \in Z$,

$$[g \otimes z_1, z_2 \otimes h] = (g^{z_1} g^{-1}) \otimes (z_2 h h^{-1}) = 1$$

so that the homomorphism ρ is well defined. Letting S denote $\rho((G \otimes Z) \times (Z \otimes H))$ we have $S \leq \ker(\alpha)$ and so α induces a homomorphism $\tilde{\alpha} : G \otimes H/S \rightarrow (G/Z) \otimes (H/Z)$. We produce the inverse of $\tilde{\alpha}$.

Define the homomorphism $\phi : (G/Z) \otimes (H/Z) \rightarrow G \otimes H/S$ on generators by $\phi(gZ \otimes hZ) = g \otimes hS$. Now ϕ is well defined since if $g \in G$, $h \in H$ and $z_1, z_2 \in Z$ then

$$\begin{aligned} \phi(gz_1Z \otimes hz_2Z) &= (gz_1 \otimes hz_2)S \\ &= (g \otimes h)(z_1 \otimes h)(g \otimes z_2)(z_1 \otimes z_2)S \\ &= (g \otimes h)S. \end{aligned}$$

The result then follows. ■

3.2.3 Proposition *Suppose that G and H are normal subgroups of E , and that actions arise from conjugation in E .*

i) *If $p \neq q$ then $|G \otimes H| = 1$.*

ii) *Suppose that $p = q$ and that G is a d -generator group of order p^n , H is a d' -generator group of order $p^{n'}$, and $|_H G|/|_H G \cap \Phi G| = p^k$. Then*

$$|G \otimes H| \leq p^{nn' - (k+n-d)(n'-d')}.$$

Proof: i) Suppose $p \neq q$. Then $[G, H]$ is trivial since it lies in the intersection of G with H . Proposition 1.2.12 thus implies an isomorphism

$$G \otimes H \cong G^{ab} \otimes_Z H^{ab}$$

where the right-hand side denotes the usual tensor product of abelian groups. Now G^{ab} is a direct sum of cyclic p -groups and H^{ab} is a direct sum of cyclic q -groups. It follows from Proposition 1.2.18 and Example 1.2.17 that $G^{ab} \otimes_Z H^{ab}$ is trivial.

ii) Suppose $p = q$. We shall consider two cases.

Case 1: Suppose $[G, H] = 1$, and let $H^{ab} \cong C_1 \times C_2 \times \dots \times C_{d'}$ where C_i denotes a cyclic p -group. Proposition 1.2.12 yields

$$G \otimes H \cong G^{ab} \otimes H^{ab} \cong (G^{ab} \otimes C_1) \times (G^{ab} \otimes C_2) \times \dots \times (G^{ab} \otimes C_{d'}).$$

Therefore,

$$\begin{aligned} |G \otimes H| &\leq |G^{ab} \otimes C_1| \times \dots \times |G^{ab} \otimes C_{d'}| \\ &\leq |G^{ab}|^{d'} \\ &\leq |G|^{d'} \\ &= p^{nd'}. \end{aligned}$$

Now because

$$p^k = \frac{|{}_H G|}{|{}_H G \cap \Phi G|} = \left| \frac{{}_H G}{{}_H G \cap \Phi G} \right| = \left| \frac{{}_H G \Phi G}{\Phi G} \right| \leq \left| \frac{G}{\Phi G} \right| = p^d$$

we have $k \leq d$ and so

$$\begin{aligned} |G \otimes H| &\leq p^{nd'} \\ &= p^{nn' - (d+n-d)(n'-d')} \\ &\leq p^{nn' - (k+n-d)(n'-d')} \end{aligned}$$

as required.

Case 2: Suppose $[G, H] \neq 1$. Let $l = n + n'$ with $n \neq 0$, $n' \neq 0$. As an inductive hypothesis, suppose that the proposition has been proved for all p -groups G' , H' with $|G'| |H'| \leq p^t$. This hypothesis is certainly true for $t = 2$, since $|G'| |H'| \leq p^2$ implies that either $G' \cap H' = 1$ or $G' = H' = C_p$, and consequently that $[G', H'] = 1$. Suppose $l = t + 1$.

Suppose now that T is a group of order p^t and S is a non-trivial normal subgroup of T of order p^s . The class equation for T may be written as

$$|T| = 1 + p^{t_1} + p^{t_2} + \dots + p^{p_h}$$

where each $t_i \geq 0$ and each term on the right hand side of the equation represents the size of a conjugacy class of T (see [49]). Since S is a normal subgroup of T it is a union of conjugacy classes and we have

$$|S| = 1 + p^{t_{i_1}} + p^{t_{i_2}} + \dots + p^{t_{i_k}}.$$

Now p divides the order of $|S|$ so $t_{i_j} = 0$ for some j . It follows that S intersects the centre of T non-trivially. There is thus a central subgroup N of GH such that $N \subseteq [G, H]$ and $|N| = p$.

By Lemma 3.2.1 the inclusion $N \subseteq [G, H]$ implies that the image of the canonical homomorphism ${}_H G \otimes N \rightarrow G \otimes H$ is contained in the image of the canonical homomorphism $N \otimes H \rightarrow G \otimes H$.

This together with taking $Z = N$ in Lemma 3.2.2 gives the exact sequence

$$(G/{}_H G \otimes N) \times (N \otimes H) \rightarrow G \otimes H \rightarrow (G/N) \otimes (H/N) \rightarrow 1.$$

This sequence and the isomorphism $(G/{}_H G) \otimes N \cong G/({}_H G \Phi G)$ yield

$$\begin{aligned}
|G \otimes H| &\leq |(G/N) \otimes (H/N)| |(G/{}_H G) \otimes N| |N \otimes H| \\
&= |(G/N) \otimes (H/N)| |G/({}_H G \Phi G)| |H/\Phi H| \\
&= |(G/N) \otimes (H/N)| p^{d-k} p^d \\
&\leq p^{(n-1)(n'-1)-(k+n-d-1)(n'-d'-1)+d-k+d'} \\
&= p^{nn'-(k+n-d)(n'-d')}
\end{aligned}$$

as required. ■

When $G = H$, Proposition 3.2.3 (ii) improves on a bound given by N.R.Rocco [51] (see also [19]).

Not all compatible actions arise from conjugation in a group E containing G and H . Suppose for instance that G and H are cyclic groups of order 8, with generators x and y . Compatible actions arise by defining $xy = y^{-1}$, $yx = x^{-1}$. These actions do not arise from conjugation, for if they did we would have $x^2 = x^y x^{-1} = x y y^{-1} = y^{-2}$, and thus the contradiction $x^2 = y x^2 = x^{-2}$.

So we now work towards a version of Proposition 3.2.3 for arbitrary compatible actions. Let us use the action of H on G to form the semi-direct product $G \rtimes H$, in which elements are multiplied according to the rule

$$(g, h).(g', h') = (g^h g', h h')$$

for $g, g' \in G$, $h, h' \in H$. Consider the subgroup P of this semi-direct product generated by the elements $(g^h g^{-1}, h^g h^{-1})$ for $g \in G$, $h \in H$. It is readily verified (or see for instance [6]) that P is a normal subgroup of $G \rtimes H$. Following [6], set $G \circ H = (G \rtimes H)/P$. The canonical homomorphisms $G \rightarrow G \rtimes H$, $g \mapsto (g, 1)$ and

$H \rightarrow G \rtimes H, h \mapsto (1, h)$ induce homomorphisms

$$\partial : G \rightarrow G \circ H$$

$$\delta : H \rightarrow G \circ H$$

The following is from [6].

3.2.4 Proposition [6] *The homomorphisms $\partial : G \rightarrow G \circ H$ and $\delta : H \rightarrow G \circ H$ are crossed modules.*

Proof: Firstly the action of H on G induces an action of $G \circ H$ on G , given by $(g, h)g' = g(hg')$. To see this we note that by compatibility of actions we have

$$\begin{aligned} (g^h g^{-1}, h^g h^{-1})g' &= g^h g^{-1} (h^g h^{-1} g') \\ &= g^h g^{-1} h^g h^{-1} g^{-1} g' \\ &= g'. \end{aligned}$$

Verification of the crossed module axioms is straight forward. ■

It follows that $\ker \partial$ is a central subgroup of G which is stable under the action of H , and thus that $\ker \partial$ is a $\mathbb{Z}H$ -module. Similarly $\ker \delta$ is a $\mathbb{Z}G$ -module.

Let $[H, \ker \partial]_G$ be that central subgroup of G generated by the elements hxx^{-1} for $h \in H, x \in \ker \partial$. Note that $[H, \ker \partial]_G$ is a $\mathbb{Z}H$ -module. We similarly have a $\mathbb{Z}G$ -module $[G, \ker \delta]_H$.

Let us denote by $H_1(G, A)$ the first Eilenberg Mac Lane homology of G with coefficients in a G -module A . Recall that $H_1(G, A)$ is isomorphic to the kernel of

the homomorphism $I_G \otimes_{\mathbb{Z}G} G \longrightarrow \mathbb{Z}G$ given by

$$(g - 1) \otimes h \mapsto (g - 1)h$$

where I_G denotes the augmentation ideal of G , that is I_G is the ideal of $\mathbb{Z}G$ generated by all $g - 1$ for $g \neq 1$ in G (see for example [49]).

We require an analogue to Lemma 3.2.2. Proposition 7 in [8] extends to yield the following.

3.2.5 Lemma *Any pair of crossed modules $\partial : G \rightarrow Q$ and $\delta : H \rightarrow Q$ yields an exact sequence*

$$(G \otimes \ker \delta) \times (\ker \partial \otimes H) \xrightarrow{\rho} G \otimes H \xrightarrow{\alpha} \bar{G} \otimes \bar{H} \rightarrow 1 \quad (3.1)$$

where $\bar{G} = \partial G$, $\bar{H} = \delta H$ are the images of G , H in Q .

Proof: Firstly we note that ρ is well defined since $\ker(\partial)$ and $\ker(\delta)$ act trivially (via Q) on both G and H . Then

$$[g \otimes_{z_1, z_2} h] = (g^{z_1} g^{-1}) \otimes ({}^{z_2} h h^{-1}) = 1$$

for all $g \in G$, $h \in H$ and $z_1 \in \ker(\delta)$, $z_2 \in \ker(\partial)$.

Letting S denote $\rho((G \otimes \ker \delta) \times (\ker \partial \otimes H))$ we have $S \leq \ker \alpha$ so that α induces a homomorphism $\tilde{\alpha} : G \otimes H/S \longrightarrow \bar{G} \otimes \bar{H}$. We produce the inverse of $\tilde{\alpha}$.

Define the homomorphism $\phi : \bar{G} \otimes \bar{H} \longrightarrow G \otimes H/S$ on generators by $\phi(\bar{g} \otimes \bar{h}) = g \otimes hS$ where g and h are any elements such that $\partial(g) = \bar{g}$ and $\delta(h) = \bar{h}$. Now ϕ is well defined since if $g \in G$, $h \in H$, $z_1 \in \ker(\partial)$ and $z_2 \in \ker(\delta)$ then

$$\phi(\bar{g}z_1 \otimes \bar{h}z_2) = (gz_1 \otimes hz_2)S$$

$$\begin{aligned}
&= (g \otimes h)(z_1 \otimes h)(g \otimes z_2)(z_1 \otimes z_2)S \\
&= (g \otimes h)S.
\end{aligned}$$

The result then follows. ■

The following is the main result of this chapter.

3.2.6 Theorem

i) If $p \neq q$ then $G \otimes H \cong [G, \ker\delta]_H \times [H, \ker\partial]_G$, and consequently

$$|G \otimes H| = |[G, \ker\delta]_H| |[H, \ker\partial]_G|.$$

ii) Suppose $p = q$, and that G is a d -generator group of order p^n , H is a d' -generator group of order $p^{n'}$, and $|{}_H G|/|{}_H G \cap \Phi G| = p^k$. Then

$$|G \otimes H| \leq K p^{nn' - (k+n-d)(n'-d')}$$

where $K = |H_1(G, \ker\delta)| |H_1(H, \ker\partial)| |[H, \ker\partial]_G| |[G, \ker\delta]_H|$.

Proof: Taking $Q = G \circ H$ in Lemma 3.2.5 note that \bar{G}, \bar{H} are both normal in $G \circ H$, and so they act on each other by conjugation. Corollary 3.3 in [30] provides an exact sequence

$$1 \rightarrow H_1(G, \ker\delta) \rightarrow G \otimes \ker\delta \xrightarrow{\mu} [G, \ker\delta]_H \rightarrow 1 \quad (3.2)$$

i) Suppose $p \neq q$. Proposition 3.2.3(i) tells us that $\bar{G} \otimes \bar{H} = 1$. Since G is a p -group and $\ker\delta$ is a q -group, we have $H_1(G, \ker\delta) = 1$ (see for instance [36]).

Sequence 3.2 implies

$$G \otimes \ker\delta \cong [G, \ker\delta]_H.$$

Similarly

$$\ker\partial \otimes H \cong [H, \ker\partial]_G$$

Since the composite homomorphism

$$[G, \ker\delta]_H \xrightarrow{\cong} G \otimes \ker\delta \rightarrow G \otimes H \rightarrow H$$

sends $[G, \ker\delta]_H$ injectively into H , the q -group $[G, \ker\delta]_H$ embeds into $G \otimes H$.

Similarly the p -group $[H, \ker\partial]_G$ embeds into $G \otimes H$. Since $p \neq q$, sequence 3.1 implies the required isomorphism

$$[G, \ker\delta]_H \times [H, \ker\partial]_G \cong G \otimes H.$$

ii) Suppose $p = q$. The required homomorphism follows from Proposition 3.2.3(ii) and the sequences 3.1 and 3.2. ■

We note that the isomorphism of Theorem 3.2.6(i) in fact holds for any two finite groups G, H with coprime exponents.

Let N be a normal subgroup in G . Using conjugation actions, we can form the tensor product $N \otimes G$. As explained in Proposition 1.2.4, there is an action of the group G on the tensor product $N \otimes G$ given by

$${}^g(n \otimes g') = ({}^g n \otimes {}^g g')$$

for $g, g' \in G, n \in N$. Conversely, an element $\tau \in N \otimes G$ acts on $g \in G$ by $\tau g = (\mu\tau)g(\mu\tau)^{-1}$. These actions are compatible and we can use them to form the tensor product $(N \otimes G) \otimes G$. This construction can be iterated to form the tensor product

$$N \otimes^{c+1} G = (((N \otimes G) \otimes G) \dots \otimes G)$$

of N with c copies of G . (In this notation, $N \otimes^3 G = (N \otimes G) \otimes G$.) Let us define a central series by

$$\gamma_1(N, G) = N,$$

$$\gamma_{i+1}(N, G) = [\gamma_i(N, G), G].$$

There is a canonical surjection $\mu : N \otimes^c G \twoheadrightarrow \gamma_c(N, G)$ which sends a tensor $((n \otimes g_1) \otimes g_2) \dots \otimes g_c$ to the commutator $[[[n, g_1], g_2] \dots, g_c]$.

The following corollary is the basis for the main results of [10] and [17].

3.2.7 Corollary *Let N be a normal subgroup of a d -generator p -group G . Suppose that $|\gamma_i(N, G)| = p^{m_i}$ for $i = 1, 2, \dots$. Then, for any $c \geq 1$, we have*

$$|N \otimes^{c+1} G| \leq p^{m_{cd} + m_{c-1}d^2 + \dots + m_1d^c}.$$

Proof: For $c = 1$ the corollary follows from Proposition 3.2.3(ii). For arbitrary $i \geq 1$ let us define

$$J_i(N, G) = \ker(\mu : N \otimes^i G \longrightarrow \gamma_i(N, G)).$$

For $c \geq 2$, there is thus an exact sequence

$$J_c(N, G) \otimes G \rightarrow (N \otimes^c G) \otimes G \rightarrow \gamma_c(N, G) \otimes G \rightarrow 1. \quad (3.3)$$

Now G and $J_c(N, G)$ act trivially on each other. Thus $|J_c(N, G) \otimes G| \leq |J_c(N, G)|^d \leq |N \otimes^c G|^d$. Also, Proposition 3.2.3(ii) implies $|\gamma_c(N, G) \otimes G| \leq p^{m_{cd}}$. So sequence 3.3 provides the recurrence relation

$$|N \otimes^{c+1} G| \leq p^{m_{cd}} |N \otimes^c G|$$

from which the corollary follows. ■

3.3 A topological application

In this section we give a topological application of the results of the last section.

For completeness sake we review here some of the pertinent topological definitions.

Suppose (X, x_0) is a pointed topological space, that is, X is a topological space

and x_0 is fixed point in X . Let I denote the closed unit interval $[0, 1]$ and ∂I denote

its boundary, the discrete space $\{0, 1\}$. By a path in X we mean a continuous

map $f : I \rightarrow X$ and by a closed path we mean a path f such that $f(0) = f(1)$.

Of particular interest is the case when $f(0) = f(1) = x_0$. We restrict ourselves to

this case and recall that in this situation the product of two paths may be defined.

3.3.1 Definition Let $f, g : I \rightarrow X$ be paths with $f(0) = f(1) = g(0) = g(1) =$

x_0 . By $f * g$ we mean the path $f * g : I \rightarrow X$ such that

$$(f * g)(t) = \begin{cases} g(2t) & 0 \leq t \leq \frac{1}{2} \\ f(2t - 1) & \frac{1}{2} \leq t \leq 1 \end{cases}$$

We may represent this composition pictorially as follows:

$$x_0 \xrightarrow[f]{X} x_0 \quad * \quad x_0 \xrightarrow[g]{X} x_0 \quad = \quad x_0 \xrightarrow[f * g]{X} x_0$$

Although the set of all closed paths in a pointed topological space can be given

an algebraic structure using $*$, a more appropriate structure is obtained when we

consider homotopy classes of paths. Here is the definition.

3.3.2 Definition Let X and Y be topological spaces with $B \subset X$ and $A \subset Y$.

Two maps $f, g : Y \rightarrow X$ with $f(A) \subset B$ and $g(A) \subset B$ are said to be *homotopic*

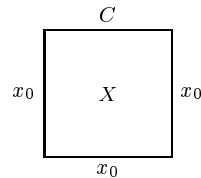
relative to A if there is a continuous map $F : Y \times I \rightarrow X$ with $F(y, 0) = f(y)$,

$F(y, 1) = g(y)$ for all $y \in Y$ and $F(a, t) \in B$ for all $a \in A$ and $t \in I$. If f and g are homotopic relative to A we write $f \simeq_A g$.

We can think of the homotopy F as a family of maps $f_t(y) = F(y, t)$ with the parameter t representing time.

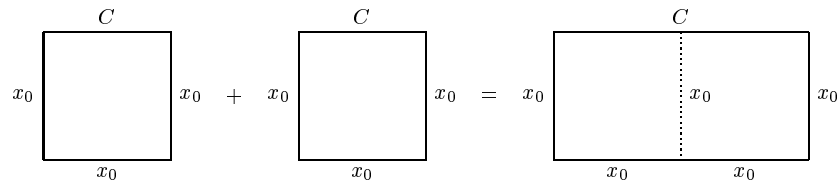
Applying this definition when f and g are paths we take Y to be I and A to be ∂I . In this case we see that two paths are homotopic if one can be “continuously deformed” into the other while holding the endpoints fixed at the point x_0 . Homotopy relative to ∂I forms an equivalence relation on the set of all paths. We denote the equivalence class of the path f by $[f]$. It is routine to check (or see [53]) that under the composition $[f][g] = [f * g]$ the homotopy classes of maps relative to ∂I form a group. This group is called the *fundamental group* of the pointed space (X, x_0) and is denoted by $\pi_1(X, x_0)$ or simply by $\pi_1 X$ when we do not wish to emphasize the point x_0 .

We now recall the second homotopy groups of a pointed space (X, x_0) . Let C be a subspace of X which contains the basepoint x_0 . The appropriate maps here are the continuous functions $f : I \times I \rightarrow X$ such that three sides of the square $I \times I$ are mapped to the basepoint x_0 while the fourth is mapped to C . Pictorially we have



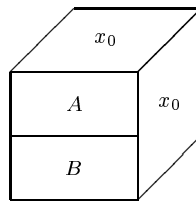
Homotopy classes of such functions can be formed relative to the boundary of $I \times I$ and again we denote the conjugacy class of f by $[f]$. Composition of

such conjugacy classes is defined analogously with the one dimensional case. The appropriate picture is:

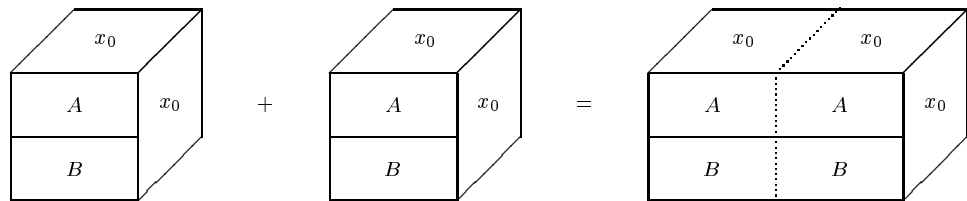


The conjugacy classes of such maps under this composition is a group called the *second homotopy group relative to C* and denoted by $\pi_2(X, C)$. The *second homotopy group* of X is the relative homotopy group in the special case that $C = \{x_0\}$ and is denoted by $\pi_2 X$.

Suppose now that A and B are subspaces of X both containing the basepoint x_0 and that $C = A \cap B$. The third relative homotopy group of X relative to A and B consists of equivalent classes of continuous functions $f : I \times I \times I \rightarrow X$ where five of the six faces of the cube $I \times I \times I$ are mapped to the base point x_0 . The top half of the sixth face is mapped to A , the bottom half to B as in the diagram. We note that the border between the two parts is of necessity mapped to C .



Homotopy classes are again defined relative to the boundary of the cube with five sides mapped to x_0 and composition of conjugacy classes is defined via:



The group formed in this way is called the *triad homotopy group* and is denoted by $\pi_3(X, A, B)$. When $A = B = \{x_0\}$ this group is denoted by $\pi_3 X$ and is called the *third homotopy group* of X .

By the *n-sphere* in \mathbb{R}^{n+1} we mean $S^n = \{x \in \mathbb{R}^{n+1} : \|x\| = 1\}$ where $\|x\|$ denotes the Euclidean norm. By the *closed n-disk* or *closed n-ball* in \mathbb{R}^n we mean $D^n = \{x \in \mathbb{R}^{n+1} : \|x\| \leq 1\}$. We note that S^0 is the two points $\{-1, 1\}$, and indeed S^{n-1} is the boundary of D^n in \mathbb{R}^n .

By an *n-cell*, e we mean a homeomorphic copy of the open n -disk $D^n \setminus S^{n-1}$. We denote the dimension of e by $\dim(e) = n$.

If a topological space X is a disjoint union of cells, $X = \cup\{e : e \in E\}$, then for each $k \geq 0$ the *k-skeleton* $X^{(k)}$ of X is defined to be $X^{(k)} = \cup\{e \in E : \dim(e) \leq k\}$.

Suppose now that X and Y are topological spaces with $A \subset X$ and $B \subset Y$. If g is a continuous map $g : X \rightarrow Y$ such that $g|_{X \setminus A} : X \setminus A \rightarrow Y \setminus B$ is a homeomorphism then we call g a *relative homeomorphism* and denote it by $g : (X, A) \rightarrow (Y, B)$.

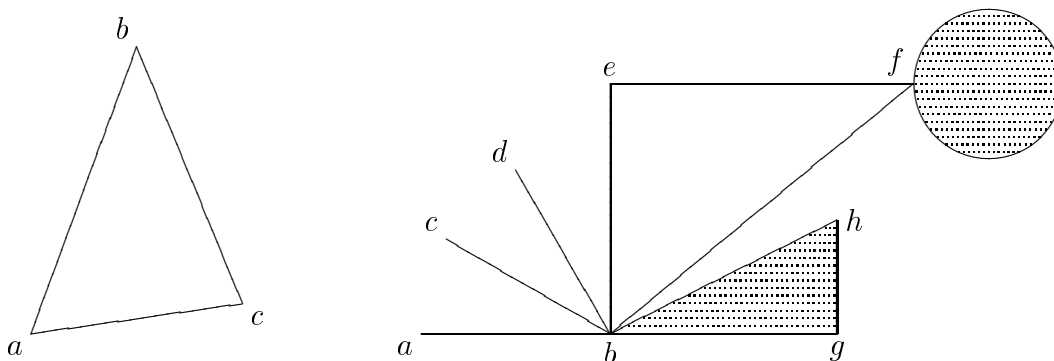
3.3.3 Definition A *CW-complex* is an ordered triple (X, E, Φ) , where X is a Hausdorff space, E is a family of cells in X , and $\Phi = \{\Phi_e : e \in E\}$ is a family of continuous maps such that

- i) $X = \cup\{e : e \in E\}$ a disjoint union,
- ii) for each k -cell $e \in E$, the map $\Phi_e(D^k, S^{k-1}) \rightarrow (e \cup X^{k-1}, X^{k-1})$ is a relative homeomorphism,

- iii) if $e \in E$ then its closure \bar{e} is contained in a finite union of cells in E ,
- iv) X has the weak topology determined by $\{\bar{e} : e \in E\}$.

Parts *iii*) and *iv*) in the definition give the *CW*-complex its name. The “C” being for “closure finite” and the “W” for the “weak topology”.

The figures in the diagram below are *CW*-complexes in R^2 . The first figure is composed of the points a , b and c (the 0-cells), and the open line segments (a,b) , (b,c) and (a,c) (the 1-cells). In the same way the second figure can be seen to be made of points, open line segments and the open plane figures (the 2-cell shaded in the diagram). We think of these component parts being glued together to form the *CW*-complex.



We note that we can regard S^n as a *CW*-space with one 0-cell and one n -cell.

Suppose that a *CW*-space X is a union $X = A \cup B$ of two path-connected *CW*-subspaces A and B whose intersection $C = A \cap B$ is path-connected. Some of the homotopy structure of the space X can be calculated in terms of the homotopy structure of the spaces A , B and C . For instance, van Kampen’s theorem on the fundamental group (see for example [53]) describes $\pi_1 X$ as an amalgamated sum

of groups:

$$\pi_1 X \cong \pi_1 A *_{\pi_1 C} \pi_1 B.$$

By the amalgamated sum $\pi_1 A *_{\pi_1 C} \pi_1 B$ we mean the quotient of the free product of $\pi_1 A$ and $\pi_1 B$ by the normal subgroup generated by all the relations of the form $\alpha([f]) = \beta([f])$ where $[f] \in \pi_1 C$ and α and β are the canonical homomorphisms $\alpha : \pi_1 C \rightarrow \pi_1 A$, $\beta : \pi_1 C \rightarrow \pi_1 B$. A “two-dimensional analogue” of van Kampen’s Theorem is used in [6] to describe the second relative homotopy group $\pi_2(X, C)$:

$$\pi_2(X, C) \cong \pi_2(A, C) \circ \pi_2(B, C).$$

Here the symbol \circ denotes the construction of the previous section, the groups $\pi_2(A, C)$, $\pi_2(B, C)$ acting on one another via the boundary homomorphisms $\pi_2(A, C) \rightarrow \pi_1 C$, $\pi_2(B, C) \rightarrow \pi_1 C$ and actions of $\pi_1 C$.

A “three-dimensional analogue” of van Kampen’s theorem is used in [9] to describe the triad homotopy group $\pi_3(X, A, B)$ under the hypothesis that the canonical homomorphisms $\pi_1 C \rightarrow \pi_1 A$, $\pi_1 C \rightarrow \pi_1 B$ are surjective:

$$\pi_3(X, A, B) \cong \pi_2(A, C) \otimes \pi_2(B, C).$$

Using the exact sequences (for $n \geq 1$) (see [56])

$$\begin{aligned} &\rightarrow \pi_n C \rightarrow \pi_n A \rightarrow \pi_n(A, C) \rightarrow \pi_{n-1} C \rightarrow \\ &\rightarrow \pi_n C \rightarrow \pi_n B \rightarrow \pi_n(B, C) \rightarrow \pi_{n-1} C \rightarrow \\ &\rightarrow \pi_n(B, C) \rightarrow \pi_n(X, A) \rightarrow \pi_n(X, A, B) \rightarrow \pi_{n-1}(B, C) \rightarrow \end{aligned}$$

one readily obtains the following bound on $\pi_3 X$.

3.3.4 Proposition *Suppose that the canonical homomorphisms $\pi_1 C \rightarrow \pi_1 A$, $\pi_1 C \rightarrow \pi_1 B$ are surjective. Then*

$$|\pi_3 X| \leq \frac{|\pi_1 A| \cdot |\pi_3 A|}{|\pi_2 A|} \cdot \frac{|\pi_1 B| \cdot |\pi_3 B|}{|\pi_2 B|} \cdot \frac{|\pi_2 C|^2}{|\pi_1 C|^2} \cdot |\pi_2(A, C) \circ \pi_2(B, C)| \cdot |\pi_2(A, C) \otimes \pi_2(B, C)|.$$

The bound is attained if, for instance, the homotopy groups $\pi_3 A$, $\pi_3 B$, $\pi_2 C$ are all trivial. ■

We clearly have

$$|\pi_2(A, C) \circ \pi_2(B, C)| \leq |\pi_2(A, C)| |\pi_2(B, C)|.$$

Thus Theorem 3.2.6 yields an explicit bound on $|\pi_3 X|$ in the case where $\pi_2(A, C)$ and $\pi_2(B, C)$ are known prime power groups.

3.4 Computer computations

As an illustration of the accuracy of the bound in Proposition 3.2.3 we now consider all pairs of normal subgroups of the quaternion group, $Q_{32} = \langle a, b | a^{16} = b^8 a^2 = ab^{-1}ab = 1 \rangle$, with actions being conjugation in Q_{32} . The nonabelian tensor product was calculated for each of these in the last chapter.

Table 3.1 lists $|G \otimes H|$ for all pairs of normal subgroups G and H in Q_{32} (see Appendix B, Q3.g). Since $G \otimes H \cong H \otimes G$ the table only includes the case $|G| \leq |H|$. We consider two pairs (G, H) and (G', H') to be isomorphic if there is an isomorphism $\phi : GH \rightarrow G'H'$ that restricts to isomorphisms $G \xrightarrow{\cong} G'$, $H \xrightarrow{\cong} H'$. Since isomorphic pairs yield isomorphic tensor products, the table contains just one entry for each isomorphism class of pairs. There is a certain asymmetry in the bound of Proposition 3.2.3 since k depends only on G , yet $G \otimes H \cong H \otimes G$.

Thus a smaller bound may sometimes be obtained by interchanging the roles of G and H . However, the table does not involve such interchanges.

Table 3.1: $|G \otimes H|$ for $G, H \triangleleft Q_{32}$

Generators G	$ G $	Generators H	$ H $	$\log_2(G \otimes H)$	$\log_2(\text{Bound of Theorem 3.2.6})$
a^2	2	a^2	2	1	1
a^2	2	a	4	1	1
a^2	2	a^2, ab^4	4	2	2
a^2	2	a, b^4	8	2	2
a^2	2	a^2b^2	8	1	1
a^2	2	a, b^2	16	2	2
a^2	2	b	16	1	1
a^2	2	a, b	32	2	2
a	4	a	4	2	2
a	4	a^2, ab^4	4	2	4
a	4	a, b^4	8	3	4
a	4	a^2b^2	8	2	2
a	4	a, b^2	16	3	4
a	4	b	16	2	5
a	4	a, b	32	3	4
b^4	4	b	16	2	2
b^4	4	a, b	32	3	4
a^2, ab^4	4	a^2, ab^4	4	4	4
a^2, ab^4	4	a, b^4	8	4	4
a^2, ab^4	4	a^2b^2	8	2	2
a^2, ab^4	4	a, b^2	16	4	4
a^2, ab^4	4	b	16	2	5
a^2, ab^4	4	a, b	32	3	4
a, b^4	8	a, b^4	8	5	6
a, b^4	8	a^2b^2	8	3	3
a, b^4	8	a, b^2	16	5	6
a, b^4	8	b	16	3	6
a, b^4	8	a, b	32	2	6
a^2b^2	8	a^2b^2	8	3	3
a^2b^2	8	a, b^2	16	4	6
a^2b^2	8	b	16	3	3
a^2b^2	8	a, b	32	4	6
ab^2	8	b	16	3	6
ab^2	8	a, b	32	4	6
a, b^2	16	a, b^2	16	6	8
a, b^2	16	b	16	4	7
a, b^2	16	a, b	32	2	8
b	16	b	16	4	4
b	16	ab	16	4	7
b	16	a, b	32	5	8
ab	16	a, b	32	5	8
a, b	32	a, b	32	7	10

Chapter 4

Bounds on nonabelian tensor squares of prime–power groups

4.1 Introduction

In this chapter we use the method of [15] to improve a bound of Rocco [51] on the order of the nonabelian tensor square $G \otimes G$ of a prime–power group G . We also give bounds for the exponent of the tensor square.

Throughout p is a prime and q can take the values 0 or p . If $N \trianglelefteq G$, we denote by $N \#^q G$ the subgroup of G generated by the commutators $ngn^{-1}g^{-1}$, and powers n^q , for $n \in N, g \in G$. We then define $\gamma_i^q G$ recursively as

$$\begin{aligned}\gamma_1^q G &= G \\ \gamma_{i+1}^q G &= (\gamma_i^q G) \#^q G \quad \text{for } i \geq 1.\end{aligned}$$

We can form the factor groups $\Gamma_i^q G = \gamma_i^q G / \gamma_{i+1}^q G$ for $i \geq 1$, and we note that when $q = 0$, $\gamma_i^0 G$ are the usual terms of the lower central series of G . In this case we often write $\gamma_i G$ for $\gamma_i^0 G$, and $\Gamma_i G$ for $\Gamma_i^0 G$.

4.2 Preliminaries

4.2.1 Lemma *Suppose M and N are normal subgroups of a group G , with N a subgroup of M and $[M, G] \leq N$. Then the image of the canonical homomorphism*

$$N \otimes G \xrightarrow{\iota} M \otimes G$$

is normal in $M \otimes G$. In particular, for each i , $\iota(\gamma_{i+1}^q G \otimes G) \trianglelefteq \gamma_i^q G \otimes G$.

Proof: By Proposition 1.2.11 there is a crossed N -module $\partial : M \otimes G \rightarrow N$.

Let $\tau \in M \otimes G$, and let $n \otimes g$ be the image under ι of a generator of $N \otimes G$.

Then $\tau(n \otimes g)\tau^{-1} = \partial\tau(n \otimes g) = \partial\tau n \otimes \partial\tau g \in \iota(N \otimes G)$. Taking $N = \gamma_{i+1}^q G$ and

$M = \gamma_i^q G$ it follows that $\iota(\gamma_{i+1}^q G \otimes G) \trianglelefteq \gamma_i^q G \otimes G$. ■

4.2.2 Lemma *For each $i \geq 1$ and $q = 0$ or p ,*

$$(G/\gamma_{i+1}^q G) \otimes \Gamma_i^q G \cong \Gamma_1^q G \otimes \Gamma_i^q G,$$

where all groups act on each other by the action inherited from conjugation in G .

Proof: From the definition of $\gamma_i^q G$ it is easy to see that $G/\gamma_{i+1}^q G$ and $\gamma_i^q G/\gamma_{i+1}^q G$

act on each other trivially. By Proposition 1.2.12 we then have

$$\frac{G}{\gamma_{i+1}^q G} \otimes \frac{\gamma_i^q G}{\gamma_{i+1}^q G} \cong \left(\frac{G}{\gamma_{i+1}^q G} \right)^{ab} \otimes_{\mathbb{Z}} \left(\frac{\gamma_i^q G}{\gamma_{i+1}^q G} \right)^{ab},$$

where $\otimes_{\mathbb{Z}}$ is the usual tensor product of abelian groups. Now $\gamma_i^q G/\gamma_{i+1}^q G$ is abelian

and we have

$$\left(\frac{G}{\gamma_{i+1}^q G} \right)^{ab} \cong \frac{G/\gamma_{i+1}^q G}{\gamma_2^q G/\gamma_{i+1}^q G} \cong \frac{G}{\gamma_2^q G \gamma_{i+1}^q G},$$

so

$$\frac{G}{\gamma_{i+1}^q G} \otimes \frac{\gamma_i^q G}{\gamma_{i+1}^q G} \cong \frac{G}{\gamma_2 G \gamma_{i+1}^q G} \otimes_{\mathbb{Z}} \Gamma_i^q G.$$

Now if $q = 0$ then $\gamma_{i+1}^q G \leq \gamma_2 G$ and the result follows. It remains to deal with the case $q = p$. Suppose $\hat{a} \in G/\gamma_2 G \gamma_{i+1}^q G$ and $\hat{b} \in \Gamma_i^q G$. Then

$$\begin{aligned} \hat{a}^q \otimes_{\mathbb{Z}} \hat{b} &= (\hat{a} \otimes_{\mathbb{Z}} \hat{b})^q \\ &= \hat{a} \otimes_{\mathbb{Z}} \hat{b}^q \\ &= 1, \end{aligned}$$

and so it follows that

$$\frac{G}{\gamma_2 G \gamma_{i+1}^q G} \otimes_{\mathbb{Z}} \Gamma_i^q G \cong \Gamma_1^q G \otimes \Gamma_i^q G. \quad \blacksquare$$

4.2.3 Lemma *Let $N \trianglelefteq G$. The images of the canonical homomorphisms*

$$\begin{aligned} [N, G] \otimes N &\xrightarrow{\alpha} N \otimes G \\ N \otimes [N, G] &\xrightarrow{\beta} N \otimes G \end{aligned}$$

coincide in $N \otimes G$.

Proof: By Proposition 1.2.11 there is a crossed module

$$\lambda : N \otimes G \rightarrow [N, G]$$

such that $\lambda(t) \otimes g = t^g t^{-1}$ and $n \otimes \lambda(t) = {}^n t t^{-1}$ for all $t \in N \otimes G$, $g \in G$ and $n \in N$.

Now if $h \in [N, G]$ then $h = \lambda(t)$ for some $t \in N \otimes G$ and if $n \in N$ we have

$$\begin{aligned} h \otimes n = \lambda(t) \otimes n &= t^n t^{-1} \\ &= ({}^n t t^{-1})^{-1} = (n \otimes \lambda(t))^{-1} = (n \otimes h)^{-1}. \end{aligned}$$

It follows that $\alpha([N, G] \otimes N) = \beta(N \otimes [N, G])$. ■

We are mainly interested in the preceding lemma in the cases where $N = \gamma_i G$ and $N = \gamma_i^p G$ and note that in the first of these cases $[N, G]$ is just $\gamma_{i+1} G$.

4.2.4 Lemma *For each i , the following sequences of canonical homomorphisms are exact.*

$$\gamma_{i+1} G \otimes G \xrightarrow{\iota} \gamma_i G \otimes G \xrightarrow{\pi} (\gamma_i G / \gamma_{i+1} G) \otimes (G / \gamma_{i+1} G) \rightarrow 0$$

$$[\gamma_i^q G, G] \otimes G \xrightarrow{\epsilon} \gamma_i^q G \otimes G \xrightarrow{\rho} \gamma_i^q G / [\gamma_i^q G, G] \otimes G / [\gamma_i^q G, G] \rightarrow 0.$$

Proof: Clearly $\iota(\gamma_{i+1} G \otimes G) \leq \ker(\pi)$ and by Lemma 4.2.1, $\iota(\gamma_{i+1} G \otimes G) \leq \gamma_i G \otimes G$. Thus π induces a homomorphism $\tilde{\pi} : (\gamma_i G \otimes G) / \iota(\gamma_{i+1} G \otimes G) \rightarrow (\gamma_i G / \gamma_{i+1} G) \otimes (G / \gamma_{i+1} G)$. We produce the inverse of this map. Define the homomorphism ψ on the generators of $(\gamma_i G / \gamma_{i+1} G) \otimes (G / \gamma_{i+1} G)$ by

$$\psi(a\gamma_{i+1} G \otimes b\gamma_{i+1} G) = a \otimes b\iota(\gamma_{i+1} G \otimes G)$$

We need to show that ψ is well defined. Suppose $n \in \gamma_{i+1} G$. Then $\psi(an\gamma_{i+1} G \otimes b\gamma_{i+1} G) = (an \otimes b)\iota(\gamma_{i+1} G \otimes G) = ({}^a n \otimes {}^a b)(a \otimes b)\iota(\gamma_{i+1} G \otimes G) = a \otimes b\iota(\gamma_{i+1} G \otimes G)$. Similarly, $\psi(a\gamma_{i+1} G \otimes bn\gamma_{i+1} G) = a \otimes bn\iota(\gamma_{i+1} G \otimes G) = (a \otimes b)({}^b a \otimes {}^b n)\iota(\gamma_{i+1} G \otimes G) = (a \otimes b)\iota(\gamma_{i+1} G \otimes G)$, since by Lemma 4.2.3 with N taken to be $\gamma_i G$ we have the images of $\gamma_{i+1} G \otimes \gamma_i G$ and $\gamma_i G \otimes \gamma_{i+1} G$ coincide in $\gamma_i G \otimes G$. Hence ψ is well-defined and clearly is the inverse of $\tilde{\pi}$. It follows that

$$(\gamma_i G / \gamma_{i+1} G) \otimes (G / \gamma_{i+1} G) \cong (\gamma_i G \otimes G) / \iota(\gamma_{i+1} G \otimes G),$$

and so the first sequence is exact.

The second sequence can be shown to be exact in the same way, this time using

Lemma 4.2.3 with $N = \gamma_i^q G$. ■

The second exact sequence in the preceding lemma can be used to produce a “ q ” version of the first. In order to do this we need the following.

4.2.5 Lemma *The images of the canonical homomorphisms*

$$\begin{aligned}\gamma_{i+1}^q G \otimes \gamma_i^q G &\xrightarrow{\alpha} \gamma_i^q G \otimes G \\ \gamma_i^q G \otimes \gamma_{i+1}^q G &\xrightarrow{\beta} \gamma_i^q G \otimes G\end{aligned}$$

coincide in $\gamma_i^q \otimes G$.

Proof: Let a be a generator of $\gamma_{i+1}^q G$ (that is a is of the form $[x, y]$ for some $x, y \in \gamma_i^q G$ or a is of the form x^q for some $x \in \gamma_i^q G$) and $b \in \gamma_i^q G$. If a is a commutator then Lemma 4.2.3 with $N = \gamma_i^q G$ shows that $b \otimes a = (a \otimes b)^{-1}$ in $\gamma_i^q G \otimes G$. In this case $b \otimes a \in \alpha(\gamma_{i+1}^q \gamma_i^q G)$.

Suppose then that a is a generator of $\gamma_{i+1}^q G$ of the form s^q for some $s \in \gamma_i^q G$.

Consider the image of $b \otimes a$ in $\gamma_i^q G \otimes G / \epsilon([\gamma_i^q G, G] \otimes G)$. By the second exact sequence of the previous lemma we have

$$\frac{\gamma_i^q G \otimes G}{\epsilon([\gamma_i^q G, G] \otimes G)} \cong \frac{\gamma_i^q G}{[\gamma_i^q G, G]} \otimes \frac{G}{[\gamma_i^q G, G]}.$$

Since $\gamma_i^q G / [\gamma_i^q G, G]$ and $G / [\gamma_i^q G, G]$ act on each other trivially we have

$$\begin{aligned}b \otimes a \epsilon([\gamma_i^q G, G]) &= b \otimes s^q \epsilon([\gamma_i^q G, G]) \\ &= b^q \otimes s \epsilon([\gamma_i^q G, G]).\end{aligned}$$

But $\gamma_i^q G \otimes G / \alpha(\gamma_i^q G \otimes G)$ is a factor group of $\gamma_i^q G \otimes G / \epsilon([\gamma_i^q G, G] \otimes G)$ and so

$$b \otimes a \alpha(\gamma_i^q G \otimes G) = b^q \otimes s \alpha(\gamma_i^q G \otimes G) = 1$$

in $\gamma_i^q G \otimes G / \alpha(\gamma_i^q G \otimes G)$. The result now follows. ■

4.2.6 Lemma *For each i and $q = 0$ or p the following sequence of canonical homomorphisms is exact.*

$$\gamma_{i+1}^q G \otimes G \xrightarrow{\iota} \gamma_i^q G \otimes G \xrightarrow{\pi} (\gamma_i^q G / \gamma_{i+1}^q G) \otimes (G / \gamma_{i+1}^q G)$$

Proof: When $q = 0$ this is just the first sequence in Lemma 4.2.4 and when $q = p$ the proof is identical to that of Lemma 4.2.4 where we now use Lemma 4.2.5 instead of Lemma 4.2.3. ■

4.2.7 Lemma *Suppose G is finite and set $\Gamma^q = \Gamma_1^q G \times \Gamma_2^q G \times \dots \times \Gamma_c^q G$. Then*

$$|G \otimes G| \leq |G^{ab} \otimes \Gamma^q|.$$

Proof: By repeated application of Lemma 4.2.6 we see that

$$\begin{aligned} |G \otimes G| &\leq |\Gamma_1^q G \otimes \Gamma_1^q G| |G \otimes \gamma_2^q G| \\ &\leq |\Gamma_1^q G \otimes \Gamma_1^q G| |(G / \gamma_3^q G) \otimes \Gamma_2^q G| |(G / \gamma_4^q G) \otimes \Gamma_3^q G| \dots \\ &\quad \dots |(G / \gamma_{c+1}^q G) \otimes \Gamma_c^q G| \\ &= |\Gamma_1^q G \otimes \Gamma_1^q G| |\Gamma_1^q G \otimes \Gamma_2^q G| |\Gamma_1^q G \otimes \Gamma_3^q G| \dots |\Gamma_1^q G \otimes \Gamma_c^q G| \\ &= |\Gamma_1^q G \otimes (\Gamma_1^q G \times \Gamma_2^q G \times \dots \times \Gamma_c^q G)| \\ &= |\Gamma_1^q G \otimes \Gamma^q|. \end{aligned} \quad \blacksquare$$

Finally we need the following result which is a consequence of Theorem 9 in [24].

We note that the proof there is based on homological arguments. Here we give an algebraic proof. I would like to thank Professor M. Newell for his help with the following proof.

4.2.8 Lemma In $G \otimes G$,

$$([x, y] \otimes {}^y z)([y, z] \otimes {}^z x)([z, x] \otimes {}^x y) = 1,$$

for all x, y , and z in G .

Proof: Consider the action of G on $G \otimes G$. We have

$${}^z(y \otimes x) = ({}^z y \otimes {}^z x). \quad (4.1)$$

Acting on the left hand side of (4.1) by $[y, z]$ gives

$$\begin{aligned} [y, z]({}^z(y \otimes x)) &= [y, z]({}^z(y \otimes x)) \\ &= {}^{yz}y^{-1}(y \otimes x) \\ &= ({}^{yz})(y \otimes x) \\ &= (y \otimes x)(y \otimes x)^{-1} ({}^{yz})(y \otimes x) \\ &= (y \otimes x)([y, x]^{-1} \otimes {}^y z) && \text{using (1.6)} \\ &= (y \otimes x)([x, y] \otimes {}^y z). \end{aligned}$$

The right hand side of (4.1) is

$$\begin{aligned} {}^z y \otimes {}^z x &= [y, z]^{-1} y \otimes {}^z x \\ &= [y, z]^{-1} (y \otimes {}^z x)([y, z]^{-1} \otimes {}^z x) && \text{using (1.1)} \\ &= [y, z]^{-1} (y \otimes {}^z x)([y, z]^{-1} ([y, z] \otimes {}^z x))^{-1} && \text{using (1.4)} \\ &= [y, z]^{-1} ((y \otimes {}^z x)([y, z] \otimes {}^z x)^{-1}). \end{aligned}$$

Acting by $[y, z]$ gives

$$[y, z]({}^z y \otimes {}^z x) = (y \otimes {}^z x)([y, z] \otimes {}^z x)^{-1}$$

Now

$$\begin{aligned}
y \otimes^z x &= y \otimes x[x^{-1}, z] \\
&= (y \otimes x)^x (y \otimes [x^{-1}, z]) && \text{using (1.2)} \\
&= (y \otimes x)^x (y \otimes^x [x^{-1}, z]) \\
&= (y \otimes x)^x (y \otimes [z, x]) \\
&= (y \otimes x)([z, x] \otimes^x y)^{-1}
\end{aligned}$$

by (1.6).

Hence

$$\begin{aligned}
(y \otimes x)([x, y] \otimes^y z) &= (y \otimes^z x)([y, z] \otimes^z x)^{-1} \\
&= (y \otimes x)([z, x] \otimes^x y)^{-1}([y, z] \otimes^z x)^{-1}
\end{aligned}$$

from which it follows that

$$([x, y] \otimes^y z)([y, z] \otimes^z x)([z, x] \otimes^x y) = 1$$

■

4.3 Bounds on the order of $G \otimes G$

Suppose now that G is a group of order p^n with p a prime and $\Gamma_{c+1}G = 1$. If

$\Gamma_1^q G \cong C_{p^{n_1}} \times C_{p^{n_2}} \times \dots \times C_{p^{n_d}}$, and $\Gamma_1^q G \times \Gamma_2^q G \times \dots \times \Gamma_c^q G \cong C_{p^{m_1}} \times C_{p^{m_2}} \times \dots \times C_{p^{m_s}}$

where C_k denotes the cyclic group of order k , then we write

$$M_q = \sum_{n_i, m_j} \min(n_i, m_j).$$

We note that m_i, n_i, d and s depend on q . When $q = 0$, c is the nilpotency class of G . When $q = p$, $m_i = 1$ for all $1 \leq i \leq s$ and $n_j = 1$ for all $1 \leq j \leq d$, so that

$$M_p = ds.$$

Let $\mathfrak{L} = \Gamma_1^p G \oplus \Gamma_2^p G \oplus \dots \oplus \Gamma_c^p G$. Then \mathfrak{L} has the structure of a restricted Lie algebra over the field \mathbb{Z}_p , where the Lie bracket $\mathfrak{b} : \mathfrak{L} \times \mathfrak{L} \longrightarrow \mathfrak{L}$ is induced by commutation in G and $\mathfrak{p} : \mathfrak{L} \longrightarrow \mathfrak{L}$ is induced by taking p^{th} powers. For each $i \geq 1$ the Lie bracket \mathfrak{b} restricts to $\mathfrak{b} : \Gamma_i^p G \times \Gamma_i^p G \longrightarrow \Gamma_{i+1}^p G$ a linear homomorphism of vector spaces (cf. [39]).

Now consider the linear homomorphisms defined for each $i \geq 2$ by

$$j_i : \Gamma_{i-1}^p G \otimes \Gamma_1^p G \otimes \Gamma_1^p G \longrightarrow \Gamma_i^p G \otimes \Gamma_1^p G,$$

$$x \otimes y \otimes z \longmapsto \mathfrak{b}(x, y) \otimes z + \mathfrak{b}(y, z) \otimes x + \mathfrak{b}(z, x) \otimes y.$$

We note that when $i < 2$, $\mathfrak{b}(y, z) \otimes x = 1$ in $\Gamma_i^p G \otimes \Gamma_1^p G$ for all $x \in \Gamma_{i-1}^p G$ and $y, z \in \Gamma_1^p G$. In this case j_i reduces to

$$j_i : \Gamma_{i-1}^p G \otimes \Gamma_1^p G \otimes \Gamma_1^p G \longrightarrow \Gamma_i^p G \otimes \Gamma_1^p G,$$

$$x \otimes y \otimes z \longmapsto \mathfrak{b}(x, y) \otimes z + \mathfrak{b}(z, x) \otimes y, \quad \text{for } i \geq 3.$$

Let a be defined as

$$a = \sum_{i=2}^c \dim_{\mathbb{Z}_p}(\text{image}(j_i)).$$

4.3.1 Theorem *Suppose G is d -generator group of order p^n . If G' denotes the commutator subgroup of G , then*

$$\text{i) } |G \otimes G| = |J(G)||G'| \leq p^{dn},$$

$$\text{ii) } |J(G)||G'| \leq p^{M_0},$$

$$\text{iii) } |J(G)||G'| \leq p^{M_p - a}$$

Proof: If for each i , $|\Gamma_i^q G| = p^{m_i}$, then

$$\begin{aligned} |G| &= |\Gamma_1^q G| |\Gamma_2^q G| \dots |\Gamma_c^q G| \\ &= p^{m_1 + m_2 + \dots + m_c} \end{aligned}$$

from which it follows that $n = m_1 + m_2 + \dots + m_c$. Recalling that $\Gamma^q = \Gamma_1^q G \otimes \Gamma_2^q G \otimes \dots \otimes \Gamma_d^q G$, Lemma 4.2.7 and the short exact sequence

$$J(G) \hookrightarrow G \otimes G \twoheadrightarrow \gamma_2 G \tag{4.2}$$

gives,

$$\begin{aligned} |J(G)| |\gamma_2 G| &= |G \otimes G| \\ &\leq |G^{ab} \otimes \Gamma^q| \\ &= |(C_{p^{n_1}} \times C_{p^{n_2}} \times \dots \times C_{p^{n_d}}) \otimes \Gamma^q| \\ &= |(C_{p^{n_1}} \otimes \Gamma^q) \times (C_{p^{n_2}} \otimes \Gamma^q) \times \dots \times (C_{p^{n_d}} \otimes \Gamma^q)| \\ &\leq \overbrace{|\Gamma^q| |\Gamma^q| \dots |\Gamma^q|}^{d \text{ copies}} \\ &= p^{nd}. \end{aligned}$$

Thus part (i) is proven.

To prove part (ii) we again start with Lemma 4.2.7 and the short exact sequence (4.2). Expanding both $\Gamma_1^q G$ and Γ^q we get

$$\begin{aligned} |J(G)| |\gamma_2 G| &= |G \otimes G| \\ &\leq |\Gamma_1^q G \otimes \Gamma^q| \\ &= |(C_{p^{n_1}} \times C_{p^{n_2}} \times \dots \times C_{p^{n_d}}) \otimes (C_{p^{m_1}} \times C_{p^{m_2}} \times \dots \times C_{p^{m_s}})| \\ &= \prod_{\substack{1 \leq i \leq d \\ 1 \leq j \leq s}} |C_{p^{n_i}} \otimes C_{p^{m_j}}| \end{aligned}$$

$$\begin{aligned}
&= \prod_{\substack{1 \leq i \leq d \\ 1 \leq j \leq s}} p^{\min(n_i, m_j)} \\
&= p^{M_q}.
\end{aligned}$$

When $q = 0$ this is ii). When $q = p$ we get $|G \otimes G| \leq p^{M_p}$. We can reduce this bound further.

Consider again the exact sequence in Lemma 4.2.6. By Lemma 4.2.8 we have

$$([x, y] \otimes {}^y z)([y, z] \otimes {}^z x)([z, x] \otimes {}^x y) = 1 \quad (4.3)$$

in $\gamma_i^p G \otimes \gamma_i^p G$ for all $x, y, z \in \gamma_1^p G$. Since there is a canonical homomorphism $\gamma_i^p G \otimes \gamma_i^p G \longrightarrow \gamma_i^p G \otimes G$ the identity (4.3) holds in $\gamma_i^p G \otimes G$ for all $x, y, z \in \gamma_i^p G$. It follows that regarded as an element of $\gamma_i^p G \otimes G$, $([x, y] \otimes {}^y z)([y, z] \otimes {}^z x)([z, x] \otimes {}^x y)$ is in the kernel of ι . Again using Lemma 4.2.2 we have

$$\gamma_{i+1}^p G \otimes G / \gamma_{i+2}^p G \cong \Gamma_{i+1}^p G \otimes \Gamma_i^p G.$$

Since ${}^a b = aba^{-1} = [a, b]a$ it follows that ${}^a b = b$ in $\Gamma_2^p G$ and so the image of $([x, y] \otimes {}^y z)([y, z] \otimes {}^z x)([z, x] \otimes {}^x y)$ is the same as the image of $([x, y] \otimes z)([y, z] \otimes x)([z, x] \otimes y)$ under the canonical homomorphism $\gamma_{i+1}^p G \otimes G \longrightarrow \gamma_i^p G / \gamma_{i+1}^p G$. The subgroup of $\Gamma_i^p G \otimes \Gamma_2^p G$ generated by the images of elements $([x, y] \otimes z)([y, z] \otimes x)([z, x] \otimes y)$ for $x, y, z \in \gamma_i^p G$ is precisely the image of j_i . Then just as in Lemma 4.2.7 we have

$$\begin{aligned}
|G \otimes G| &\leq |\Gamma_1^q G \otimes \Gamma_1^q G| |\Gamma_1^q G \otimes \Gamma_2^q G| |\Gamma_1^q G \otimes \Gamma_3^q G| \dots |\Gamma_1^q G \otimes \Gamma_c^q G| / p^a \\
&= |\Gamma_1^q G \otimes \Gamma^q| / p^a.
\end{aligned}$$

It follows that $|G \otimes G| \leq p^{M_p - a}$. ■

4.3.2 Theorem *Suppose G is a group of order p^n and nilpotency class c . Let q*

be 0 or p . Then if G^{ab} is the abelianization of G we have

$$\exp(G \otimes G) \leq \prod_{1 \leq i \leq c} \min(\exp(\Gamma_1^q G), \exp(\Gamma_i^q G)) \leq \exp(G^{ab})^c.$$

Proof: Firstly if A and B are abelian groups acting on each other trivially then $\exp(A \otimes B) \leq \min(\exp(A), \exp(B))$. Considering again the exact sequence of Lemma 4.2.6 we have

$$\begin{aligned} \exp(G \otimes G) &\leq \exp(\Gamma_1^q G \otimes \Gamma_1^q G) \exp(\Gamma_1^q G \otimes \Gamma_2^q G) \dots \exp(\Gamma_1^q G \otimes \Gamma_c^q G) \\ &\leq \prod_{1 \leq i \leq c} \min(\exp(\Gamma_1^q G), \exp(\Gamma_i^q G)) \\ &\leq \exp(G^{ab})^c. \end{aligned} \quad \blacksquare$$

We note here that Theorem 4.3.1 complements Theorem 2 of [19].

4.4 Examples

We illustrate Theorem 1 using the Burnside group, $B = B(3, 3)$, on three generators and exponent three. This group has order 3^7 and its tensor square has order 3^{20} (see [27]). Parts (i) and (ii) of Theorem 1 each give 3^{21} as an upper bound for the order of the tensor square. If x, y , and z generate B then their image in $\Gamma_2^p B \times \Gamma_1^p B$ is isomorphic to $C_3 \times C_3 \times C_3$; the images of $[x, y]$, $[x, z]$, and $[y, z]$ generate $\Gamma_2^p B \cong C_3 \times C_3 \times C_3$, and the image of $[[x, y], z]$ generate $\Gamma_3^p B \cong C_3$. It follows that $M_p = 21$. The image of the map j_2 is found to be one dimensional and the maps j_i are trivial for $i \geq 3$. Thus, Theorem 1 part (iii) gives an upper bound of 3^{20} for the order of the tensor square.

As a second illustration we consider the nonabelian, 3-generator, groups of order

3^4 . There are three such groups and we chose one whose tensor square is of minimal order. One such group is given by the presentation

$$G = \langle x, y, z : x^3 = y^3 = z^9 = [x, z] = [y, z] = e, {}^y x = z^2 x z \rangle .$$

The tensor square of G is of order 3^9 while once again parts (i) and (ii) of Theorem 1 give the same bound 3^{12} . Now the images of x , y and z in $\Gamma_1^p G$ generate $\Gamma_1^p G \cong C_3 \times C_3 \times C_3$, and the image of $[x, y]$ generates $\Gamma_2^p G \cong C_3$. It follows that $M_p = 12$, and since the image of j_2 is of dimension 1 and j_i is trivial for $i \geq 3$, we have $a = 1$. Thus part (iii) gives an upper bound of 3^{11} for the order of $G \otimes G$.

4.5 The p -tensor product

Our aim in this section is to produce a result similar to Theorem 4.3.1 for the p -tensor product $G \otimes^p G$ of Definition 1.3.1. Indeed we can use the work in the last section together with the exact sequence of Theorem 15 in [24] to get the following.

4.5.1 Proposition *Let G be a d -generator group of order p^n and suppose $G^{ab} = p^k$. Then*

$$\text{i) } |G \otimes^p G| \leq p^{dn+k}$$

$$\text{ii) } |G \otimes^p G| \leq p^{M_0+k}$$

$$\text{iii) } |G \otimes^p G| \leq p^{M_p-a+k}$$

Proof: Part of the exact sequence in [24] is

$$G \otimes^{qr} G / rJ(G, qr) \longrightarrow G \otimes^r G \longrightarrow G/G\#^r G \longrightarrow 1$$

where $q \geq 0$, $r \geq 1$ and $J(G, qr)$ is the kernel of the homomorphism $\delta : G \otimes^{qr} G \rightarrow G$, $g \otimes^{qr} h \mapsto [g, h]$, $\{g\} \mapsto g^{qr}$.

Restricting to the case $r = p$ and $q = 0$ we get

$$G \otimes G / pJ(G, 0) \longrightarrow G \otimes^p G \longrightarrow G^{ab},$$

from which it follows that

$$\begin{aligned} |G \otimes^p G| &\leq |G \otimes G / pJ(G, 0)| |G^{ab}| \\ &\leq |G \otimes G| |G^{ab}|. \end{aligned}$$

■

We now use the methods of the last section to improve the bounds given in Proposition 4.5.1 at least for some groups G . Most of the lemmas of the previous section carry over to the p -tensor case. However Lemma 4.2.2 is an exception, for in general $(G/\gamma_{i+1}G) \otimes^p \Gamma_i G \not\cong \Gamma_1 G \otimes^p \Gamma_i G$.

4.5.2 Lemma *Suppose M and N are normal subgroups of a group G , with N a subgroup of M and $M\#^p G \leq N$. Then the image of the canonical homomorphism*

$$N \otimes^p G \xrightarrow{\iota} M \otimes^p G$$

is normal in $M \otimes^p G$. In particular, for each i , $\iota(\gamma_{i+1}^q G \otimes^p G) \trianglelefteq \gamma_i^q G \otimes^p G$.

Proof: By Proposition 1.2.11 there is a crossed N -module $\partial : M \otimes^p G \rightarrow N$.

Let $\tau \in M \otimes^p G$ and suppose ρ is the image under ι of a generator of $N \otimes^p G$.

Then

$$\tau(n \otimes^p g)\tau^{-1} = {}^{\partial\tau}(n \otimes^p g).$$

If $\rho = n \otimes^p g$ we have $\tau\rho\tau^{-1} = {}^{\partial\tau}n \otimes^p {}^{\partial\tau}g \in \iota(N \otimes^p G)$. Similarly if $\rho = \{n\}$ we get $\tau\rho\tau^{-1} = {}^{\partial\tau}\{n\} = \{{}^{\partial\tau}\}$ $\in \iota(N \otimes^p G)$. Taking $N = \gamma_{i+1}^q G$ and $M = \gamma_i^q G$ it follows that $\iota(\gamma_{i+1}^q G \otimes^p G) \leq \gamma_i^q G \otimes^p G$. \blacksquare

4.5.3 Lemma *For each i ,*

$$G/\gamma_{i+1}^q G \otimes^p \Gamma_i^q G \cong \Gamma_1^p G \otimes^p \Gamma_i^q G$$

where groups act on each other through the inherited conjugation action of G .

Proof: The groups $G/\gamma_{i+1}^q G$ and $\Gamma_i^q G$ act on each other trivially and so by Proposition 1.24 [13] we have

$$\begin{aligned} G/\gamma_{i+1}^q G \otimes^p \Gamma_i^q G &\cong \frac{G/\gamma_{i+1}^q G}{G/\gamma_{i+1}^q G \#^p G/\gamma_{i+1}^q G} \otimes_{\mathbb{Z}} \frac{\Gamma_i^q G}{\Gamma_i^q G \#^p \Gamma_i^q G} \\ &\cong \frac{G}{(G \#^p G)\gamma_{i+1}^q G} \otimes_{\mathbb{Z}} \frac{\gamma_i^q G}{(\gamma_i^q G \#^p \gamma_i^q G)\gamma_{i+1}^q G} \end{aligned}$$

where again $\otimes_{\mathbb{Z}}$ denotes the usual tensor product of abelian groups.

Now if $q = p$ we have $\gamma_{i+1}^p \leq G \#^p G$ and $\gamma_i^p G \#^p \gamma_i^p G \leq \gamma_{i+1}^p G$ so that

$$G/\gamma_{i+1}^p \otimes^p \gamma_i^p G/\gamma_{i+1}^p G \cong \Gamma_i^p G \otimes^p \Gamma_i^p G.$$

On the other hand if $q = 0$ then since $\gamma_{i+1} G \leq G \#^p G$ we have

$$G/\gamma_{i+1} G \otimes^p \Gamma_i G \cong \Gamma_1 G \otimes_{\mathbb{Z}} \gamma_i G / (\gamma_i G \#^p \gamma_i G)\gamma_{i+1} G$$

Now $[\gamma_i G, \gamma_i G] \leq \gamma_{i+1} G$ and if $\bar{a} \in \Gamma_1^p G$ and $\bar{b} \in \gamma_i G/\gamma_{i+1} G$ then in $\Gamma_1 G \otimes_{\mathbb{Z}}$

$\gamma_i G / \gamma_{i+1} G$ we have

$$a \otimes_{\mathbb{Z}} b^p = (a \otimes_{\mathbb{Z}} b)^p = a^p \otimes_{\mathbb{Z}} b = 1$$

and so

$$\begin{aligned} G / \gamma_{i+1} G \otimes^p \Gamma_i G &\cong \Gamma_1^p G \otimes_{\mathbb{Z}} \gamma_i G / \gamma_{i+1} G \\ &\cong \Gamma_1^p G \otimes_{\mathbb{Z}} \Gamma_i G \end{aligned} \quad \blacksquare$$

4.5.4 Lemma *Consider the canonical homomorphisms*

$$\begin{aligned} \gamma_{i+1}^q G \otimes^p \gamma_i^q G &\xrightarrow{\alpha} \gamma_i^q G \otimes^p G \\ \gamma_i^q G \otimes^p \gamma_{i+1}^q G &\xrightarrow{\beta} \gamma_i^q G \otimes^p G. \end{aligned}$$

The images of α and β are the same.

Proof: If $N \trianglelefteq G$ then from Propostion 1.16 in [13] we have a crossed module

$$\lambda : N \otimes^p G \twoheadrightarrow N \#^p G$$

such that $\lambda(t) \otimes^p g = t^g t^{-1}$ and $h \otimes^p \lambda(t) = {}^g t t^{-1}$ for all $t \in NG \otimes^p G$ and $g \in G$,

$h \in N$. Taking $N = \gamma_i^q G$ we get

$$\begin{aligned} n \otimes^p g = \lambda(t) \otimes^p g &= t^g t^{-1} \\ &= ({}^g t t^{-1})^{-1} = (g \otimes^p \lambda(t))^{-1} = (g \otimes^p n)^{-1} \end{aligned}$$

for all $n \in \gamma_{i+1}^q G$ and $g \in \gamma_i^q G$. The result follows. ■

4.5.5 Lemma *For each i , the canonical sequence of homomorphisms*

$$\gamma_{i+1}^q G \otimes^p G \xrightarrow{\iota} \gamma_i^q G \otimes^p G \xrightarrow{\pi} (\gamma_i^q G / \gamma_{i+1}^q G) \otimes^p (G / \gamma_{i+1}^q G)$$

is exact.

Proof: By Lemma 4.5.2, $\iota(\gamma_{i+1}^q G \otimes^p G) \leq \gamma_i^q \otimes^q G$ and since $\iota(\gamma_{i+1}^q G \otimes^p G) \leq \ker \pi$, π induces a homomorphism $\tilde{\pi} : (\gamma_i^q G \otimes^p G) / \iota(\gamma_{i+1}^q G \otimes^p G) \longrightarrow (\gamma_i^q G / \gamma_{i+1}^q G) \otimes^p (G / \gamma_{i+1}^q G)$.

We produce the inverse of this map by defining ψ on generators of $(\gamma_i^q G / \gamma_{i+1}^q G) \otimes^p (G / \gamma_{i+1}^q G)$ by

$$\begin{aligned}\psi(a\gamma_{i+1}G \otimes^p b\gamma_{i+1}^qG) &= a \otimes^p b\iota(\gamma_{i+1}^qG \otimes^p G), \\ \psi(\{a\}\gamma_{i+1}G) &= \{a\}\iota(\gamma_{i+1}^qG \otimes^p G)\end{aligned}$$

for all $a \in \gamma_{i+1}^q G$ and $b \in G$.

We need to show that ψ is well defined and to this end suppose $n \in \gamma_{i+1}^q G$. Then

$$\begin{aligned}\psi(an\gamma_{i+1}^qG \otimes^p b\gamma_{i+1}^qG) &= (an \otimes^p b)\iota(\gamma_{i+1}^qG \otimes^p G) \\ &= ({}^a n \otimes^p {}^a b)(a \otimes^p b)\iota(\gamma_{i+1}^qG \otimes^p G) \\ &= (a \otimes^p b)\iota(\gamma_{i+1}^qG \otimes^p G).\end{aligned}$$

In the same way

$$\begin{aligned}\psi(a\gamma_{i+1}^qG \otimes^p bn\gamma_{i+1}^qG) &= a \otimes^p bn\iota(\gamma_{i+1}^qG \otimes^p G) \\ &= (a \otimes^p b)({}^b a \otimes^p {}^b n)\iota(\gamma_{i+1}^qG \otimes^p G) \\ &= (a \otimes^p b)\iota(\gamma_{i+1}^qG \otimes^p G)\end{aligned}$$

since by Lemma 4.5.4 the image of $\gamma_{i+1}^q G \otimes^p \gamma_i^q G$ and $\gamma_i^q G \otimes^p \gamma_{i+1}^q G$ coincide in $\gamma_i^q G \otimes^p G$. Finally we have

$$\begin{aligned}\psi(\{an\gamma_{i+1}^qG\}) &= \{an\}\iota(\gamma_{i+1}^qG \otimes^p G) \\ &= \{a\} \prod_{k=1}^{p-1} (a^{-1} \otimes^p (a^{1-p+k} n)^k) \{n\} \iota(\gamma_{i+1}^qG \otimes^p G) \\ &= \{a\}\iota(\gamma_{i+1}^qG \otimes^p G)\end{aligned}$$

since $a^{-1} \otimes^q (a^{1-p+k} n)^k \in \iota(\gamma_{i+1}^q G \otimes^p G)$ for all k .

It follows that

$$\Gamma_i^q G \otimes^p G / \gamma_{i+1}^q \cong (\gamma_i^q G \otimes^p G) / \iota(\gamma_{i+1}^q G \otimes^p G)$$

and so the given sequence is exact ■

4.5.6 Lemma *Suppose G is nilpotent with $\gamma_{c+1} = 1$. Then*

$$|G \otimes^p G| \leq w |\Gamma_1^q \otimes \Gamma^q|,$$

where $w = |\Gamma_1^q G \otimes^p \Gamma_1^q G| / |\Gamma_1^q G \otimes \Gamma_1^q G|$.

Proof: By repeated application of Lemma 4.5.5 we have

$$\begin{aligned} |G \otimes^p G| &\leq |\Gamma_1^q G \otimes^p \Gamma_1^q G| |G \otimes^p \gamma_2^q G| \\ &\leq |\Gamma_1^q G \otimes^p \Gamma_1^q G| |(G/\gamma_3^q G) \otimes^p \Gamma_2^q G| |(G/\gamma_4^q G) \otimes^p \Gamma_3^q G| \dots \\ &\quad \dots |(G/\gamma_c^q G) \otimes^p \Gamma_{c-1}^q G| \\ &= |\Gamma_1^q G \otimes^p \Gamma_1^q G| |\Gamma_3^p G \otimes^p \Gamma_2^q G| |\Gamma_4^p G \otimes^p \Gamma_3^q G| \dots |\Gamma_c^p G \otimes^p \Gamma_{c-1}^q G|. \end{aligned}$$

Now if $q = p$ by Propostion 1.24 in [13] we have for each i

$$\Gamma_{i+1}^p G \otimes^p \Gamma_i^q G \cong \Gamma_1^p G \otimes_{\mathbb{Z}} \Gamma_i^q G \cong \Gamma_1^p G \otimes \Gamma_i^q G.$$

On the other hand if $q = 0$ then

$$|\Gamma_{i+1}^p G \otimes^p \Gamma_i G| = |\Gamma_1^p G \otimes_{\mathbb{Z}} \Gamma_i G| \leq |\Gamma_1 G \otimes \Gamma_i^q G|.$$

In either case

$$\begin{aligned} |G \otimes^p G| &\leq w |\Gamma_1^q G \otimes \Gamma_1^q G| |\Gamma_1^q G \otimes \Gamma_2^q G| |\Gamma_1^q G \otimes \Gamma_3^q G| \dots |\Gamma_1^q G \otimes \Gamma_{c-1}^q G| \\ &= w |\Gamma_1^q G \otimes (\Gamma_1^q G \times \Gamma_2^q G \times \dots \times \Gamma_{c-1}^q G)| \\ &= w |\Gamma_1^q G \otimes \Gamma^q G|. \end{aligned}$$
■

We are now in a position to state the following results.

4.5.7 Theorem *Suppose G is d -generator group of order p^n . Then*

$$\text{i) } |G \otimes^p G| \leq p^{dn+w},$$

$$\text{ii) } |G \otimes^p G| \leq p^{M_0+w},$$

$$\text{iii) } |G \otimes^p G| \leq p^{M_p+w-a}$$

where $w = |\Gamma_1^q G \otimes^p \Gamma_1^q G| / |\Gamma_1^q G \otimes \Gamma_1^q G|$.

Proof: The proof of this is identical to that of Theorem 4.3.1. ■

4.5.8 Example Let $G = \mathbb{Z}_2 \oplus \mathbb{Z}_2$. In this case $|G^{ab}| = |G| = 2^2$, so k in Proposition 4.5.1 is 2. However because $|G \otimes^2 G| = 32$ and $|G \otimes G| = 16$ we have w in the preceding theorem is only 2.

4.5.9 Theorem *Suppose G is a group of order p^n and nilpotency class c . Then*

$$\exp(G \otimes^p G) \leq p^c.$$

Proof: As before we have

$$\begin{aligned} \exp(G \otimes^p G) &\leq \exp(\Gamma_1^p G \otimes \Gamma_1 G) \exp(\Gamma_1^p G \otimes \Gamma_2 G) \dots \exp(\Gamma_1^p G \otimes \Gamma_c G) \\ &= \prod_{1 \leq i \leq c} \min(\exp(\Gamma_1^p G), \exp(\Gamma_i G)) \\ &= p^c. \end{aligned}$$
■

4.6 Placing bounds on the order of the kernel J

Suppose G is a group of order p^n and consider again the exact sequence

$$J \rightarrow G \otimes G \twoheadrightarrow [G, G].$$

By placing restrictions on the order of J we determine all possible G that can arise. In particular we enumerate all G for which $|J| = p^{n^2-l}$ for small l .

The following theorem is an analogue of a result of Berkovich [4] on the Schur multiplier.

4.6.1 Theorem *Let G be a group of order p^n and suppose $|J| = p^{n^2-l}$. Then*

i) $l = 0$ if and only if G is elementary abelian,

ii) $l \neq 1$ or 3 ,

iii) $l = 2$ if and only if $G \cong C_{p^2}$,

iv) $l = 4$ if and only if G is isomorphic to one of $C_{p^2} \times C_p \times C_p$, $C_{p^2} \times C_p$, or $p = 2$ and $G \cong \langle a, b, c \mid b^p = a, c^p = a, a^p = [a, b] = [a, c] = 1, [b, c] = a \rangle$,

v) if $l = 5$ then either $p > 1000$ or $p = 2$ and $G \cong \langle a, b, c \mid a^2 = b^2 = c^2 = [a, b] = [a, c] = 1, [b, c] = a \rangle$,

where C_m denotes the cyclic group of order m .

Proof: (i) If $l = 0$ then $p^{n^2} = |J| \leq |G \otimes G| \leq p^{n^2}$ from which it follows that by Theorem 4.3.1 that $|G \otimes G| = p^{n^2} \leq p^{nd}$ where d is the number of generators of G . Therefore $d(G) = n$ and G is elementary abelian.

Conversely if G is elementary abelian then $J = G \otimes G$ and so $|J| = p^{n^2}$.

(ii) Suppose now that $|J| = p^{n^2-1}$, where $n \geq 1$.

If $|[G, G]| = 1$ then $|G \otimes G| = |J| |[G, G]| = p^{n^2-1} \leq p^{nd}$. It follows that $n(n-d) \leq 1$ and since $n \geq 1$ we have $n = 1$ and G is cyclic of order p . But in that case $|G \otimes G| = p$ and $|J| = p \neq p^{1^2-1}$ a contradiction.

If $|[G, G]| = p$ we have $|G \otimes G| = p^{n^2}$ and so by part (i) G is elementary abelian a contradiction.

The order of $[G, G]$ can not be greater than p since otherwise we would have $|G \otimes G| > p^{n^2}$ which contradicts Theorem 4.3.1 .

Suppose $|J| = p^{n^2-3}$, where $n \geq 2$.

If $|[G, G]| = 1$ then $|G \otimes G| = p^{n^2-3}$ and $n(n-d) \leq 3$. There are four possibilities:

$n = 3, d = 3$ which means G is elementary abelian which contradicts (i),

$n = 3, d = 2$ in which case $G \cong C_{p^2} \times C_p$ and $|G \otimes G| = p^5 \neq p^{n^2-3}$,

$n = 2, d = 2$ then again G is elementary abelian a contradiction, or

$n = 2, d = 1$ which means $G \cong C_{p^2}$ and $|G \otimes G| = p^2 \neq p^{n^2-3}$.

If $|[G, G]| = p$ then $n(n-d) \leq 2$ and $n = 2$. Both the cases $d = 2$ or $d = 1$ mean G is abelian a contradiction.

Similarly $|[G, G]| = p^2$ means $n = 1$ and G is abelian.

(iii) Suppose $|J| = p^{n^2-2}$, where $n \geq 2$.

If $|[G, G]| = 1$ then $|G \otimes G| = p^{n^2-2}$ and $n(n-d) \leq 2$. This means that $n = 2$. If $d = 2$ then $G \cong C_p \times C_p$ and $|G \otimes G| = p^4 \neq p^{n^2-2}$ a contradiction. On the other hand if $d = 1$ we have $G \cong C_{p^2}$ and $|G \otimes G| = p^2 = p^{n^2-2}$. Furthermore in this case we have $|J| = |G \otimes G| = p^{n^2-2}$ as required.

If $|[G, G]| = p$ then $|G \otimes G| = p^{n^2-1} \leq p^{nd}$ implies $n(n-d) \leq 1$. This means G is cyclic a contradiction.

(iv) Suppose $|J| = p^{n^2-4}$, where $n \geq 2$.

If $|[G, G]| = 1$ then $n(n-d) \leq 4$. Again we dismiss the elementary abelian case ($n = 4, d = 4$) and if $n = 4, d = 3$, $G \cong C_{p^2} \times C_p \times C_p$ and $|G \otimes G| = p^{10} = p^{n^2-4}$.

The other possibilities are

$n = 3, d = 3$ giving G elementary abelian a contradiction,

$n = 3, d = 2$ giving $G \cong C_{p^2} \times C_p$ and $|G \otimes G| = p^5 = p^{n^2-4}$,

$n = 2, d = 2$ giving G elementary abelian a contradiction,

$n = 2, d = 1$ giving G cyclic and $|G \otimes G| = p^2 \neq p^{n^2-4}$.

If $|[G, G]| = p$ then $n(n-d) \leq 3$. Dismissing the elementary abelian case ($n = 3, d = 3$) we are left with the cases $n = 3, d = 2$ and $n = 2$. The $n = 2$ cases imply G is abelian and so can be ignored.

It is shown for example in [49] that there are only two isomorphism types for nonabelian groups of order p^3 , and that both are of nilpotency class 2. The two

isomorphism types are

$$\langle a, b, c \mid a^p = b^p = c^p = [a, b] = [a, c] = 1, [b, c] = a \rangle$$

and

$$\langle a, b, c \mid b^p = a, c^p = a, a^p = [a, b] = [a, c] = 1, [b, c] = a \rangle$$

where in both cases a generates both the centre and commutator subgroup of G .

Suppose p is odd. By Theorem 4.3 of [2] $G \otimes G$ is elementary abelian of order p^6 for the first isomorphism type given here and by Theorem 4.2 of [2] $G \otimes G$ is elementary abelian of order p^4 for the second isomorphism type. Only the first of these gives $|G \otimes G| = p^{n^2-3}$ and in that case again by Theorem 4.3 of [2] we have $|J| = p^4 \neq p^{n^2-4}$.

This leaves the case when $p = 2$. The GAP program p3.g in appendix C computes $G \otimes G$ and J in this case. For the first isomorphism type above $G \otimes G \cong C_{p^2} \times C_p \times C_p \times C_p$ and for the second we have $G \otimes G \cong C_{p^2} \times C_p \times C_p \times C_p \times C_p$. Only the second has the required order, p^{n^2-3} , and in this case $|J| = p^{n^2-4}$.

The cases $|[G, G]| = p^2$ and $|[G, G]| = p^3$ both imply G is abelian.

(v) Suppose $|J| = p^{n^2-5}$ where $n \geq 3$.

We can only partially solve this case. We begin as usual with the abelian groups.

If $|[G, G]| = 1$ then $n(n-d) \leq 5$. Ignoring the elementary abelian cases ($n = d$) we are left with

$$n = 5, d = 4 \text{ so that } G \cong C_{p^2} \times C_p \times C_p \times C_p \text{ and } |G \otimes G| = p^{17} \neq p^{n^2-5},$$

$$n = 4, d = 3 \text{ so that } G \cong C_{p^2} \times C_p \times C_p \text{ and } |G \otimes G| = p^{10} \neq p^{n^2-5},$$

$n = 3, d = 2$ so that $G \cong C_{p^2} \times C_p$ and $|G \otimes G| = p^5 \neq p^{n^2-5}$.

If $|[G, G]| = p$ then $n(n - d) \leq 4$. We deal first with the case $n = 4, d = 3$. It is shown in [12] page 145 that there are 15 isomorphism types for groups of order p^4 where p is an odd prime. Of these only three are 3 generator nonabelian groups. These isomorphism types are

$$\begin{aligned} A = \langle a, b, c, d \mid & a^p = b^p = c^p = d^p = 1, \\ & [c, a] = [c, b] = [d, a] = [d, b] = [d, c] = 1, \\ & [b^{-1}, a^{-1}] = d \rangle \end{aligned}$$

$$\begin{aligned} B = \langle a, b, c, d \mid & b^p = c^p = d^p = 1, \\ & [c, a] = [c, b] = [d, a] = [d, b] = [d, c] = 1, \\ & a^p = d, [b^{-1}, a^{-1}] = d \rangle \end{aligned}$$

$$\begin{aligned} C = \langle a, b, c, d \mid & a^p = b^p = d^p = 1, \\ & [c, a] = [c, b] = [d, a] = [d, b] = [d, c] = 1, \\ & c^p = d, [b^{-1}, a^{-1}] = d \rangle \end{aligned}$$

Programs p4.g and p42.g in Appendix C compute $G \otimes G$ for each of these groups for $p < 1000$. If $p < 1000$ is odd then $G \otimes G$ is elementary abelian of order p^{11}, p^9 and p^9 for the isomorphism types A, B and C respectively. When $p = 2$, $G \otimes G$ is not necessarily elementary abelian and has order p^{10}, p^{11} and p^9 respectively. In no case is this p^{n^2-4} .

If $n = 3, d = 2$ we can again use the results of [2] when p is odd. We then have $|G \otimes G| = p^4$ or $|G \otimes G| = p^6$, neither of which is p^{n^2-4} . When $p = 2$ only the first isomorphism type gives $|G \otimes G| = p^5 = p^{n^2-4}$ and in this case we have $|J| = p^{n^2-5}$ as required. If $|[G, G]| = p^2$ we have $n(n - d) \leq 3$ and $n = 3$. But no group of order p^3 has $|[G, G]| = p^2$. ■

Chapter 5

The computation of the Frattini subgroup of prime–power groups

5.1 Introduction

In [11] computer methods were used to improve the results of [45] on determining $[G, G]$ when the group $Q = G/Z(G)$ is given. In this chapter we extend these techniques to the variety $G/[G, G]G^q$. We tackle the following problem. Given a finite presentation for a group Q , determine all possible $[G, G]G^q$, where $Q = G/Z(G, q)$, q is an integer and $Z(G, q)$ is defined as $\{g \in Z(G) : g^q = 1\}$. We note that when q is a prime and G a q -power group then $[G, G]G^q$ is the Frattini subgroup of G . An algorithm is given for solving this problem and implemented in the case where q is a prime and $|Q| < 32$.

5.2 Definitions and Notation

5.2.1 Definition Let Q be a group and q a non–negative integer. We define the *epicentre of Q mod q* as

$$Z^\wedge(Q, q) = \{g \in Q \mid \langle g \rangle = 1 \in Q \wedge^q Q \text{ and } g \wedge x = 1 \in Q \wedge^q Q \text{ for all } x \in Q\}.$$

As mentioned above, we shall denote by $Z(Q, q)$ those elements in the centre of Q whose order divides q .

Extending terminology of Hall and Senior [32] we say Q is *capable mod q* if there is a group G with $Q \cong G/Z(G, q)$.

5.2.2 Lemma

- i) Suppose F/S is a presentation for the group G and $Q \cong G/Z(G, q)$. Then there exists a presentation F/R of Q and a homomorphism $\rho : F/[R, F]R^q \rightarrow G$.
- ii) The homomorphism ρ defines a projective q -crossed Q module where the action of Q on $F/[R, F]R^q$ is defined via conjugation in F .
- iii) Let $R \hookrightarrow F$ be a free presentation of Q . Then

$$Q \wedge^q Q \cong [D, D]D^q,$$

where $D = F/[F, R]R^q$.

Proof: (i) Let ϕ complete the following commutative diagram of homomorphisms and take R to be $\ker \phi$. This provides the presentation F/R of Q .

$$\begin{array}{ccc}
 R & & S \\
 & \searrow & \downarrow \\
 & & F \\
 & & \downarrow \psi \\
 & & G \\
 & \nearrow \phi & \longrightarrow Q \cong G/Z(G, q)
 \end{array} \tag{5.1}$$

Let $r \in R$. Since $\phi(r) = 1 \in Q$, $\psi(r) \in Z(G, q)$. Then $\psi(r^q) = 1$ and $\psi([r, f]) = 1$ in G for all $f \in F$ and so it follows that $[R, F]R^q \leq S$. The homomorphism ψ then gives rise to the homomorphism $\rho : F/[R, F]R^q \rightarrow G$.

(ii) The q -crossed module axioms follow directly from the definition of the conjugation action, that is if $fR \in Q$ and $\bar{d} = d[R, F]R^q \in F/[R, F]R^q$ then

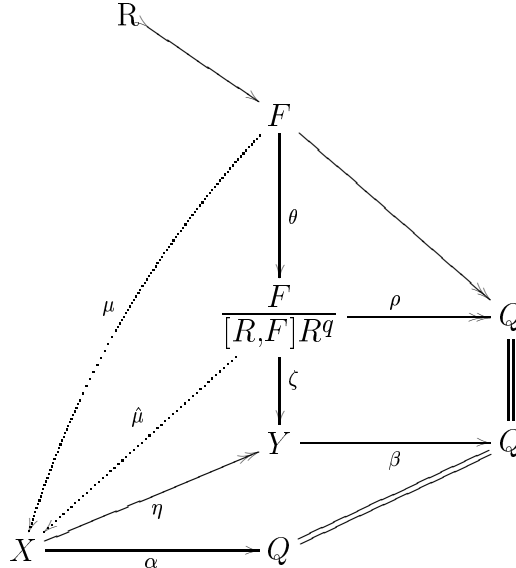
$$\begin{aligned} \rho({}^fR\bar{d}) &= \rho(fd f^{-1}[R, F]R^q) \\ &= fRdRf^{-1}R \\ &= fR\rho(\bar{d})f^{-1}R \end{aligned}$$

Similarly if $\bar{d}' = d'[R, F]R^q \in F/[R, F]R^q$ then

$$\begin{aligned} \rho(\bar{d})\bar{d}' &= dR\bar{d}' \\ &= dd'd^{-1}[R, F]R^q \\ &= \bar{d}\bar{d}'\bar{d}^{-1}. \end{aligned}$$

The kernel of ρ is $R/[R, F]R^q$ and so $r^q = 1$ for $r \in \ker \rho$. It remains then to show that ρ is projective as a q -crossed Q module.

Suppose $\alpha : X \rightarrow Q$ and $\beta : Y \rightarrow Q$ are q -crossed Q modules and $\zeta : \rho \rightarrow \beta$ and $\eta : \alpha \rightarrow \beta$ are morphisms of q -crossed Q modules as in the diagram.



Using the forgetful functor ζ , θ and η provide homomorphisms of groups which we also denote by ζ , θ and η . Now in the category of groups F is free and so projective.

It follows that there exists a homomorphism $\mu : F \longrightarrow X$ such that $\eta \circ \mu = \zeta \circ \theta$.

We now show that this homomorphism factors to give $\hat{\mu} : F/[R, F]R^q \longrightarrow X$.

Suppose $r \in R$. Then the image of r is the identity in Q from which it follows that $\mu(r) = \ker \alpha$. Since α is a q -crossed Q module, $r^q = 1 \in X$. Also if $f \in F$ then

$$\begin{aligned} \mu([r, f]) &= [\mu(r), \mu(f)] \\ &= \alpha(\mu(r)) \mu(f) \mu(f)^{-1} \\ &= \mu(f) \mu(f)^{-1} \end{aligned}$$

It follows that both R^q and $[R, F]$ are trivial in X and so $[R, F]R^q \leq \ker \mu$. The homomorphism $\hat{\mu} : F/[R, F]R^q \longrightarrow X$ therefore exists and satisfies $\hat{\mu} \circ \theta = \mu$. By construction this map provides the morphism of q -crossed Q modules.

(iii) Proposition 5 of [24] states that if $\delta : C \longrightarrow G'$ is a projective q -crossed G' -module with $\delta C = G$, then

$$[C, C]C^q \cong G \wedge^q G$$

from which the result follows. ■

We now give a computational criterion to decide when a group is capable mod q . We note that a similar result is found in [24] but when $q > 0$ the definitions of “ q -capable” and “ Z_q^\wedge ” of that paper differ from the definitions of “capable mod q ” and “ $Z^\wedge(\cdot, q)$ ” given here.

5.2.3 Proposition *A group Q is capable mod q if and only if $Z^\wedge(Q, q) = 1$.*

Proof: Let F/R be a presentation for Q as in Lemma 5.2.2 and denote $F/[R, F]R^q$

by D and $R/[R, F]R^q$ by \bar{R} . The exact sequence

$$\frac{R}{[R, F]R^q} \twoheadrightarrow \frac{F}{[R, F]R^q} \xrightarrow{\rho} Q$$

gives $Q \cong D/\bar{R}$. Suppose $Z^\wedge(Q, q) = 1$ and $x \in Z(D, q)$. Because of Lemma 5.2.2 (iii), $\rho(x) \in Z^\wedge(Q, q)$ and so $x \in \bar{R}$. Hence $\bar{R} = Z(D, q)$ and Q is capable mod q .

Conversely, suppose Q is capable mod q with $Q \cong G/Z(G, q)$. By Lemma 5.2.2 (i) we may choose presentations F/R for Q and F/S for G with $[R, F]R^q \leq S$.

Let $fR \in Z^\wedge(Q, q)$. Then $f[R, F]R^q \in Z(D, q)$. But then Diagram 5.1 implies that $fS \in Z(G, Q)$ and so $fR = 1$ in Q . ■

For all primes p , Table 5.1 lists all p -groups Q , $|Q| < 32$ which are capable mod p . Our computer method for producing the list is based on Proposition 5.2.3.

Lemma 5.2.2 concerns the existence of certain presentations. We now start with a fixed presentation F/R for a group Q . We suppose this presentation to be finite and Q to be capable mod q with $Q \cong G/Z(G, q)$. Letting D denote $F/[R, F]R^q$, we have

$$\frac{R}{[R, F]R^q} \twoheadrightarrow D \cong \frac{F}{[R, F]R^q} \twoheadrightarrow Q \tag{5.2}$$

The group D is easily presented. Indeed starting with the free group F we may take as relators all commutators of the form $[r, f]$ where r is a generator of R and f a generator of F together with all r^q . Now $R/[R, F]R^q$ is finitely presented since it is abelian and R is finitely generated. Also $R/[R, F]R^q$ is of finite exponent and so itself is finite. The short exact sequence 5.2 implies that D is finite.

Proposition 7 of [24] gives the exact sequence

$$Z(G, q) \wedge^q G \longrightarrow G \wedge^q G \xrightarrow{\pi} Q \wedge^q Q$$

There is also a natural homomorphism $\eta : G \wedge^q G \rightarrow [G, G]G^q$ defined on generators

by $g \wedge^q h \mapsto [g, h]$ and $\{g\} \mapsto g^q$.

$$\begin{array}{ccc} Z(G, q) \wedge^q G & \longrightarrow & G \wedge^q G \xrightarrow{\pi} Q \wedge^q Q \\ & & \downarrow \eta \quad \nearrow \delta \\ & & [G, G]G^q \end{array}$$

Since $\ker \pi \leq \ker \eta$ we have a homomorphism $\delta : [D, D]D^q \rightarrow [G, G]G^q$. We seek an extension of δ , from D into G .

Because of the homomorphism $G \rightarrow Q$ and the fact that F is free and so projective, there exists a homomorphism $\psi : F \rightarrow G$. As in the proof of Lemma 1 (i) it can be shown that $[F, R]R^q \leq \ker(\psi)$, so ψ factors to give $\tilde{\psi} : D \rightarrow G$. We note however that unlike the corresponding ψ in Lemma 5.2.2, ψ here is not necessarily onto. In particular since we start with the presentation F/R of Q the rank of F may not be sufficient to guarantee that ψ is onto G . Instead we investigate the relationship between G and the image of $\tilde{\psi}$ in G . This is the subject of our next lemma. The situation is summarized in the following diagram

$$\begin{array}{ccccc} [R, F]R^q & \longrightarrow & R & \longrightarrow & \frac{R}{[R, F]R^q} \\ & \searrow & \searrow & & \downarrow \\ & & [R, F]R^q & \longrightarrow & \frac{F}{[R, F]R^q} \\ & & \downarrow \psi & \searrow \phi & \downarrow \\ Z(G, q) & \longrightarrow & G & \longrightarrow & Q \end{array}$$

5.2.4 Lemma

- i) $G = \text{Im}(\tilde{\psi})Z(G, q)$
- ii) $[G, G]G^q \cong [\text{Im}(\tilde{\psi}), \text{Im}(\tilde{\psi})]\text{Im}(\tilde{\psi})^q$

iii) $Q \cong \text{Im}(\tilde{\psi})/Z(\text{Im}(\tilde{\psi}), q)$

Proof: (i) Let $x \in G$. Then $xZ(G, q) \in Q$ and so there exists $f \in F$ such that $\rho(f[R, F]R^q) = xZ(G, q)$. But then $x = \tilde{\psi}(f[R, F]R^q)s$ for some $s \in Z(G, q)$.

(ii) Suppose $x, y \in G$. Then by part (i), $x = x'a$ and $y = y'b$ for some $x', y' \in \text{Im}(\tilde{\psi})$ and $a, b \in Z(G, q)$. Then $[x, y] = [x'a, y'b] = x'ay'b(x'a)^{-1}(y'b)^{-1} = [x', y']$ and $x^q = (x'a)^q = x'^q a^q = x'^q$. The result follows.

(iii) Suppose $x \in G$ and $y \in Z(\text{Im}(\tilde{\psi}), q)$. Then by part (i) $x = x'a$ for some $x' \in \text{Im}(\tilde{\psi})$ and $a \in Z(G, q)$. In this case we have $xy = x'ay = x'ya = yx'a = yx$ so that y is central in G and so $Z(\text{Im}(\tilde{\psi}), q) \leq Z(G, q)$. Then

$$\begin{aligned} \text{Im}(\tilde{\psi})/Z(\text{Im}(\tilde{\psi}), q) &= \text{Im}(\tilde{\psi})/(\text{Im}(\tilde{\psi}) \cap Z(G, q))Z(\text{Im}(\tilde{\psi}), q) \\ &\cong Z(G, q)\text{Im}(\tilde{\psi})/Z(G, q)Z(\text{Im}(\tilde{\psi}), q) \\ &= G/Z(G, q) \cong Q. \quad \blacksquare \end{aligned}$$

Parts (ii) and (iii) of the preceding lemma allows us to state the following.

5.2.5 Theorem *Let F/R be a presentation of Q . If Q is capable mod q , then all the possible $[G, G]G^q$ where $Q \cong G/Z(G, q)$ are of the form $[D, D]D^q/K$ for some $K = L \cap [D, D]D^q$ and L some subgroup of $Z(D, q)$ with $D/L/Z(D/L, q) \cong Q$.*

Proof: Let L be the kernel of $\tilde{\psi}$ so that $\text{Im}(\tilde{\psi}) \cong D/L$. By part (ii) of Lemma 5.2.4 we have

$$\begin{aligned} [G, G]G^q &\cong [\text{Im}(\tilde{\psi}), \text{Im}(\tilde{\psi})]\text{Im}(\tilde{\psi})^q \\ &\cong [D/L, D/L](D/L)^q \\ &\cong [D, D]D^q/(L \cap [D, D]D^q) \end{aligned}$$

Part (iii) of Lemma 5.2.4 shows that $Q \cong D/L/Z(D/L, q)$ so that all that remains

to show is that $L \leq Z(D, q)$. Let $l \in L$. Then $\tilde{\psi}(l) = 1 \in G$ and so $\rho(l) = 1 \in Q$. But then $l \in \bar{R} \leq Z(D, q)$. ■

Given Q , Theorem 5.2.5 allows the enumeration of all possible $[G, G]G^q$. As it stands, however, this would entail the calculation of all $L \leq Z(D, q)$ with $K = L \cap [D, D]D^q$. For large $Z(D, q)$ this can prove computationally expensive. We aim instead to rewrite Theorem 5.2.5 placing restrictions directly on the subgroup K . In this the following notation will be helpful.

For $A \leq D$, let $\tilde{A} = \{d \in D \mid d^q \in A \text{ and } [d, x] \in A \text{ for all } x \in D\}$.

5.2.6 Lemma

- i) If $A = B \cap [D, D]D^q$ then $\tilde{A} = \tilde{B}$
- ii) $Q \cong D/A/Z(D/A, q)$ if and only if $\tilde{A} \cong Z(D, q)$.

Proof: (i) Let $A = B \cap [D, D]D^q$ and $a \in \tilde{A}$. Then $a^q \in A$ and $[a, x] \in A$ for all $x \in D$. But $A \leq B$ so $a \in \tilde{B}$.

Conversely suppose $b \in \tilde{B}$. Then $b^q \in B$ and $[b, x] \in B$ for all $x \in D$. But both b^q and $[b, x]$ are in $[D, D]D^q$ and so are in A . It follows that $b \in \tilde{A}$.

(ii) $Z(D/A, q) \cong \{d \in D \mid d^q \in A \text{ and } [d, x] \in A \text{ for all } x \in D\}/A = \tilde{A}/A$. Now $D/A/Z(D/A, q) \cong D/\tilde{A} = D/Z(D, q)$ if and only if $Z(D, q) = \tilde{A}$. ■

We may now reformulate Theorem 5.2.5.

5.2.7 Theorem *Let F/R be a presentation of Q . If Q is capable mod q , then all the possible $[G, G]G^q$ where $Q \cong G/Z(G, q)$ are of the form $[D, D]D^q/K$ for some $K \leq Z(D, q) \cap [D, D]D^q$ and $\tilde{K} = Z(D, q)$.*

Proof: This follows immediately from Theorem 5.2.5 and Lemma 5.2.6. ■

We are now in a position to give an algorithm for determining all possible $[G, G]G^q$ from the finite presentation F/R of Q . However, when $q = 0$ the group D given above is not necessarily finite. In this case we may use the construction given in [11], taking $D = F/[R, F]\langle B \rangle$ where B is any one to one lift to F of a basis for the image of R in $F/[R, F]$. In practice we take B to be a one to one lift to F of a convenient generating set for the image of R in $F/[R, F]$.

Function:	Frattini														
Dependencies:	<ul style="list-style-type: none"> • None 														
Input:	<table border="1"> <thead> <tr> <th>Object</th> <th>Type</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>F</td> <td>free group</td> <td>The free group F is generated by the set \bar{F}.</td> </tr> <tr> <td>\bar{R}</td> <td>List of words</td> <td>The words in \bar{R} are written in the generators of F, so that \bar{F}/\bar{R} is a presentation for the group Q.</td> </tr> <tr> <td>q</td> <td>integer</td> <td>An integer typically 0 or a prime when Q is a prime-power group.</td> </tr> </tbody> </table>	Object	Type	Notes	F	free group	The free group F is generated by the set \bar{F} .	\bar{R}	List of words	The words in \bar{R} are written in the generators of F , so that \bar{F}/\bar{R} is a presentation for the group Q .	q	integer	An integer typically 0 or a prime when Q is a prime-power group.		
Object	Type	Notes													
F	free group	The free group F is generated by the set \bar{F} .													
\bar{R}	List of words	The words in \bar{R} are written in the generators of F , so that \bar{F}/\bar{R} is a presentation for the group Q .													
q	integer	An integer typically 0 or a prime when Q is a prime-power group.													
Method:	<ul style="list-style-type: none"> • Create a list <i>CommPowerList</i> consisting of all commutators $[r, f]$ where $r \in \bar{R}$ and $f \in \bar{F}$. • If $q = 0$ then compute the group $F/([R, \bar{F}])$ and its subgroup $\langle \bar{R} \rangle / ([R, \bar{F}])$. Since the images of \bar{R} form a generating set for this last group we take B to be these images mapped into F. Append B to the list <i>CommPowerList</i>. • If $q \neq 0$ then append to <i>CommPowerList</i> words of the form r^q where $r \in \bar{R}$. • Compute the group $D = F/\langle \text{CommPowerList} \rangle$. • Create an empty output list <i>OutList</i> • For each subgroup $L \leq Z(D, q) \cap [D, D]D^q$ compute \tilde{L}. • If $\tilde{L} = Z(D, q)$ then append $[D, D]D^q/L$ to <i>OutList</i> 														
Return:	<i>OutList</i>														

Figure 1 Algorithm for computing $[G, G]G^q$ from presentation for Q

This algorithm has been implemented in **GAP** [54] and used to list all possible $[G, G]G^p$ for Q a p -group of order less than 32 and Q capable mod p . Table 5.1 lists all occurring $[G, G]G^q$. This table is based in part on programs pcomm2.g, pcomm3.g, pcomm5.g and pcomm7.g in Appendix D.

In the case where $Q \cong \mathbb{Z}_2^4$, $[D, D]D^q \cong \mathbb{Z}_2^{10}$ and $Z(D, q) \cap [D, D]D^q \cong \mathbb{Z}^6$. Instead of

enumerating all the subgroups of $Z(D, q) \cap [D, D]D^q \cong \mathbb{Z}^6$ a different approach was taken. Since $[D, D]D^q$ is elementary abelian the isomorphism type of $[D, D]D^q/L$ is determined completely by $|L|$. If $|\tilde{L}| = |Z(G, q)|$ for a given subgroup L then we know that $[D, D]D^q/L$ occurs as $[G, G]G^q$ for some G . Any other subgroup L' of $Z(D, q) \cap [D, D]D^q$ of the same order as L will give $[D, D]D^q/L' \cong [D, D]D^q/L$ so that once a subgroup L is found such that $|\tilde{L}| = |Z(G, q)|$ then no further subgroups of the same order need be checked. This is the basis of Program *pcomm2.g* in appendix D where the GAP's internal pseudo-random generator is used to generate random subgroups of $Z(D, q) \cap [D, D]D^q$ of a given order and then check if $|\tilde{L}| = |Z(G, q)|$. Once such a group is found the next size is checked. We note that this strategy depends heavily on the fact that the density of subgroups L in $Z(D, q) \cap [D, D]D^q$ of a given order with $|\tilde{L}| = |Z(G, q)|$ is high.

Table 5.1 : All prime-power groups Q , $|Q| \leq 32$ and Q capable mod q together with all occurring $[G, G]G^q$ where $Q \cong G/Z(G, q)$.

Name	q	$ Q $	Defining Relations	Possible $[G, G]G^q$
\mathbb{Z}_2	2	2		\mathbb{Z}_2
\mathbb{Z}_4	2	4		\mathbb{Z}_4
V_4	2	4		\mathbb{Z}_2^n for $1 \leq n \leq 3$
\mathbb{Z}_8	2	8		\mathbb{Z}_8
$\mathbb{Z}_2 \oplus \mathbb{Z}_4$	2	8		$\mathbb{Z}_2^n \oplus \mathbb{Z}_4$ for $0 \leq n \leq 2$
D_4	2	8	$a^2 = b^4 = (ab)^2 = 1$	$\mathbb{Z}_2^n \oplus \mathbb{Z}_4$ for $0 \leq n \leq 2$
\mathbb{Z}_2^3	2	8		\mathbb{Z}_2^n for $1 \leq n \leq 6$
\mathbb{Z}_{16}	2	16		\mathbb{Z}_{16}
$\mathbb{Z}_4 \oplus \mathbb{Z}_4$	2	16		$\mathbb{Z}_4 \oplus \mathbb{Z}_4$ $\mathbb{Z}_2 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_4$
	2	16	$a^4 = b^4 = (ab)^4 = (a^{-1}b)^2 = 1$	\mathbb{Z}_2^3 $\mathbb{Z}_2 \oplus \mathbb{Z}_4$ \mathbb{Z}_2^4 $\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_4$ $\mathbb{Z}_4 \oplus \mathbb{Z}_4$ $\mathbb{Z}_2^3 \oplus \mathbb{Z}_4$ $\mathbb{Z}_2 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_4$ $\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_4$
	2	16	$a^4 = b^4 = a^{-1}bab = 1$	$\mathbb{Z}_2 \oplus \mathbb{Z}_4$ $\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_4$ $\mathbb{Z}_4 \oplus \mathbb{Z}_4$ $\mathbb{Z}_2 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_4$
$\mathbb{Z}_2 \oplus \mathbb{Z}_8$	2	16		$\mathbb{Z}_2 \oplus \mathbb{Z}_8$ $\mathbb{Z}_2 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_4$ $\mathbb{Z}_2^3 \oplus \mathbb{Z}_8$
D_8	2	16	$a^2 = b^8 = (ab)^2 = 1$	\mathbb{Z}_8 $\mathbb{Z}_2 \oplus \mathbb{Z}_8$ $\mathbb{Z}_2 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_4$
$\mathbb{Z}_4 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$	2	16		$\mathbb{Z}_2^n \oplus \mathbb{Z}_4$ for $0 \leq n \leq 5$
$\mathbb{Z}_2 \oplus D_4$	2	16	$a^2 = b^2 = c^2 =$ $(ca)^2 = (ab)^2 = (bc)^4 = 1$	$\mathbb{Z}_2^n \oplus \mathbb{Z}_4$ for $0 \leq n \leq 5$
\mathbb{Z}_2^4	2	16		\mathbb{Z}_2^n for $1 \leq n \leq 9$
\mathbb{Z}_3	3	3		\mathbb{Z}_3
\mathbb{Z}_9	3	9		\mathbb{Z}_9
$\mathbb{Z}_3 \oplus \mathbb{Z}_3$	3	9		\mathbb{Z}_3^n for $1 \leq n \leq 3$
\mathbb{Z}_{27}	3	27		\mathbb{Z}_{27}
$\mathbb{Z}_3 \oplus \mathbb{Z}_9$	3	27		$\mathbb{Z}_3^n \oplus \mathbb{Z}_9$ for $0 \leq n \leq 2$
$B(2, 3)$	3	27	$a^3 = b^3 = (ab)^3 = (a^{-1}b)^3 = 1$	\mathbb{Z}_3^n for $2 \leq n \leq 5$
\mathbb{Z}_3^3	3	27		\mathbb{Z}_3^n for $1 \leq n \leq 5$
\mathbb{Z}_5	5	5		\mathbb{Z}_5
\mathbb{Z}_{25}	5	25		\mathbb{Z}_{25}
$\mathbb{Z}_5 \oplus \mathbb{Z}_5$	5	25		\mathbb{Z}_3^n for $1 \leq n \leq 3$
\mathbb{Z}_7	7	7		\mathbb{Z}_7

Chapter 6

Groups with perfect subgroups

6.1 Introduction

Let G be a group and $N \trianglelefteq G$ be *perfect*, that is N is its own derived subgroup. In [35] R. H. Higgs showed that the sequence

$$1 \longrightarrow M(G/N) \longrightarrow M(G) \longrightarrow M(N) \tag{6.1}$$

is exact where M denotes the Schur multiplier of a group. For finite G , since $M(G) \cong H_2(G, \mathbb{Z}) \cong H^2(G, \mathbb{C}^*)$ the dual of (6.1) is the exact sequence

$$\longrightarrow M(N) \longrightarrow M(G) \longrightarrow M(G/N) \longrightarrow 1 \tag{6.2}$$

Noting that $M(Q) \cong \ker(Q \wedge Q \longrightarrow Q)$ in this chapter we give an analogue of (6.2) where $J(Q) = \ker(Q \otimes Q \longrightarrow Q)$ replaces M .

6.2 Results

6.2.1 Lemma *Let G be a group and $N \trianglelefteq G$ be perfect. Let $\overline{N \otimes G}$, $\overline{G \otimes N}$ and $\overline{N \otimes N}$ denote the images of $N \otimes G$, $G \otimes N$ and $N \otimes N$ in $G \otimes G$ under the*

canonical homomorphisms.

$$(i) \quad \overline{N \otimes G} = \overline{G \otimes N} = \overline{N \otimes N} \text{ in } G \otimes G.$$

(ii) *There is an exact sequence*

$$\overline{N \otimes G} \rightarrow G \otimes G \rightarrow G/N \otimes G/N.$$

Proof: (i) By Proposition 1.2.11 there is a crossed module $\lambda : G \otimes G \rightarrow G$ defined on generators by $\lambda(g \otimes h) = [g, h]$. Let $n \in N$. Since N is perfect there exists $\tau \in G \otimes G$ such that $\lambda(\tau) = n$. Then by 1.2.11 we have $n \otimes g = \lambda(\tau) \otimes g = \tau^g \tau^{-1} \in N$, and so $\overline{N \otimes G} = \overline{N \otimes N}$. Furthermore

$$\begin{aligned} \tau^g \tau^{-1} &= \left({}^g \tau \tau^{-1} \right)^{-1} \\ &= \left(g \otimes \lambda(\tau) \right)^{-1} \\ &= \left(g \otimes n \right)^{-1} \end{aligned}$$

from which it follows that $\overline{N \otimes G} = \overline{G \otimes N}$.

(ii) Clearly $\overline{N \otimes G} \trianglelefteq \ker(\phi)$ where $\phi : G \otimes G \rightarrow G/N \otimes G/N$. Then ϕ induces a homomorphism $\tilde{\phi} : G \otimes G / \overline{N \otimes G} \rightarrow G/N \otimes G/N$. We give the inverse of this map. Let $\psi : G/N \otimes G/N \rightarrow G \otimes G / \overline{N \otimes G}$, be defined on generators by $\bar{g} \otimes \bar{h} \xrightarrow{g} \otimes h \overline{N \otimes G}$, where $\bar{g} = gN$ and $\bar{h} = hN$. Now ψ is well defined since if $n_1, n_2 \in N$ then

$$\begin{aligned} gn_1 \otimes hn_2 \overline{N \otimes G} &= {}^g \left((n_1 \otimes h)^h (n_1 \otimes n_2) \right) (g \otimes h)^h (g \otimes n_2) \overline{N \otimes G} \\ &= g \otimes h \overline{N \otimes G}, \end{aligned}$$

where we make use of part (i). The homomorphism is the inverse of $\tilde{\phi}$ and so the result follows. ■

6.2.2 Theorem *Let G be a group and N a perfect normal subgroup of G . The sequence*

$$M(N) \longrightarrow J(G) \longrightarrow J(G/N) \longrightarrow 1$$

is exact, where M denotes the Schur multiplier and JG denotes the kernel of the canonical homomorphism $G \otimes G \longrightarrow G$.

Proof: Using the preceding lemma we have the following diagram.

$$\begin{array}{ccccc} \overline{N \otimes G} & \longrightarrow & G \otimes G & \xrightarrow{\phi} & G/N \otimes G/N \\ \downarrow & & \downarrow \lambda & & \downarrow \\ N & \longrightarrow & G & \longrightarrow & G/N \end{array}$$

Taking kernels of the vertical homomorphisms yields

$$\ker(\overline{N \otimes G} \rightarrow G) \longrightarrow J(G) \longrightarrow J(G/N).$$

Making use of Lemma 6.1 we have under the canonical homomorphism $\chi : N \otimes N \longrightarrow G \otimes G$,

$$\chi(\ker(N \otimes N \rightarrow G)) \leq \ker(\overline{N \otimes N} \rightarrow G) = \ker(\overline{N \otimes G} \rightarrow G).$$

Furthermore if $\bar{\tau} \in \ker(\overline{N \otimes G} \rightarrow G)$ then $t\bar{a}u \in \ker(\overline{N \otimes N} \rightarrow G)$ and so there exists $\tau \in N \otimes N$ such that $\chi(\tau) = \bar{\tau}$. It remains to show that the map $J(G) \longrightarrow J(G/N)$ is surjective. Let $x \in J(G/N)$. Since ϕ is surjective there exists $\tau \in G \otimes G$ such that $\phi(\tau) = x$. Then $\lambda(\tau) \in N$ and since N is perfect

$$\lambda(\tau) = [n_1, m_1][n_2, m_2] \dots [n_t, m_t]$$

for some $n_1, n_2, \dots, n_t, m_1, m_2, \dots, m_t \in N$. Then in $G \otimes G$ we have

$$\tau = (n_1 \otimes m_1)(n_2 \otimes m_2) \dots (n_t \otimes m_t)\kappa$$

for some $\kappa \in J(G)$. But then $\phi(\tau) = \phi(\kappa)$ and the restriction of ϕ to $J(G) \rightarrow J(G/N)$ is surjective. ■

6.3 Example

Let Λ be a ring with identity and $GL(n, \Lambda)$ denote the general linear group consisting of all $n \times n$ invertible matrices over Λ . A matrix in $GL(n, \Lambda)$ is called elementary if it is identical to the identity matrix except for a single off-diagonal entry. Let $e_{ij}^\lambda \in GL(n, \Lambda)$ with $1 \leq i, j \leq n$ and $\lambda \in \Lambda$ denote the elementary matrix with λ in the (i, j) th position. With this notation we have for example the identity $e_{ij}^\lambda e_{ij}^\mu = e_{ij}^{\lambda+\mu}$. R. Steinberg introduced [55] an abstract group designed to imitate the behaviour of the elementary matrices.

The subgroup generated by all the elementary matrices will be denoted by $E(n, \Lambda)$.

6.3.1 Definition For $n \geq 3$ the *Steinberg group* $St(n, \Lambda)$ is the group generated by symbols x_{ij}^λ , $1 \leq i, j \leq n$, $\lambda \in \Lambda$, satisfying the following relations

$$\begin{aligned} x_{ij}^\lambda x_{ij}^\mu &= x_{ij}^{\lambda+\mu} \\ [x_{ij}^\lambda, x_{jl}^\mu] &= x_{il}^{\lambda\mu} \text{ for } i \neq l \\ [x_{ij}^\lambda, x_{kl}^\mu] &= 1 \text{ for } j \neq k, i \neq l. \end{aligned}$$

The identities in the definition of the Steinberg group are satisfied by the elemen-

tary matrices so there is a canonical homomorphism

$$\phi : St(n, \Lambda) \longrightarrow GL(n, \Lambda)$$

given by $\phi(x_{ij}^\lambda) = e_{ij}^\lambda$.

Let $GL(\Lambda)$ denote the direct limit (or union) of the sequence

$$GL(1, \Lambda) \subset GL(2, \Lambda) \subset GL(3, \Lambda) \subset \dots$$

where each $GL(n, \Lambda)$ is injected into $GL(n + 1, \Lambda)$ by the correspondence

$$A \mapsto \begin{pmatrix} A & 0 \\ 0 & 1 \end{pmatrix}.$$

By taking limits as $n \rightarrow \infty$ we obtain corresponding groups, $E(\Lambda)$ of elementary matrices, and $St(\Lambda)$ the Steinberg group. There is a corresponding homomorphism

$$\phi : St(\Lambda) \longrightarrow GL(\Lambda).$$

The kernel of ϕ is called $K_2\Lambda$.

J. H. C. Whitehead showed that $E(\Lambda)$ is the commutator subgroup of $GL(\Lambda)$ (cf. [48]). The quotient $GL(\Lambda)/E(\Lambda)$ is called the Whitehead group and denoted by $K_1\Lambda$. The subgroup $E(\Lambda)$ is perfect [48]

Applying Theorem 6.2 gives

$$M(E(\Lambda)) \longrightarrow J(GL(\Lambda)) \longrightarrow J(GL(\Lambda)/E(\Lambda)) \longrightarrow 1.$$

Taking $\Lambda = \mathbb{Z}$ for example it is shown in [48, page 28 and corollary 10.2] that $K_1\mathbb{Z} = GL(\mathbb{Z})/E(\mathbb{Z}) \cong \mathbb{Z}_2$ and $K_2\mathbb{Z} \cong M(E(\mathbb{Z}))$, the Schur multiplier of $E(\mathbb{Z})$.

Therefore

$$K_2\mathbb{Z} \longrightarrow J(GL(\mathbb{Z})) \longrightarrow J(K_1\mathbb{Z}) \longrightarrow 1$$

$$\mathbb{Z}_2 \longrightarrow J(GL(\mathbb{Z})) \longrightarrow \mathbb{Z}_2 \longrightarrow 1$$

are exact. It follows that either $|J(GL(\mathbb{Z}))| = 2$ or $|J(GL(\mathbb{Z}))| = 4$.

Appendix A

Functions for computing tensor products

This appendix lists GAP functions for implementing actions between groups and producing the tensor product of two groups. Functions for producing the tensor square, the exterior product, the exterior square, and the q -tensor square are similar to the ones printed here.

The following is a full list of the functions for Appendix A. Only those marked with the \bullet symbol are printed here. The rest are available on the accompanying disk.

File		Function
ActSupport.g	\bullet	Implements actions between groups
ExteriorProduct.g		Computes Exterior product of two groups
ExteriorSquare.g		Computes Exterior square of a group
TensorProduct.g	\bullet	Computes Tensor product of two groups
TensorSquare.g		Computes Tensor square of a group
TensorSquareGen.g		Computes Tensor square of a group using the generating set for the computation of $W(x, a, b)$.
TensorSquareSupport.g		Tietze transformation wrappers.
q-tensor-simple.g		Computes the q tensor square of a group.

Program: TensorProduct.g

```
TensorProduct := function(arg)
# -----
# @(#) TensorProduct.g 1.1 97/12/16
# @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.TensorProduct.g
# -----
# Author: A.~McDermott
# Email: aidan.mcdermott@ucg.ie
```

```

#
# TensorProduct uses the method of Ellis \cite{E} to compute the tensor
# product of a finitely presented group.
#
# Usage:
# a)      TensorProduct(G,H,tietze,actionGH,actionHG)
# b)      TensorProduct(G,H,tietze,prime,bound,actionGH,actionHG)
#
# In usage a) G is a Finitely Presented Group and TensorSq returns
# the tensor square of G. The tietze option is explained below.
#
# In usage b) G should be a prime-power finitely presented group.
# In this case ANU's p-quotient function is used and a class bound for
# G is required.
#
# The arrays actionGH and actionHG give the action of G on H and H on
# G respectively. The i,j th element of actionGH gives the image of the
# jth generator of H under the action of the ith generator of G (see
# the function GactsonH for details).
#
# The tietze option is boolean, true if you wish to apply tietze
# transformations false otherwise. The usual Tietze functions
# will eliminate generators from the presentation provided (even
# when you ask it not to). Furthermore information about these
# generators are not passed back to the user. This makes calculations
# of things like  $\ker(G \backslash T G \rightarrow G)$  difficult.
#
# In order to overcome this problem the function xtz is called.
# This function will not eliminate any generators (provided G has
# no trivial generators).
#
# You may attach a record to G called 'mytzops' containing options to
# control the tietze trasformations (see xtz, xTzGo, and xTzGoGo for
# details).
# -----
local noarg,      # number of arguments
  G,              # Input fp-group arg[1]
  p, bound,      # arg[3] and arg[4]
  n,              # Number of generators of G
  cosrepG,       # coset representation of G
  phiG,          # function connecting G and cosrepG
  ucsG, lucsg,   # upper central series of cosrepG and its length
  F,             # Free group with 2n generators
  Fgen1, Fgen2,  # first and second n gens of F
  GensetG,       # list of generators form ucsG
  GenlistG,      # above mapped back to G
  CG,           # Centre of cosrepG
  Frels,        # relations for F
  T,            # F/Frels
  TT,          # Pq return from T
  TTgen1, TTgen2, # Generators for TT (2n)
  K1, K2,      # Normal closure in TT of TTgen1 etc.
  K,           # intersection of K1 and K2
  i, x, y, z, z1, x1, y2, ttt, ttt1, v1, v2, w1, GensG, CHG, DHG,
  A, B, elB, H, elg, elh, GensH, GensetH, GenlistH, actionGH,actionHG,
  G1,H1, m, ucsH, lucsh, GensG1, GensH1, GensetG1, GensetH1,

```

```

    elg1, elh1;
# -----
#
# -----
# Check arguments
# -----
noarg := Length(arg);
if not (noarg=5 or noarg=7) then
    Error( "TensorProduct requires 4 or 7 arguments" );
fi;

G:=arg[1];
H:=arg[2];
if noarg = 7 then
    p := arg[4];
    bound := arg[5];
    actionGH := arg[6];
    actionHG := arg[7];

    # ----- Pquotient -----
    G1 := Pq(G,rec(Prime:=p,ClassBound:=bound));
    H1 := Pq(H,rec(Prime:=p,ClassBound:=bound));
else
    actionGH := arg[4];
    actionHG := arg[5];
fi;

n := Length(G.generators);
m := Length(H.generators);
F := FreeGroup(n+m,"F");

Fgen1 := []; Fgen2 := [];
for i in [1..n] do
    Fgen1[i] := F.generators[i];    # First n generators
od;
for i in [1..m] do
    Fgen2[i] := F.generators[i+n];  # last m generators
od;

# -----
# For p-groups we can find the UCS and so a suitable
# generating set. This is just the generating set returned
# from Pq. We need to identify these elements in G and H.
# -----
if noarg = 7 then

    # -----
    # For G
    # -----
    GensetG1 := Set(Copy(G1.generators));
    GensetG := [];
    for elg in Elements(G) do
        for elg1 in GensetG1 do
            if MappedWord(elg,G.generators,G1.pqImages) = elg1 then
                Append(GensetG,[elg]);
            end if
        end for
    end for
end if

```



```

        RemoveSet(GensetG1,elg1);
    fi;
od;
od;

# -----
# For H
# -----
GensetH1 := Set(Copy(H1.generators));
GensetH := [];
for elh in Elements(H) do
    for elh1 in GensetH1 do
        if MappedWord(elh,H.generators,H1.pqImages) = elh1 then
            Append(GensetH,[elh]);
            RemoveSet(GensetH1,elh1);
        fi;
    od;
od;

else

    GensetG := Elements(G);
    GensetH := Elements(H);

fi;

GensetG := Set(GensetG);
RemoveSet(GensetG,G.identity);

GensetH := Set(GensetH);
RemoveSet(GensetH,H.identity);

GenlistG := G.generators;
GenlistH := H.generators;

# -----
# Main loop to create the words W(g,a,b) where g \in GensetG,
# a in GenlistG and b in GenlistH
# -----
Frels := [];
for z in GensetG do
    z1 := MappedWord(z,G.generators,Fgen1);

    for x in GenlistG do
        x1 := MappedWord(x,G.generators,Fgen1);
        ttt1 := MappedWord(x^z,G.generators,Fgen1);
        for y in GenlistH do
            y2 := MappedWord(y,H.generators,Fgen2);
            ttt := MappedWord(GactionH(G,H,z,y,actionGH),H.generators,Fgen2);

            v1 := z1 * x1*y2*x1^-1*y2^-1 * z1^-1 *
                ((z1*x1*z1^-1)*ttt*(z1*x1*z1^-1)^-1*ttt^-1)^-1;

            Append(Frels,[v1]);
        od;
    Frels:=Set(Frels);

```

```

    od;
od;

# -----
# Main loop to create the words  $W(h,a,b)$  where  $h \in \text{GensetH}$ ,
#  $a \in \text{GenlistG}$  and  $b \in \text{GenlistH}$ 
# -----
for z in GensetH do
  w1 := MappedWord(z,H.generators,Fgen2);
  for x in GenlistG do
    x1 := MappedWord(x,G.generators,Fgen1);
    ttt1 := MappedWord(GactionH(H,G,z,x,actionHG),G.generators,Fgen1);
    for y in GenlistH do
      y2 := MappedWord(y,H.generators,Fgen2);
      ttt := MappedWord(y^z,H.generators,Fgen2);

      v2 := w1 * x1*y2*x1^-1*y2^-1 * w1^-1 *
            (ttt1*(w1*y2*w1^-1)*ttt1^-1*(w1*y2*w1^-1)^-1)^-1;

      Append(Frels,[v2]);
    od;
    Frels:=Set(Frels);
  od;
od;

# -----
# Add the images of the relators of G in F
# -----
for z in G.relators do
  z1 := MappedWord(z,G.generators,Fgen1);
  Append(Frels,[z1]);
od;

# -----
# Add the images of the relators of G in F
# -----
for z in H.relators do
  w1 := MappedWord(z,H.generators,Fgen2);
  Append(Frels,[w1]);
od;

# -----
# Compute the big group  $((G \text{ \semi } H) \text{ \semi } G) \text{ \semi } H$ 
# -----
Frels := Set(Frels);
RemoveSet(Frels,F.identity);
T := F/Frels;

# -----
# If tietze then ....
# -----
if arg[2] = true then
  if IsBound(G.mytzops) then T.mytzops := G.mytzops; fi;
  T := xtz(T);
fi;

```

```

if noarg = 7 then
# -----
# If G and H are a p-groups then so is T and we can apply Pq.
# Need to keep track of how the old generators of T map
# into TT.
# -----
TT := Pq(T,rec(Prime:=p,ClassBound:=bound));

TTgen1 := [];
TTgen2 := [];
for i in [1..n] do
  TTgen1[i] := MappedWord(T.generators[i],T.generators,TT.pqImages);
  TTgen2[i] := MappedWord(T.generators[i+n],T.generators,TT.pqImages);
od;

elif noarg = 5 then
# -----
# This is a dummy version of the above.
# -----
TT := T;

TTgen1 := [];
TTgen2 := [];
for i in [1..n] do
  TTgen1[i] := MappedWord(T.generators[i],T.generators,TT.generators);
  TTgen2[i] := MappedWord(T.generators[i+n],T.generators,TT.generators);
od;
TT.pqImages := TT.generators;
fi;

# -----
# Compute the tensor product as the intersection of the
# Normal Closures.
# -----
K1 := NormalClosure(TT,Subgroup(TT,TTgen1));
K2 := NormalClosure(TT,Subgroup(TT,TTgen2));
K := Intersection(K1,K2);
K := AsSubgroup(TT,K);
return K;
end;

```

Program: ActSupport.g

```

GactsonH := function(G,H,g,h,Gacts)
# -----
# @(#) ActSupport.g 1.1 97/12/16
# @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.ActSupport.g

```

```

# -----
# Author: A.~McDermott
# Email: aidan.mcdermott@ucg.ie
# -----
# This function implements the action of one finitely presented group on
# another. That is, it works out  $gh$  for  $g \in G$  and  $h \in H$ .
#
# The action of  $G$  on  $H$  is given using the 2 dimensional array Gacts the
# columns of which correspond to the generators and the rows to the
# generators of  $H$ . The  $(i,j)$ th element of Gacts gives the image of the
#  $j$ th generator of  $H$  when acted on by the  $i$ th generator of  $G$ , ie
#  $^{g_i} h_j = \text{Gacts}[i][j]$ .
#
# The basic idea is we think of  $g$  a word in the generators of  $G$ 
# that is  $g = g_1 g_2 \dots g_n$ . Starting with  $g_n$  we figure out its
# action from Gacts and then apply it to  $h$ . We then apply  $g_{n-1}$  to
# the result etc.
# Of course, we are thinking of action on the left.
# -----

local G,          # Input group G
      H,          # Input group H
      g,h,        # Elements. We want g acting on h.
      Gacts,      # The action array see above
      lg,         # length of g in generators of G
      hTemp,      # Temporary result at the end of each iteration
      lhTemp,     # Length of hTemp in generators of H
      result,     # result computed during each iteration.
      gen,        # the particular generator of g we are working with
      Ga,         # the appropriate action of gen on H.generators
      subwd,      # Subword of g or h
      i,j,k,el,genH;

result := H.identity;
hTemp := h;
lg := LengthWord(g);

# -----
# First the simple case that g is the id of G
# Note that LengthWord(g) is then 0 and we don't go
# through the loop below.
# -----
if ( g in TrivialSubgroup(G) ) then
  result := h;
fi;

# -----
# Initialize the temporary result hTemp to h
# and go through the generators of g one at a time.
# -----

for i in [1..lg] do
  subwd := Subword(g,lg-i+1,lg-i+1);
  # Backwards because action on left

```

```

# -----
# Identify subwd among the generators of G or as
# one of their inverses.
# -----
for k in [1..Length(G.generators)] do
  gen := G.generators[k];
  if subwd = gen then          # its a generator
    Ga := Gacts[k];
  # -----
  # Its an inverse of a generator.
  # We use the fact that the map g --> action of g on H
  # is a homomorphism.  If n is the order of g in G then
  #  $g^n = 1$  and so (action of g on H)n = 1 in Aut(H).
  # One appropriate action for  $g^{-1}$  is
  # (action of g on H){n-1}.
  #
  # Without knowledge of Aut(H) I can't think of a better
  # (general) way of doing this although it is inefficient.
  # -----
  elif subwd = (gen)-1 then
    Ga := [];
    for genH in H.generators do
      for j in [1..(Order(G,subwd)-1)] do
        genH := GactsonH(G,H,gen,genH,Gacts);
        # Note: recursive call but well dampened.
      od;
      Append(Ga,[genH]);
    od;
  fi;
od;
lhTemp := LengthWord(hTemp);
# -----
# Work out gen action on hTemp. Store answer in result.
# -----
result := H.identity;
for j in [1..lhTemp] do
  subwd := Subword(hTemp,j,j);
  for k in [1..Length(H.generators)] do
    gen := H.generators[k];
    if (subwd-1*gen) in TrivialSubgroup(H) then
      result := result * Ga[k];
    elif (subwd*gen) in TrivialSubgroup(H) then
      result := result * (Ga[k])-1;
    fi;
  od;
od;
# -----
# The following is necessary to keep control over
# the size of words produced.  Without it they
# tend to grow impossibly large.
# -----
for el in Elements(H) do
  if result*(el-1) in TrivialSubgroup(H) then
    result := el;
  fi;
od;

```

```
    hTemp := result;  
  od;  
  return result;  
end;
```

Appendix B

Programs for Chapter 2

This appendix lists the programs used to produce the tables and other results in Chapter 2. The following table lists all the programs associated with this appendix. Only those marked with the • symbol are actually printed here, the rest can be found on the accompanying diskette.

File/Program		Function	Table/Page
Q3.g	•	Computes $G \otimes H$ for all subgroups G and H of Q_{32}	Table 2.2, Table 3.1
Q3w.g		Computes $G \wedge H$ for all subgroups G and H of Q_{32} .	Table 2.3
actions.g	•	Computes all pairs of actions between D4 and Q8. Finds isomorphic pairs of actions.	pp 44–45
compatible_actions.g	•	Computes all pairs of actions between D4 and Q8. Decides which pairs of actions are compatible.	pp 44–45
compatible_classes.g		Takes all isomorphic pairs of actions between D4 and Q8. Decides which classes give rise to compatible pairs of actions.	pp 44–45
TProd_D4Q8.g	•	Computes Tensor product of D4 with Q8 under all pairs of compatible actions.	Table 2.4
TProd_D4Q8_BL2.g		Computes Tensor product of D4 with Q8 under all non-compatible actions. Uses Brown and Loday definition.	Table 2.5
TProd_D4Q8_In2.g	•	Computes Tensor product of D4 with Q8 under all non-compatible actions. Uses N. Inassaridze's definition.	Table 2.5

Program: Q3.g

```
LogTo("Q3.out");

## -----
## @(#) Q3.g 1.1 97/12/16
## @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.Q3.g
## -----
## Author: A.~McDermott
## Email: aidan.mcdermott@ucg.ie
## -----
## For each pair of subgroups G and H of the Quaternion group Q of
## order 32 this program computes the tensor product  $G \setminus T H$  where
## the actions are those arising from conjugation in Q. The
## subgroup  $\ker(G \setminus T H \rightarrow Q)$  is also identified in each case.
## -----

RequirePackage("anupq");
Read("/usr/users/aidan/reruns/ap1/ActSupport.g");
Read("/usr/users/aidan/reruns/ap1/TensorSquareSupport.g");
Read("/usr/users/aidan/reruns/ap1/TensorProduct.g");

## -----
## Initialize some variables
## -----
A := 0;; J := 0;;

## -----
## Set up Q32 and its normal subgroups.
## These are set up as finitely presented groups with the
## mapping of each into Q32 itself.
## -----

n := 5;
F := FreeGroup(2,"F");

## -----
## Subgroups of order 2
## -----
I2Rel := [ F.1^2, F.2 ];
## -----
## Subgroups of order 4
## -----
I3Rel := [ F.1^4, F.2 ];
I4Rel := [ F.1^4, F.2 ];
I5Rel := [ F.1^2, F.2^2, Comm(F.1,F.2) ];
## -----
## Subgroups of order 8
## -----
I6Rel := [ F.1^4, Comm(F.1,F.2), F.2^-1*F.1^-1*F.2^-1*F.1^-1 ];
I7Rel := [ F.1^8, F.2 ];
I8Rel := [ F.1^8, F.2 ];
## -----
## Subgroups of order 16
## -----
```



```

I9Rel := [ F.1^4, F.2*F.1*F.2^-1*F.1^-1, F.1^2*F.2^4 ];
I10Rel := [ F.1^16, F.2 ];
I11Rel := [ F.1^16, F.2 ];
## -----
## Q32 itself
## -----
I12Rel := [ F.1^(2^(n-1)), F.2^(2^(n-2))*F.1^(-2), F.1*F.2^(-1)*F.1*F.2 ];

## -----
## Form the array of 12 fp groups and identify Q32 as the last.
## Print out basics about each.
## -----
Gps := [];
Gps[2] := F/I2Rel;
Gps[3] := F/I3Rel;
Gps[4] := F/I4Rel;
Gps[5] := F/I5Rel;
Gps[6] := F/I6Rel;
Gps[7] := F/I7Rel;
Gps[8] := F/I8Rel;
Gps[9] := F/I9Rel;
Gps[10] := F/I10Rel;
Gps[11] := F/I11Rel;
Gps[12] := F/I12Rel;
Q32 := Gps[12];

for i in [2..12] do
  Print("size of Gp",i," is ",Size(Gps[i]),"\n");
  Print("Abelian ",IsAbelian(Gps[i]),"\n");
od;

## -----
## Mapsto tells us how the generators of the pq groups are
## mapped into Q32.
## -----
mapsto := [ [ Q32.identity, Q32.identity ],
             [ Q32.1^2, Q32.identity ],
             [ Q32.1, Q32.identity ],
             [ Q32.2^4, Q32.identity ],
             [ Q32.1^2, Q32.1*Q32.2^4 ],
             [ Q32.1, Q32.2^4 ],
             [ Q32.1^2*Q32.2^2, Q32.identity ],
             [ Q32.1*Q32.2^2, Q32.identity ],
             [ Q32.1, Q32.2^2 ],
             [ Q32.2, Q32.identity ],
             [ Q32.1*Q32.2, Q32.identity ],
             [ Q32.1, Q32.2 ] ];

## -----
## Main Loop: For each pair of groups G,H work out G \T H.
## -----

for i in [2..12] do

```

```

G := Gps[i];
for j in [2..12] do
  H := Gps[j];

## -----
## These are the different types of actions
## that arise among the subgroups of Q32.
## -----
Tr := [ [H.1,H.2], [H.1,H.2] ];
B3 := [ [H.1^3,H.2], [H.1,H.2] ];
B3a := [ [H.1,H.2], [H.1^3,H.2] ];
D3 := [ [H.1, H.1*H.2], [ H.1, H.2 ] ];
D3a := [ [H.1, H.2], [ H.1, H.1*H.2 ] ];
F3 := [ [H.1^5,H.2], [H.1,H.2] ];
F3a := [ [H.1,H.2], [H.1^5,H.2] ];
H9 := [ [H.1^9,H.2], [H.1,H.2] ];
H9a := [ [H.1,H.2], [H.1^9,H.2] ];
H9b := [ [H.1^9,H.2], [H.1^9,H.2] ];
X := [ [H.1,H.1^2*H.2], [H.1,H.2] ];
X2 := [ [H.1,H.2], [H.1,H.1^2*H.2] ];
X3 := [ [H.1,H.2], [H.1^2*H.2,H.2] ];
B3X := [ [H.1^3,H.1^2*H.2], [H.1,H.2] ];
B3Xa := [ [H.1,H.1^2*H.2], [H.1^3,H.2] ];

## -----
## These are the actions of the subgroup G on
## subgroup H.
## -----
BListH := [
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, B3, B3, B3a ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, D3, D3, D3a ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, B3, B3, B3a ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, F3, F3, F3a ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, B3, B3, B3a ],
  [ Tr, Tr, H9, Tr, H9a, H9, Tr, H9, H9, Tr, H9, H9 ],
  [ Tr, Tr, H9, Tr, H9a, H9, Tr, H9, H9, H9, Tr, H9b ],
  [ Tr, Tr, X, Tr, X2, X3, Tr, X, X3, B3, B3X, B3Xa ] ];

## -----
## These are the different types of actions
## that arise among the subgroups of Q32.
## written in the generators of G.
## -----
Tr := [ [G.1,G.2], [G.1,G.2] ];
B3 := [ [G.1^3,G.2], [G.1,G.2] ];
B3a := [ [G.1,G.2], [G.1^3,G.2] ];
D3 := [ [G.1, G.1*G.2], [ G.1, G.2 ] ];
D3a := [ [G.1, G.2], [ G.1, G.1*G.2 ] ];

```

```

F3 := [ [G.1^5,G.2], [G.1,G.2] ];
F3a := [ [G.1,G.2], [G.1^5,G.2] ];
H9 := [ [G.1^9,G.2], [G.1,G.2] ];
H9a := [ [G.1,G.2], [G.1^9,G.2] ];
H9b := [ [G.1^9,G.2], [G.1^9,G.2] ];
X := [ [G.1,G.1^2*G.2], [G.1,G.2] ];
X2 := [ [G.1,G.2], [G.1,G.1^2*G.2] ];
X3 := [ [G.1,G.2], [G.1^2*G.2,G.2] ];
B3X := [ [G.1^3,G.1^2*G.2], [G.1,G.2] ];
B3Xa := [ [G.1,G.1^2*G.2], [G.1^3,G.2] ];

## -----
## These are the actions of the subgroup H on
## subgroup G.
## -----
BListG := [
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, B3, B3, B3a ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, D3, D3, D3a ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, B3, B3, B3a ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, F3, F3, F3a ],
  [ Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, Tr, B3, B3, B3a ],
  [ Tr, Tr, H9, Tr, H9a, H9, Tr, H9, H9, Tr, H9, H9 ],
  [ Tr, Tr, H9, Tr, H9a, H9, Tr, H9, H9, H9, Tr, H9b ],
  [ Tr, Tr, X, Tr, X2, X3, Tr, X, X3, B3, B3X, B3Xa ] ];

## -----
## Pick the correct action and call the
## TensorProduct function.
## -----
ActGH := BListH[j][i];
ActHG := BListG[i][j];
Print("Tensor Product of group ",i," with group ",j,"\n");
A := TensorProduct(G,H,false,2,60,ActGH,ActHG);

## -----
## Print out some information to identify the
## group.
## -----
Print("Size of A = ",Size(A),"\n");
Print("A is abelian = ",IsAbelian(A),"\n");
Print("A is cyclic = ",IsCyclic(A),"\n");
ord := List(Elements(A),x->Order(A,x));
Sort(ord);
Print("orders = ",ord,"\n");

## -----
## Need to identify the ker(G \ T H --> Q).
## We do this in two steps

```

```

## a) Identify those elements of A which are
## of the form g \T h.
## b) Identify the rest of the elements as
## products of those found in a.
## -----
list1 := [A.parent.pqImages[1],A.parent.pqImages[2]];
list2 := [A.parent.pqImages[3],A.parent.pqImages[4]];
Hlist := []; Alist := []; ker:= []; ident := [];

notfound := Copy(Elements(A)); ## a list of elements we still
## have to indentify.

for e11 in Elements(G) do
  e11t := MappedWord(e11,G.generators,list1);
  for e12 in Elements(H) do
    e12t := MappedWord(e12,H.generators,list2);
    if not Comm(e11t,e12t) in ident then
      e11q := MappedWord(e11,G.generators,mapsto[i]);
      e12q := MappedWord(e12,H.generators,mapsto[j]);
      inH := [ e11q, e12q, Comm(e11q,e12q) ];
      inA := [ e11t, e12t, Comm(e11t,e12t) ];
      Append(inA,inH);
      Append(Hlist,[inH]);
      Append(Alist,[inA]);
      Append(ident,[inA[3]]);
      RemoveSet(notfound,inA[3]);
      if inH[3] in TrivialSubgroup(Q32) then
        Append(ker,[inA[3]]);
      fi;
    fi;
  od;
od;
ident := Set(ident);

## -----
## How many elements have we got so far?
## -----
Print("Has Found ",Length(ident)," elements in A\n");

## -----
## Part b
## -----
newfind := []; prodlist := [];
for A111 in Alist do
  if Length(notfound) <> 0 then
    for A121 in Alist do
      if A111[3]*A121[3] in notfound then
        Append(newfind,[A111[3]*A121[3]]);
        ellist := [A111,A121,A111[3]*A121[3]];
        Append(prodlist,[ellist]);
        RemoveSet(notfound,A111[3]*A121[3]);
        if A111[6]*A121[6] in TrivialSubgroup(Q32) then
          Append(ker,[A111[3]*A121[3]]);
        fi;
      fi;
    fi;
  fi;
fi;

```

```

        od;
        fi;
    od;
newfind := Set(newfind);
## -----
## How many elements have we still not found?
## Should be zero or we have more work to do.
## -----
Print("Has not Found ",Length(notfound)," elements in A\n");

## -----
## J is ker(G \T H --> Q)
## -----
J := Subgroup(A,ker);
Print("Size of J is ",Size(J)," \n");
Print("is cyclic? ",IsCyclic(J)," \n");
ord := List(Elements(J),x->Order(J,x));
Sort(ord);
Print("orders of J= ",ord," \n");
Print("\n\n");

    od; ## End main loop
od;

```

Program: actions.g

```

LogTo("actions.out");

## -----
## @(#) actions.g 1.1 97/12/16
## @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.actions.g
## -----
## Author: A.~McDermott
## Email: aidan.mcdermott@ucg.ie
## -----
## This program computes all pairs of actions between D4 and Q8
## where D4 is the dihedral group of order 8 and Q8 is the
## Quaternionian group of order 8.
##
## These actions are then broken into equivalence classes of
## isomorphic actions and the number and size of each is reported.
## -----

Read("/usr/users/aidan/reruns/ap1/ActSupport.g");

## -----

```

```

## Initialize some variables
## -----
imx := 0;; ima := 0;;

## -----
## Make the two groups, G = D4 and H = Q8.
## -----
F := FreeGroup(2);
x := F.1;; y := F.1;; a := F.2;;

relG := [ x^2, a^4, x*a*x^-1*a ];;      G := F/relG;
relH := [ a^4, y^2*a^-2, y*a*y^-1*a ];; H := F/relH;

elG := Copy(Elements(G));
elH := Copy(Elements(H));

## -----
## The automorphisms of the groups G and H.
## The automorphism group of G=D4 is D4 itself and
## the automorphism group of H=Q8 is S4.
## We need three versions of these, one written in the
## abstract words u,v, and w and one as finitely presented
## groups. A third one is below.
## -----

u := AbstractGenerator("u");;
v := AbstractGenerator("v");;
w := AbstractGenerator("w");;
AutGabsels := [ IdWord, u, v, v*u, u*v*u, u*v, v^2, u*v^2 ];;
AutHabsels := [ IdWord, v, u*v*u, v^2, w*v, w, w*v^2, w*u*v*u, w^2,
                w^2*v, w^2*v^2, w^2*u*v*u, w^2*u, w^2*v*v*u, w^2*u*v^2,
                w^2*u*v, u, v*u, u*v, u*v^2, w*v*u, w*u, w*u*v^2, w*u*v ];;

F := FreeGroup(2);
AutGrel := [ F.1^2, F.2^4, F.1*F.2*F.1^-1*F.2 ];;
AutG := F/AutGrel;;
F := FreeGroup(3);
AutHrel := [ F.1^2, F.2^4, F.3^3,
             F.1*F.2*F.1^-1*F.2,
             F.3*F.2*F.3*F.2, F.3^2*F.1*F.3*F.2^-2 ];;
AutH := F/AutHrel;;

## -----
## Some checking that all went well so far.
## -----
Print("Size of G = ",Size(G)," and size of Aut G = ",Size(AutG),"\\n");
Print("Size of H = ",Size(H)," and size of Aut H = ",Size(AutH),"\\n");

a := AbstractGenerator("a");;
b := AbstractGenerator("b");;

```

```

## -----
## Find all pairs of elements of G which can be images
## of the generators under an automorphism. ( ie check
## which pairs satisfy the relations etc ). This is the
## third version of Aut(G).
## -----

imG := [];
elG3 := [];
absG := [];
for imx in Elements(G) do
  for ima in Elements(G) do

    if Order(G,imx) = 2 and Order(G,ima) = 4
      and imx*ima*imx^-1*ima in TrivialSubgroup(G)
      and Size(Subgroup(G,[imx,ima])) = Size(G) then

        Append(imG,[[imx,ima]]);

        ## -----
        ## Now the elements imx and ima are just as good a generating
        ## set as the elements we started with G.1 and G.2.
        ## elG is a list of the elements of G written in the gen
        ## set G.1 and G.2.
        ## What will each of the elements of elG look like when
        ## written in the generating set [imx,ima]?
        ## -----
        elG2 := List(elG, kk->MappedWord(kk,[G.1,G.2],[imx,ima]));
        for n in [1..Length(elG)] do
          el2 := elG[n];
          found := false;
          k := 0;
          while found = false do
            k := k+1;
            if el2*elG2[k]^-1 in TrivialSubgroup(G) then
              elG3[n] := MappedWord(elG[k],[G.1,G.2],[a,b]);
              found := true;
            fi;
          od;
        od;
        elG4 := Copy(elG3);
        Append(absG,[elG4]);
        ## -----
        ## Print out the results so we can check.
        ## -----
        Print("a =",imx,"\t b = ",ima,"\t",elG3,"\n");
      fi;
    od;
  od;
  #
  Print("\n\n");

  ## -----
  ## Do the same for H
  ## -----

```

```

imH := [];
elH3 := [];
absH := [];
for imx in Elements(H) do
  for ima in Elements(H) do

    if Order(H,ima) = 4
      and imx^2*ima^-2 in TrivialSubgroup(H)
      and imx*ima*imx^-1*ima in TrivialSubgroup(H)
      and Size(Subgroup(H,[imx,ima])) = Size(H) then

        ## -----
        ## Again what will each of the elements of elH look like
        ## when written in the generating set [imx,ima]?
        ## -----
        Append(imH,[[imx,ima]]);
        elH2 := List(elH, kk->MappedWord(kk,[H.1,H.2],[imx,ima]));
        for n in [1..Length(elH)] do
          el2 := elH[n];
          found := false;
          k := 0;
          while found = false do
            k := k+1;
            if el2*elH2[k]^-1 in TrivialSubgroup(H) then
              elH3[n] := MappedWord(elH[k],[H.1,H.2],[a,b]);
              found := true;
            fi;
          od;
          od;
          elH4 := Copy(elH3);
          Append(absH,[elH4]);
          ## -----
          ## Print out the results so we can check.
          ## -----
          Print("a =",imx,"\t b = ",ima,"\t",elH3,"\n");
        fi;
      od;
    od;
  od;

  ## -----
  ## ImG and imH are now arrays of automorphisms of G and H
  ## respectively. We choose two elements of imG (one for
  ## each generator of H) and check that it is an action
  ## (ie that it satisfies the relations for H).
  ## -----

  # All pairs of actions
  actions := [];; mark := [];;
  for i in [1..Length(imG)] do
    phiG1 := imG[i];
    for j in [1..Length(imG)] do
      phiG2 := imG[j];

      ## -----
      ## Need to check that this choice is an action
    od;
  od;

```



```

## The idea is that we map the chosen elements onto the
## corresponding elements in AutG and check the relations
## there. (Only AutG has a GAP group structure).

abstractact1 := AutGabsels[i];
abstractact2 := AutGabsels[j];
act1 := MappedWord(abstractact1,[u,v],AutG.generators);
act2 := MappedWord(abstractact2,[u,v],AutG.generators);

## -----
## If not an action go back and pick again.
## -----
if act2^4 in TrivialSubgroup(AutG)
and act1^2*act2^-2 in TrivialSubgroup(AutG)
and act1*act2*act1^-1*act2 in TrivialSubgroup(AutG) then

## -----
## Now pick two elements in the automorphism list of H
## Check that it is an action of G on H.
## -----
for k in [1..Length(imH)] do
  phiH1 := imH[k];
  for l in [1..Length(imH)] do
    phiH2 := imH[l];

    abstractact1 := Authabsels[k];
    abstractact2 := Authabsels[l];
    act1 := MappedWord(abstractact1,[u,v,w],Auth.generators);
    act2 := MappedWord(abstractact2,[u,v,w],Auth.generators);

    if act1^2 in TrivialSubgroup(Auth)
and act2^4 in TrivialSubgroup(Auth)
and act1*act2*act1^-1*act2 in TrivialSubgroup(Auth) then

## -----
## Found a pair of actions. Append to list actions.
## -----
acts := Copy([ phiG1, phiG2, phiH1, phiH2 ]);
Append(actions,[acts]);
Append(mark,[true]);
    fi;
  od;
od;
fi;
od;
actions := Set(actions);;
actionscpy := Copy(actions);;
Print("Number of actions = ",Length(actions),"\n");

## -----
## Part II
## Go through all actions and find the equivalence classes.
## -----

```

```

for i in [1..Length(actions)] do

  if mark[i] = true then      ## mark[i] being true means we have
                              ## not already found it equivalent to
                              ## an earlier action.

    ## -----
    ## These are arrays for the actions used by GactsonH
    ## -----
    actHG := [ [ actions[i][1][1], actions[i][1][2] ],
               [ actions[i][2][1], actions[i][2][2] ] ];

    actGH := [ [ actions[i][3][1], actions[i][3][2] ],
               [ actions[i][4][1], actions[i][4][2] ] ];

    orig := Copy(actions[i]);
    findacts := [];
    ## -----
    ## What will this action do to the equivalent generating
    ## sets in imG and imH? What will the action look like
    ## from these equivalent generating sets.
    ## -----
    for l in [1..Length(imG)] do
      phiG := imG[l];
      for m in [1..Length(imH)] do
        phiH := imH[m];

        ## g1h1 stands for g1 acting in h1 etc
        g1h1 := GactsonH(G,H,phiG[1],phiH[1],actGH);
        g1h2 := GactsonH(G,H,phiG[1],phiH[2],actGH);
        g2h1 := GactsonH(G,H,phiG[2],phiH[1],actGH);
        g2h2 := GactsonH(G,H,phiG[2],phiH[2],actGH);

        h1g1 := GactsonH(H,G,phiH[1],phiG[1],actHG);
        h1g2 := GactsonH(H,G,phiH[1],phiG[2],actHG);
        h2g1 := GactsonH(H,G,phiH[2],phiG[1],actHG);
        h2g2 := GactsonH(H,G,phiH[2],phiG[2],actHG);

        ## -----
        ## Each of these elements is in the groups G or H.
        ## They represent our given action from the perspective
        ## of the new generating sets phiG and phiH. However
        ## they are still written in terms of the old generators.
        ## We now find what they look like when written in terms
        ## of the new generating sets.
        ## First find the elements in G (or H) and from its position
        ## in the list elG (elH) we know its abstract form in the
        ## abstract generators a and b.
        ## By mapping this back into G (or H) we can then find try
        ## and find it in the action list actions.
        ## -----
        for k in [1..Length(elG)] do
          if h1g1*elG[k]^-1 in TrivialSubgroup(G) then
            ah1g1 := absG[1][k];

```

```

        xh1g1 := MappedWord(ah1g1,[a,b],[G.1,G.2]);
    fi;
    if h1g2*e1G[k]^-1 in TrivialSubgroup(G) then
        ah1g2 := absG[1][k];
        xh1g2 := MappedWord(ah1g2,[a,b],[G.1,G.2]);
    fi;
    if h2g1*e1G[k]^-1 in TrivialSubgroup(G) then
        ah2g1 := absG[1][k];
        xh2g1 := MappedWord(ah2g1,[a,b],[G.1,G.2]);
    fi;
    if h2g2*e1G[k]^-1 in TrivialSubgroup(G) then
        ah2g2 := absG[1][k];
        xh2g2 := MappedWord(ah2g2,[a,b],[G.1,G.2]);
    fi;
od;
for k in [1..Length(e1H)] do
    if g1h1*e1H[k]^-1 in TrivialSubgroup(H) then
        ag1h1 := absH[m][k];
        xg1h1 := MappedWord(ag1h1,[a,b],[H.1,H.2]);
    fi;
    if g1h2*e1H[k]^-1 in TrivialSubgroup(H) then
        ag1h2 := absH[m][k];
        xg1h2 := MappedWord(ag1h2,[a,b],[H.1,H.2]);
    fi;
    if g2h1*e1H[k]^-1 in TrivialSubgroup(H) then
        ag2h1 := absH[m][k];
        xg2h1 := MappedWord(ag2h1,[a,b],[H.1,H.2]);
    fi;
    if g2h2*e1H[k]^-1 in TrivialSubgroup(H) then
        ag2h2 := absH[m][k];
        xg2h2 := MappedWord(ag2h2,[a,b],[H.1,H.2]);
    fi;
od;

## This is an equivalent action.
equiv := Copy([ [xh1g1,xh1g2], [xh2g1,xh2g2],
                [xg1h1,xg1h2], [xg2h1,xg2h2] ]);
Append(fndacts,[equiv]);
od;
od;

## -----
## fndacts are all the equivalent to the given action.
## -----
fndacts := Set(fndacts);
Print("Number of equivalent = ",Length(fndacts),"n");

## -----
## We don't want to consider any of the equivalent actions
## again, so we find their position in the set actions
## and mark the equivalent position in mark as false.
## -----
RemoveSet(fndacts,orig);
equals := [Position(actions,orig)];
for acts in fndacts do

```

```

        pos := Position(actions,acts);
        mark[pos] := false;
        Append(equals,[pos]);
        RemoveSet(actionscpy,acts);
    od;
    Print("Found for action ",i," equivalents = ",equals,"\n");

fi;
od;
Print("Number of Actions          = ",Length(actions),"\n");
Print("Number of equivalent actions = ",Length(actionscpy),"\n");
Print("The first member in each equivalent class is\n");
Print(actionscpy,"\n");

```

Program: compatible_actions.g

```

LogTo("compatible_actions.out");

## -----
## @(#) compatible_actions.g 1.1 97/12/16
## @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.compatible_actions.g
## -----
## Author: A.~McDermott
## Email:  aidan.mcdermott@ucg.ie
## -----
## This program computes all pairs of actions between D4 and Q8
## where D4 is the dihedral group of order 8 and Q8 is the
## Quaternion group of order 8.  It then decides which of these
## pairs of actions are compatible.
## -----

Read("/usr/users/aidan/reruns/ap1/ActSupport.g");

## -----
## Initialize some variables
## -----
imx := 0;; ima := 0;; count:=0;;

## -----
## Make the two groups, G = D4 and H = Q8.
## -----
F := FreeGroup(2);
x := F.1;; y := F.1;; a := F.2;;

relG := [ x^2, a^4, x*a*x^-1*a ];;      G := F/relG;;
relH := [ a^4, y^2*a^-2, y*a*y^-1*a ];;  H := F/relH;;

#ord := List(Elements(G),j->Order(G,j));

```

```

#Elements(G);
#ord := List(Elements(H),j->Order(H,j));
#Elements(H);
#ord;
#

## -----
## The automorphisms of the groups G and H.
## The automorphism group of G=D4 is D4 itself and
## the automorphism group of H=Q8 is S4.
## We need three versions of these, one written in the
## abstract words u,v, and w and one as finitely presented
## groups. A third one is below.
## -----

u := AbstractGenerator("u");;
v := AbstractGenerator("v");;
w := AbstractGenerator("w");;
AutGabsels := [ IdWord, u, v, v*u, u*v*u, u*v, v^2, u*v^2 ];;
AutHabsels := [ IdWord, v, u*v*u, v^2, w*v, w, w*v^2, w*u*v*u, w^2,
                w^2*v, w^2*v^2, w^2*u*v*u, w^2*u, w^2*v*u, w^2*u*v^2,
                w^2*u*v, u, v*u, u*v, u*v^2, w*v*u, w*u, w*u*v^2, w*u*v ];;

F := FreeGroup(2);
AutGrel := [ F.1^2, F.2^4, F.1*F.2*F.1^-1*F.2 ];;
AutG := F/AutGrel;;
F := FreeGroup(3);
AutHrel := [ F.1^2, F.2^4, F.3^3,
             F.1*F.2*F.1^-1*F.2,
             F.3*F.2*F.3*F.2, F.3^2*F.1*F.3*F.2^-2 ];;
AutH := F/AutHrel;;

## -----
## Some checking that all went well so far.
## -----
Print("Size of G = ",Size(G)," and size of Aut G = ",Size(AutG)," \n");
Print("Size of H = ",Size(H)," and size of Aut H = ",Size(AutH)," \n");

## -----
## Find all pairs of elements of G which can be images
## of the generators under an automorphism. ( ie check
## which pairs satisfy the relations etc ). This is the
## third version of Aut(G).
## -----

imG := [];
for imx in Elements(G) do
  for ima in Elements(G) do

    if Order(G,imx) = 2 and Order(G,ima) = 4
      and imx*ima*imx^-1*ima in TrivialSubgroup(G)
      and Size(Subgroup(G,[imx,ima])) = Size(G) then

      Append(imG,[[imx,ima]]);

```

```

    fi;
  od;
od;

## -----
## Do the same for H
## -----
imH := [];
for imx in Elements(H) do
  for ima in Elements(H) do

    if Order(H,ima) = 4
      and imx^2*ima^-2 in TrivialSubgroup(H)
      and imx*ima*imx^-1*ima in TrivialSubgroup(H)
      and Size(Subgroup(H,[imx,ima])) = Size(H) then

        Append(imH,[[imx,ima]]);
      fi;
    od;
  od;
od;

## -----
## ImG and imH are now arrays of automorphisms of G and H
## respectively. We choose two elements of imG (one for
## each generator of H) and check that it is an action
## (ie that it satisfies the relations for H).
## -----

for i in [1..Length(imG)] do
  phiG := imG[i];
  for j in [1..Length(imG)] do
    psiG := imG[j];

    ## -----
    ## Need to check that this choice is an action
    ## The idea is that we map the chosen elements onto the
    ## corresponding elements in AutG and check the relations
    ## there. (Only AutG has a GAP group structure).

    abstractact1 := AutGabsels[i];
    abstractact2 := AutGabsels[j];
    act1 := MappedWord(abstractact1,[u,v],AutG.generators);
    act2 := MappedWord(abstractact2,[u,v],AutG.generators);

    ## -----
    ## If not an action go back and pick again.
    ## -----
    if act2^4 in TrivialSubgroup(AutG)
      and act1^2*act2^-2 in TrivialSubgroup(AutG)
      and act1*act2*act1^-1*act2 in TrivialSubgroup(AutG) then

      ## Haction is how H acts on G.
      Haction := [ phiG, psiG ];
    fi;
  od;
od;

```

```

## -----
## Now pick two elements in the automorphism list of H
## Check that it is an action of G on H.
## -----
for k in [1..Length(imH)] do
  phiH := imH[k];
  for l in [1..Length(imH)] do
    psiH := imH[l];

    abstractact1 := Authabsels[k];
    abstractact2 := Authabsels[l];
    act1 := MappedWord(abstractact1,[u,v,w],AutH.generators);
    act2 := MappedWord(abstractact2,[u,v,w],AutH.generators);

    if act1^2 in TrivialSubgroup(AutH)
      and act2^4 in TrivialSubgroup(AutH)
      and act1*act2*act1^-1*act2 in TrivialSubgroup(AutH) then

      ## Gaction is how G acts on H.
      Gaction := [ phiH, psiH ];

      ## -----
      ## Found a pair of actions. See is it compatible.
      ## -----
      compat := true;

      for g in Elements(G) do
        if compat = true then
          for h in Elements(H) do

            gonh := GactsonH(G,H,g,h,Gaction);
            hong := GactsonH(H,G,h,g,Haction);

            for kk in Elements(G) do
              ## -----
              ## is  $(^g h) kk = \{g h g^{-1}\} kk$  ??
              ## -----
              lhs := GactsonH(H,G,gonh,kk,Haction);
              rhs := g*GactsonH(H,G,h,g^-1*kk*g,Haction)*g^-1;
              if not lhs*rhs^-1 in TrivialSubgroup(G) then

                compat := false;
              fi;
            od;

            if compat = true then
              for kk in Elements(H) do
                ## -----
                ## is  $(^h g) kk = \{h g h^{-1}\} kk$  ??
                ## -----
                lhs := GactsonH(G,H,hong,kk,Gaction);
                rhs := h*GactsonH(G,H,g,h^-1*kk*h,Gaction)*h^-1;
                if not lhs*rhs^-1 in TrivialSubgroup(H) then

                  compat := false;
                fi;
              od;
            fi;
          end if;
        end if;
      end for;
    end if;
  end for;
end for;

```

```

        od;
        fi;
    od;
    fi;
od;
if compat = true then
    count := count + 1;
    Print("Haction = ",Haction,"\n");
    Print("Gaction = ",Gaction,"\n");
fi;
fi;
od;
od;
fi;
od;
od;
Print("There are ",count," compatible actions\n");

```

Program: TProd_D4Q8.g

```

LogTo("TProd_D4Q8.out");

## -----
## @(#) TProd_D4Q8.g 1.1 97/12/16
## @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.TProd_D4Q8.g
## -----
## Author: A.~McDermott
## Email: aidan.mcdermott@ucg.ie
## -----
## For the first member of each class of compatible actions
## between D4 and Q8, this program computes the tensor product
## of D4 and Q8.
## -----

RequirePackage("anupq");
Read("/usr/users/aidan/reruns/ap1/ActSupport.g");
Read("/usr/users/aidan/reruns/ap1/TensorSquareSupport.g");
Read("/usr/users/aidan/reruns/ap1/TensorProduct.g");

## -----
## Initialize some variables
## -----
el:=0;; A:=0;;

## -----
## Make the two groups, G = D4 and H = Q8.
## -----

```



```

F := FreeGroup(2);
x := F.1;; y := F.1;; a := F.2;;

relG := [ x^2, a^4, x*a*x^-1*a ];;      G := F/relG;;
relH := [ a^4, y^2*a^-2, y*a*y^-1*a ];; H := F/relH;;

## -----
## This is a list of representatives of each compatible
## class as computed by the program compatible_classes.g.
## They are written in the abstract generators a and b.
## Note each element of actlist represents a
## pair of actions. The first is the action of H on G
## the second the action of G on H.
## -----
a := AbstractGenerator("a");;
b := AbstractGenerator("b");;
actlist := [
  [ [ a, b ],      [ a, b ],      [ a, b ],      [ a, b ] ],
  [ [ a, b ],      [ a, b ],      [ a, b ],      [ a, a^2*b ] ],
  [ [ a, b ],      [ a, b ],      [ a, a^2*b ],    [ a, b ] ],
  [ [ a, b ],      [ a, b ],      [ a, a^2*b ],    [ a^3, b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a, b ],      [ a, b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a, b ],      [ a, a^2*b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a, b ],      [ a^3, b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a, a^2*b ],   [ a, b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a, a^2*b ],   [ a, a^2*b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a, a^2*b ],   [ a^3, b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a^3, b ],     [ a, b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a^3, b ],     [ a, a*b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a^3, b ],     [ a, a^2*b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a^3, b ],     [ a^3, b ] ],
  [ [ a, b ],      [ a, a*b*a ],    [ a^3, b ],     [ a^3, a^2*b ] ],
  [ [ a, b ],      [ a*b^2, b ],     [ a, b ],      [ a, b ] ],
  [ [ a, b ],      [ a*b^2, b ],     [ a, b ],      [ a, a^2*b ] ],
  [ [ a, b ],      [ a*b^2, b ],     [ a, b ],      [ a^3, b ] ],
  [ [ a, b ],      [ a*b^2, b ],     [ a, a^2*b ],   [ a, b ] ],
  [ [ a, b ],      [ a*b^2, b ],     [ a, a^2*b ],   [ a^3, b ] ],
  [ [ a, b ],      [ a*b^2, b ],     [ a^3, b ],     [ a, b ] ],
  [ [ a, b ],      [ a*b^2, b ],     [ a^3, b ],     [ a, a^2*b ] ],
  [ [ a, a*b*a ],  [ a*b^2, b ],     [ a, b ],      [ a, b ] ],
  [ [ a, a*b*a ],  [ a*b^2, b ],     [ a, b ],      [ a, a^2*b ] ],
  [ [ a, a*b*a ],  [ a*b^2, b ],     [ a, b ],      [ a^3, b ] ],
  [ [ a, a*b*a ],  [ a*b^2, b ],     [ a, b ],      [ a^3, a^2*b ] ],
  [ [ a, a*b*a ],  [ a*b^2, b ],     [ a, a^2*b ],   [ a, b ] ],
  [ [ a, a*b*a ],  [ a*b^2, b ],     [ a, a^2*b ],   [ a^3, b ] ],
  [ [ a, a*b*a ],  [ a*b^2, b ],     [ a, a^2*b ],   [ a^3, a^2*b ] ],
  [ [ a, a*b*a ],  [ a*b^2, b ],     [ a^3, b ],     [ a, b ] ],
  [ [ a, a*b*a ],  [ a*b^2, b ],     [ a^3, b ],     [ a^3, a^2*b ] ],
  [ [ a, a*b*a ],  [ a*b^2, b ],     [ a^3, a^2*b ], [ a^3, b ] ],
  [ [ a*b, a*b*a ], [ b*a, a*b*a ],    [ b, a ],      [ a^3, a^2*b ] ]];;

## -----
## For each pair of actions compute the Tensor product and try
## and identify it.
## -----
for i in [1..Length(actlist)] do

```

```

## -----
## rewrite the action in terms of the generators of G and
## H.
## -----
a11 := MappedWord(actlist[i][1][1],[a,b],G.generators);
a12 := MappedWord(actlist[i][1][2],[a,b],G.generators);
a21 := MappedWord(actlist[i][2][1],[a,b],G.generators);
a22 := MappedWord(actlist[i][2][2],[a,b],G.generators);

ActHG := [ [ a11, a12 ],
           [ a21, a22 ]];

a11 := MappedWord(actlist[i][3][1],[a,b],H.generators);
a12 := MappedWord(actlist[i][3][2],[a,b],H.generators);
a21 := MappedWord(actlist[i][4][1],[a,b],H.generators);
a22 := MappedWord(actlist[i][4][2],[a,b],H.generators);

ActGH := [ [ a11, a12 ],
           [ a21, a22 ]];

## -----
## Find the tensor product and identify the group.
## -----
A := TensorProduct(G,H,true,ActGH,ActHG);

## -----
## The GroupId function takes too long with group 13 and 32
## Instead print out its size, is it abelian and orders of
## its elements.
## -----
if i <> 13 and i <> 32 then
  Gpid := GroupId(A);
  Print("i = ",i,"\tSize = ",Size(A),
        "\t Abelian = ",IsAbelian(A),
        "\t name = ",Gpid.names,"\n");
else
  ord := List(Elements(A),el->Order(A,el));
  Sort(ord);
  Print("i = ",i,"\tSize = ",Size(A),
        "\t Abelian = ",IsAbelian(A),
        "\tnorders = ",ord,"\n");
fi;
od;

```

Program: TProd_D4Q8_In2.g

```
LogTo("TProd_D4Q8_In2.out");
```

```

## -----
## @(#) TProd_D4Q8_In2.g 1.1 97/12/16
## @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.TProd_D4Q8_In2.g
## -----
## Author: A.~McDermott
## Email: aidan.mcdermott@ucg.ie
## -----
## For the first member of each class of noncompatible
## actions this program calculates the tensor product between D4
## and Q8 as defined by Niko Inassaridze.
## (Note the actions here are all non-compatible.)
##
## This program simply calculates the Tensor Product directly
## from the generators and relations presentation.
## -----

Read("/usr/users/aidan/reruns/ap1/ActSupport.g");

## -----
## Make the two groups, G = D4 and H = Q8.
## -----
F := FreeGroup(2);
x := F.1;; y := F.1;; a := F.2;;

relG := [ x^2, a^4, x*a*x^-1*a ];; G := F/relG;;
relH := [ a^4, y^2*a^-2, y*a*y^-1*a ];; H := F/relH;;

## -----
## In what follows G is sometimes referred to as S1 and
## H as S2. We need lists of elements etc.
## -----
S1 := G;; S2 := H;;
ES1 := Elements(S1);; ES2 := Elements(S2);;
lenES1 := Length(ES1);; lenES2 := Length(ES2);;

Rlist := [];
TS1 := Subgroup(S1,[IdWord]);
TS2 := Subgroup(S2,[IdWord]);

st := Size(S1)*Size(S2);
F := FreeGroup(st);;

## -----
## This is a list of representatives of each
## class as computed by the program actions.g.
## They are written in the abstract generators a and b.
## Note each element of actlist represents a
## pair of actions. The first is the action of H on G
## the second the action of G on H.
## -----
a := AbstractGenerator("a");;
b := AbstractGenerator("b");;

```

```

actlist := [
  [ [ a, b ], [ a, b ], [ a, b ], [ b, a ] ],
  [ [ a, b ], [ a, b ], [ a, a^2*b ], [ b, a^3 ] ],
  [ [ a, b ], [ a, b ], [ a, a^2*b ], [ a^3, a*b ] ],
  [ [ a, b ], [ a, b ], [ b, a ], [ a, b ] ],
  [ [ a, b ], [ a, b ], [ b, a ], [ a^3, a^2*b ] ],
  [ [ a, b ], [ a, a*b*a ], [ a, b ], [ b, a ] ],
  [ [ a, b ], [ a, a*b*a ], [ a, b ], [ a^3, a*b ] ],
  [ [ a, b ], [ a, a*b*a ], [ a, a^2*b ], [ b, a^3 ] ],
  [ [ a, b ], [ a, a*b*a ], [ a, a^2*b ], [ a^3, a*b ] ],
  [ [ a, b ], [ a, a*b*a ], [ b, a ], [ a, b ] ],
  [ [ a, b ], [ a, a*b*a ], [ b, a ], [ b, a ] ],
  [ [ a, b ], [ a, a*b*a ], [ b, a ], [ b, a^3 ] ],
  [ [ a, b ], [ a, a*b*a ], [ b, a ], [ a^3, a^2*b ] ],
  [ [ a, b ], [ a, a*b*a ], [ b, a ], [ a^2*b, a^3 ] ],
  [ [ a, b ], [ a, a*b*a ], [ a^3, b ], [ a*b, a^2*b ] ],
  [ [ a, b ], [ a, a*b*a ], [ a^3, a*b ], [ a, b ] ],
  [ [ a, b ], [ a, a*b*a ], [ a^3, a*b ], [ a, a*b ] ],
  [ [ a, b ], [ a, a*b*a ], [ a^3, a*b ], [ a, a^2*b ] ],
  [ [ a, b ], [ a, a*b*a ], [ a^3, a*b ], [ a^3, a*b ] ],
  [ [ a, b ], [ a, a*b*a ], [ a^3, a*b ], [ a^3, b*a ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a, b ], [ a, b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a, b ], [ a, a^2*b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a, b ], [ b, a ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a, b ], [ a^3, b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a, b ], [ a^3, a*b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a, a^2*b ], [ a, b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a, a^2*b ], [ b, a^3 ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a, a^2*b ], [ a^3, b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a, a^2*b ], [ a^3, a*b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ b, a ], [ a, b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ b, a ], [ a^3, a^2*b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ b, a ], [ a^2*b, a^3 ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a^3, b ], [ a, b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a^3, b ], [ a, a*b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a^3, b ], [ a, a^2*b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a^3, a*b ], [ a, b ] ],
  [ [ a, b ], [ a*b, a*b*a ], [ a^3, a*b ], [ a, a^2*b ] ],
  [ [ a, b ], [ a*b^2, b ], [ a, b ], [ b, a ] ],
  [ [ a, b ], [ a*b^2, b ], [ a, b ], [ a^3, a*b ] ],
  [ [ a, b ], [ a*b^2, b ], [ a, a^2*b ], [ b, a^3 ] ],
  [ [ a, b ], [ a*b^2, b ], [ a, a^2*b ], [ a^3, a*b ] ],
  [ [ a, b ], [ a*b^2, b ], [ b, a ], [ a, b ] ],
  [ [ a, b ], [ a*b^2, b ], [ b, a ], [ a^3, a^2*b ] ],
  [ [ a, b ], [ a*b^2, b ], [ b, a ], [ a^2*b, a^3 ] ],
  [ [ a, b ], [ a*b^2, b ], [ a^3, b ], [ a, a*b ] ],
  [ [ a, b ], [ a*b^2, b ], [ a^3, a*b ], [ a, b ] ],
  [ [ a, b ], [ a*b^2, b ], [ a^3, a*b ], [ a, a^2*b ] ],
  [ [ a, a*b*a ], [ a*b^2, b ], [ a, b ], [ b, a ] ],
  [ [ a, a*b*a ], [ a*b^2, b ], [ a, b ], [ a*b, a^2*b ] ],
  [ [ a, a*b*a ], [ a*b^2, b ], [ a, b ], [ a^3, a*b ] ],
  [ [ a, a*b*a ], [ a*b^2, b ], [ a, a^2*b ], [ b, a^3 ] ],
  [ [ a, a*b*a ], [ a*b^2, b ], [ a, a^2*b ], [ a*b, b ] ],
  [ [ a, a*b*a ], [ a*b^2, b ], [ a, a^2*b ], [ a^3, a*b ] ],
  [ [ a, a*b*a ], [ a*b^2, b ], [ b, a ], [ a, b ] ],
  [ [ a, a*b*a ], [ a*b^2, b ], [ b, a ], [ b, a^3 ] ],

```

```

    [ [ a, a*b*a ], [ a*b^2, b ], [ b, a ], [ a^3, a^2*b ] ],
    [ [ a, a*b*a ], [ a*b^2, b ], [ a*b, a^2*b ], [ a, b ] ],
    [ [ a, a*b*a ], [ a*b^2, b ], [ a*b, a^2*b ], [ b*a, a^2*b ] ],
    [ [ a, a*b*a ], [ a*b^2, b ], [ a*b, a^2*b ], [ a^3, b ] ],
    [ [ a, a*b*a ], [ a*b^2, b ], [ a^3, a*b ], [ a^3, b*a ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, b ], [ a, b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, b ], [ a, a^2*b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, b ], [ b, a ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, b ], [ a*b, a^2*b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, b ], [ a^3, a^2*b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, a^2*b ], [ a, b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, a^2*b ], [ b, a^3 ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, a^2*b ], [ a*b, b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, a^2*b ], [ a^3, b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, a^2*b ], [ a^3, a*b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a, a^2*b ], [ a^3, a^2*b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ b, a ], [ a, b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ b, a ], [ a^2*b, a^3 ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a*b, a^2*b ], [ a, b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a*b, a^2*b ], [ a^3, b ] ],
    [ [ a*b, a*b*a ], [ b*a, a*b*a ], [ a^3, a^2*b ], [ a, b ] ]];;

```

```

## -----
## For each pair of actions compute the Tensor product and try
## and identify it. (this is a problem for a couple of groups)
## -----
for kk in [1..Length(actlist)] do

    Rlist := [];

    ## -----
    ## rewrite the action in terms of the generators of G and
    ## H.
    ## -----
    a11 := MappedWord(actlist[kk][1][1],[a,b],G.generators);
    a12 := MappedWord(actlist[kk][1][2],[a,b],G.generators);
    a21 := MappedWord(actlist[kk][2][1],[a,b],G.generators);
    a22 := MappedWord(actlist[kk][2][2],[a,b],G.generators);

    ActHG := [ [ a11, a12 ],
               [ a21, a22 ]];

    a11 := MappedWord(actlist[kk][3][1],[a,b],H.generators);
    a12 := MappedWord(actlist[kk][3][2],[a,b],H.generators);
    a21 := MappedWord(actlist[kk][4][1],[a,b],H.generators);
    a22 := MappedWord(actlist[kk][4][2],[a,b],H.generators);

    ActGH := [ [ a11, a12 ],
               [ a21, a22 ]];

    ## -----
    ## Find the tensor product and identify the group.
    ## -----

```

```

for i in [1..lenES1] do
  for j in [1..lenES2] do

    ## -----
    ## These are Inassaridze's extra conditions
    ## for the non-compatible case.
    ## -----
    for k in [1..lenES1] do
      for l in [1..lenES2] do

        cmagd := GactsonH(S2,S1,ES2[j]^-1,ES1[k],ActHG);
        cmagd := ES1[i]^-1 * cmagd * ES1[i];
        cmagd := GactsonH(S2,S1,ES2[j],cmagd,ActHG);
        cmagd := ES1[i] * cmagd * ES1[i]^-1;

        cmahd := ES2[j]^-1 * ES2[l] * ES2[j];
        cmahd := GactsonH(S1,S2,ES1[i]^-1,cmahd,ActGH);
        cmahd := ES2[j] * cmahd * ES2[j]^-1;
        cmahd := GactsonH(S1,S2,ES1[i],cmahd,ActGH);

        cmagd2 := ES1[k]^-1 * ES1[i] * ES1[k];
        cmagd2 := GactsonH(S2,S1,ES2[l]^-1,cmagd2,ActHG);
        cmagd2 := ES1[k] * cmagd2 * ES1[k]^-1;
        cmagd2 := GactsonH(S2,S1,ES2[l],cmagd2,ActHG);

        cmahd2 := GactsonH(S1,S2,ES1[k]^-1,ES2[j],ActGH);
        cmahd2 := ES2[l]^-1 * cmahd2 * ES2[l];
        cmahd2 := GactsonH(S1,S2,ES1[k],cmahd2,ActGH);
        cmahd2 := ES2[l] * cmahd2 * ES2[l]^-1;

        for m in [1..lenES1] do
          if ( cmagd * ES1[m]^-1 in TS1 ) then cmagdi := m; fi;
          if ( cmagd2 * ES1[m]^-1 in TS1 ) then cmagd2i := m; fi;
        od;
        for m in [1..lenES2] do
          if ( cmahd * ES2[m]^-1 in TS2 ) then cmahdi := m; fi;
          if ( cmahd2 * ES2[m]^-1 in TS2 ) then cmahd2i := m; fi;
        od;
        Append(Rlist, [F.generators[lenES2*(i-1)+j] *
          F.generators[lenES2*(k-1)+1] *
          ((F.generators[lenES2*(cmagdi-1)+cmahdi] *
            F.generators[lenES2*(i-1)+j])^-1)]);

        Append(Rlist, [F.generators[lenES2*(i-1)+j] *
          F.generators[lenES2*(k-1)+1] *
          ((F.generators[lenES2*(k-1)+1] *
            F.generators[lenES2*(cmagd2i-1)+cmahd2i])^-1)]);

      od;
    od;

    ## -----
    ## These are the usual BL tensor product conditions.
    ## -----
    for k in [1..lenES2] do
      nnd := ES2[j] * ES2[k];

```

```

nam := GactsonH(S2,S1,ES2[j],ES1[i],ActHG);
nand := ES2[j] * ES2[k] * ES2[j]^-1;
for l in [1..lenES1] do
  if ( nam *ES1[l]^-1 in TS1 ) then nami := l; fi;
od;
for l in [1..lenES2] do
  if ( nnd * ES2[l]^-1 in TS2 ) then nndi := l; fi;
  if ( nand * ES2[l]^-1 in TS2 ) then nandi := l; fi;
od;
Append(Rlist,[F.generators[lenES2*(i-1)+nndi] *
              ((F.generators[lenES2*(i-1)+j] *
                F.generators[lenES2*(nami-1)+nandi])^-1)]);

od;
od;

for j in [1..lenES1] do
for k in [1..lenES2] do
  mmd := ES1[i] * ES1[j];
  mamd := ES1[i] * ES1[j] * ES1[i]^-1;
  man := GactsonH(S1,S2,ES1[i],ES2[k],ActGH);
  for l in [1..lenES1] do
    if ( mmd * ES1[l]^-1 in TS1 ) then mmdi := l; fi;
    if ( mamd * ES1[l]^-1 in TS1 ) then mamdi := l; fi;
  od;
  for l in [1..lenES2] do
    if ( man * ES2[l]^-1 in TS2 ) then mani := l; fi;
  od;
  Append(Rlist,[F.generators[lenES2*(mmdi-1)+k] *
                (( F.generators[lenES2*(mamdi-1)+mani] *
                  F.generators[lenES2*(i-1)+k])^-1)]);

od;
od;
od;

## -----
## Compute the Tensor product and print out details.
## -----
Print("The Tensor Product\n");
GTH := F/Rlist;

## -----
## Tietze transformations:
## -----
x := PresentationFpGroup(GTH);
TzGo(x);
TzGoGo(x);
y := FpGroupPresentation(x);

Print("Action no = ",kk,"\n");
Print("Size = ",Size(y),"\n");
Print("Abelian = ",IsAbelian(y),"\n");
Print("Cyclic = ",IsCyclic(y),"\n");
Print("Generators = ",y.generators,"\n");
Print("Relations = ",y.relators,"\n");
Print("GroupID = ",GroupId(y),"\n");

```

```
Print(" =====\n");  
od;
```


Appendix C

Programs for Chapter 4

This appendix lists the programs used to produce the tables and other results in Chapter 4. The following table lists all the programs associated with this appendix. Only those marked with the • symbol are actually printed here, the rest can be found on the accompanying diskette.

File		Function	Table/Page
p4.g	•	Computes tensor square for the three 3-generator non-abelian groups of order p^4 , for $3 \leq p \leq 1000$.	p 94
p42.g	•	Computes tensor square for the three 3-generator non-abelian groups of order 2^4 .	p 94
p3.g		Computes tensor square for the three 3-generator non-abelian groups of order 2^3 .	p 91

Program: p4.g

```
LogTo("p4.out");

## -----
## @(#) Q3.g 1.1 97/12/16
## @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.Q3.g
## -----
## Author: A.~McDermott
## Email: aidan.mcdermott@ucg.ie
## -----
## This program computes the Tensor Product for all 3 generator
## groups of order  $p^4$  where p is an odd prime less than 1000.
## The case when  $p=2$  is dealt with elsewhere.
```

```

##
## The list of group presentations was taken form Burnside's,
## "The Theory of Groups."
##
## It uses a modified version of the TensorSq function which
## exploits the meta-abelian structure of the groups under
## consideration. All three groups contain a normal subgroup
## of order  $p^2$  ( cyclic groups generated by a in the first
## two cases and the subgroups generated by a and b in the third).
## The remaining generators then provide coset representatives for
## the factor. It follows that when computing the words
##  $W(x,a,b)$  we need only consider  $x \in G$ .generators.
## The function TensorSqGen implements the TensorSq algorithm
## in this way.
## -----

RequirePackage("anupq");
Read("/usr/users/aidan/reruns/ap1/ActSupport.g");
Read("/usr/users/aidan/reruns/ap1/TensorSquareSupport.g");
Read("/usr/users/aidan/reruns/ap1/TensorSquareGen.g");

## -----
## Set this to zero just in case
## -----
CosetTableFpGroupDefaultMaxLimit := 0;

## -----
## All the groups will be given on four generators.
## -----
F := FreeGroup(4);
a := F.1;; b := F.2;; c := F.3;; d := F.4;;

## -----
## Loop through the primes and compute the tensor square.
## Note: Groups of order  $2^4$  are dealt with elsewhere.
## -----
for p in Primes do
  if p > 2 then

    rel2 := [ a^(p^2), b^p, c^p,
              c^-1*b*c*(b*a^p)^-1, b^-1*a*b*a^-1,
              c^-1*a*c*a^-1,d];

    rel4 := [ a^(p^2), b^p, c^p, c^-1*a*c*a^(-1*(1+p)),
              a^-1*b*a*b^-1, c^-1*b*c*b^-1,d ];

    rel9 := [ a^p, b^p, c^p, d^p, d^-1*c*d*(c*a)^-1,
              d^-1*b*d*b^-1, d^-1*a*d*a^-1,
              c^-1*b*c*b^-1, c^-1*a*c*a^-1, b^-1*a*b*a^-1 ];

    rels := [rel2,rel4,rel9];
  end if
end for

```

```

## -----
## Find the Tensor Square and identify the groups
## -----
for k in [1..3] do

    G := F/rels[k];
    A := [];

    A := TensorSqGen(G,false,p,60);
    Print("prime = ",p," gp = ",k,
          " Size = ",Size(A),
          " is ea = ",IsElementaryAbelian(A),"\\n");

    od;
    Print("=====\\n");
fi;
od;

```

Program: p42.g

```

LogTo("p42.out");

## -----
## @(#) Q3.g 1.1 97/12/16
## @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.Q3.g
## -----
## Author: A.~McDermott
## Email: aidan.mcdermott@ucg.ie
## -----
## This program computes the Tensor Product for all 3 generator
## group of order  $2^4$ . The cases when the order is  $p^4$  for
##  $p$  an odd prime are dealt with in the program p4.g
##
## The list of group presentations was taken form Burnside's,
## "The Theory of Groups."
## -----

RequirePackage("anupq");
Read("/usr/users/aidan/reruns/ap1/ActSupport.g");
Read("/usr/users/aidan/reruns/ap1/TensorSquareSupport.g");
Read("/usr/users/aidan/reruns/ap1/TensorSquare.g");

## -----
## Set this to zero just in case
## -----
CosetTableFpGroupDefaultMaxLimit := 0;

```

```

## -----
## All the groups will be given on four generators.
## -----
F := FreeGroup(4);
a := F.1;; b := F.2;; c := F.3;; d := F.4;;

## -----
## Loop through the primes and compute the tensor square.
## Note: Groups of order 2^4 are dealt with elsewhere.
## -----
for p in [2] do
  if p = 2 then

    rel2 := [ a^(p^2), b^p, c^p,
              c^-1*b*c*(b*a^p)^-1, b^-1*a*b*a^-1,
              c^-1*a*c*a^-1,d];

    rel4 := [ a^(p^2), b^p, c^p, c^-1*a*c*a^(-1*(1+p)),
              a^-1*b*a*b^-1, c^-1*b*c*b^-1,d ];

    rel9 := [ a^(p^2), b^(p^2), c^p, d, b^-1*a*b*a,
              b^2*a^-2, c^-1*b*c*b^-1, c^-1*a*c*a^-1 ];

    rels := [rel2,rel4,rel9];

    ## -----
    ## Find the Tensor Square and identify the groups
    ## -----
    for k in [1..3] do

      G := F/rels[k];
      A := [];

      A := TensorSq(G,false,p,60);
      Print("prime = ",p," gp = ",k,
            " Size = ",Size(A),
            " is ea = ",IsElementaryAbelian(A),"\\n");

    od;
    Print("=====\n");
  fi;
od;

```

Appendix D

Programs for Chapter 5

This appendix lists the programs and support functions used in Chapter 5. The following table lists all the programs associated with this appendix. Only those marked with the • symbol are actually printed here, the rest can be found on the accompanying diskette.

File		Function	Table/Page
comms.g	•	Functions to compute the commutator subgroup and centre of a finitely presented group.	
qcapable.g	•	Function to decide if a finitely presented group is capable mod q .	
subgroups.g	•	Program to calculate all the subgroups of the elementary abelian group of order 64.	
elabelian64.g		Program fragment to give all the subgroups of the elementary abelian group of order 64 written in abstract generators.	
pcomm2.g	•	Computes $[G, G]G^p$ for $p = 2$	Table 5.1
pcomm3.g		Computes $[G, G]G^p$ for $p = 3$	Table 5.1
pcomm5.g		Computes $[G, G]G^p$ for $p = 5$	Table 5.1
pcomm7.g		Computes $[G, G]G^p$ for $p = 7$	Table 5.1

Program: pcomm2.g

```
LogTo("pcomm2.out");
```

```
## -----  
## @(#) pcomm2.g 1.1 97/12/16  
## @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.pcomm2.g
```

```

## -----
## Author: A.~McDermott
## Email: aidan.mcdermott@ucg.ie
## -----
## For each the 2--group Q of order less than 32 this program
## decides if it is Capable mod 2 and if so computes all possible
## [G,G]G^2 (ie Frattini Subgroups) where Q = G/Z(G,2).
##
## It uses the commutator functions from comms.g and a list of
## all subgroups of the elementary abelian group of order 64
## (there are over 2800) of these and are computed elsewhere.
## Here they are read in form the file elabelian64.g.
## The function IsCapableModq is read from qcapable.g.
## -----
Read("comms.g");
Read("qcapable.g");
Read("genset2.g");
Read("elabelian64.g");

## Here q = 2 change this for other q's
q := 2;

RandomSeed(481913);

## Initialize some variables:
x := 0;; F:=0;; Firstparto:=0;; Firstpart:=0;; D:=0;; Fact:=0; FpInt:=0;;
gen:=0;; fact:=0;; subsgens64:=0;;
count := 0;
for pow in [1..4] do
  all2gps := AllTwoGroups(Size,2^pow);
  for s in [1..Length(all2gps)] do

    ## -----
    ## First find a presentation for the group all2gps[s]
    ## and identify the group.
    ## -----

    count      := count+1;

    sz         := Size(all2gps[s]);
    gen        := all2gps[s].abstractGenerators;
    rel        := all2gps[s].abstractRelators;
    F := FreeGroup(pow,"X");

    xgen := List(gen,x->MappedWord(x,gen,F.generators));
    xrel := List(rel,x->MappedWord(x,gen,F.generators));

    X := F/xrel;

    Xid := GroupId(X);
    Print("Group number ",count," is ",Xid.names,"\n");

    ## -----
    ## Next decide if the group is Capable mod q
    ## this is a bit redundant since CapableModq also
    ## computes D -- but it's fast enough here.

```

```

## -----
if CapableModq(X,q) then

## -----
## Taking F/R to be a presentation for X we compute
##   D = F/[R,F]R^q
##   CommD = [D,D]
##   HashD = D #^q D = [D,D]D^q
##   CenD = Z(D) the centre
##   PCenD = Z(D,q) the centre mod q
##   Int = HashD \cup PCenD the intersection
## -----
D := CoverfpGroup(F,xrel,q);

Print("Is abelian D? ",IsAbelian(D),"\\n");
Print("Order of D is ",Size(D),"\\n");
CommD := CommfpGroup(D);
ellist := Elements(CommD);
for el in Elements(D) do
  Append(ellist,[el^q]);
od;
ellist := Set(ellist);
HashD := Subgroup(D,ellist);

## -----
## This is a check : the presentation below assumes that
## HashD is abelian
## -----
pqD := Pq(D,rec(Prime:=q,ClassBound:=63));
pqCommD := CommutatorSubgroup(pqD,pqD);
ellist := pqCommD.generators;
for el in Elements(pqD) do
  Append(ellist,[el^q]);
od;
ellist := Set(ellist);
RemoveSet(ellist,pqD.identity);
pqHashD := Subgroup(pqD,ellist);

Print("Order of [D,D]D^q is ",Size(pqHashD),"\\n");
Print("[D,D]D^q is abelian?           ",IsAbelian(pqHashD),"\\n");
Print("[D,D]D^q is Elementary abelian? ",IsAbelian(pqHashD),"\\n");

CenD := CentrefpGroup(D);
pCenD := [];
for el in Elements(CenD) do
  if el^q in TrivialSubgroup(D) then
    Append(pCenD,[ el ] );
  fi;
od;
PCenD := Subgroup(D,pCenD);

inter := Intersection(Elements(HashD),Elements(PCenD));
Int := Subgroup(D,inter);
Print("Size of Int = ",Size(Int),"\\n");

## -----

```

```

## We want a presentation for the subgroup HashD = [D,D]D^q
## so that we can easily identify the subgroup Int inside.
## We start by finding a generating set for Int and then
## extending it to a generating set for HashD.
## We then build presentations built on these sets.
## -----
ellist := Elements(Int);
gens := [];
A := TrivialSubgroup(D);
while Size(A) < Size(Int) do      ## Loop until a generating set
  ELA := Set(Elements(A));
  for el in ELA do
    RemoveSet(ellist,el);
  od;
  pick := RandomList([1..Length(ellist)]);
  Append(gens,[ ellist[pick] ]);
  A := Subgroup(D,gens);
od;

NoIntGens := Length(gens);      ## Number of generators for Int
Firstparto := ShallowCopy(gens); ## a list of those generators.

## -----
## Continue to build the generating set for HashD.
## -----
ellist := Elements(HashD);
while Size(A) < Size(HashD) do
  ELA := Set(Elements(A));
  for el in ELA do
    RemoveSet(ellist,el);
  od;
  pick := RandomList([1..Length(ellist)]);
  Append(gens,[ ellist[pick] ]);
  A := Subgroup(D,gens);
od;

## -----
## the relators for Int (we know Int is elementary abelian)
## FpInt is then a finitely presented version of Int
## In particular we can find all its subgroups using the
## function LowIndexSubgroupsFpGroup.
## -----
FInt := FreeGroup(NoIntGens);
RInt := List(FInt.generators,x -> x^q );
for i in [1..NoIntGens] do
  ip1 := i + 1;
  for j in [ip1..NoIntGens] do
    Append(RInt,[ Comm(FInt.generators[i],FInt.generators[j]) ]);
  od;
od;

FpInt := FInt / RInt;
## -----
## The following function LowIndexSubgroupFpGroup will not
## work if FpInt is Elementary Abelian of order 64.
## In that case we use the previously worked out list

```



```

## subgps64 read in from file.
## -----
tobig := 0;
if IsElementaryAbelian(FpInt) and Size(FpInt) = 64 then
  tobig:=1;
elif IsElementaryAbelian(FpInt) and Size(FpInt) = 1024 then
  tobig:=2;
fi;

if tobig=0 then
  subs := LowIndexSubgroupsFpGroup(FpInt,
    TrivialSubgroup(FpInt),Size(FpInt));
elif tobig=1 then
  subs := subgps64;
fi;

## -----
## Now for a presentation for HashD = [D,D]D^q.
## FirstpartFHashD here refers to the first set of generators
## corresponding to Int.
## -----
FHashD := FreeGroup(Length(gens));

FirstpartFHashD := [];
for k in [1..NoIntGens] do
  Append(FirstpartFHashD,[FHashD.generators[k]]);
od;

RHashD := [];
for i in [1..Length(FHashD.generators)] do

  if i <= NoIntGens then
    Append(RHashD,[FHashD.generators[i]^q ] );
  else
    Append(RHashD,[FHashD.generators[i]^Order(HashD,gens[i])]);

    ## search for powers of generators of HashD in the Int part.
    ogm1 := Order(HashD,gens[i]) - 1;
    for k in [1..ogm1] do
      if gens[i]^k in Int then

        for el in Elements(FpInt) do
          x := MappedWord(el,FpInt.generators,Firstparto);
          if gens[i]^k*x^-1 in TrivialSubgroup(Int) then
            y := MappedWord(el,FpInt.generators,FirstpartFHashD);
            Append(RHashD,
              [FHashD.generators[i]^k*y^-1]);
          fi;
        od;
      fi;
    od;
  fi;
od;

## -----
## Warning:

```

```

## In general the RHashD developed here is not sufficient
## to produce a presentation. We make use of the fact
## that all the HashD we are interested in are abelian
## so we add the following relators.
##
## If the group you are interested in has non abelian HashD
## you will have to work out a presentation yourself.
## -----
for i in [1..Length(gens)] do
  ip1 := i + 1;
  for j in [ip1..Length(gens)] do
    Append(RHashD, [ Comm(FHashD.generators[i],
                        FHashD.generators[j]) ]);
  od;
od;
FpHashD := FHashD / RHashD;

## -----
## Make two copies of the generating sets for Int
## One in FpHashD and the other in HashD itself.
## -----
Firstpart := [];
Firstparto := [];
for k in [1..NoIntGens] do
  Append(Firstpart, [FpHashD.generators[k]]);
  Append(Firstparto, [gens[k]]);
od;

## -----
## Loop through all the subgroups of FpInt finding the
## corresponding subgroup in HashD and Int.
## -----
markers := [0,0,0,0,0,0,0,0,0,0,0];
if tobig=2 then
  subs := [TrivialSubgroup(FpInt)];
  subs[11] := FpInt;
  for j in [2,3,4,5,6,7,8,9,10] do
    subgens := []; k := 0; pkgp := TrivialSubgroup(FpInt);
    while k < (j-1) do
      pick := RandomList([1..Size(FpInt)]);
      if not Elements(FpInt)[pick] in pkgp then
        Append(subgens, [ Elements(FpInt)[pick] ]);
        pkgp := Subgroup(FpInt,subgens);
        k := k + 1;
      fi;
    od;
    subs[j] := pkgp;
  od;
fi;

while Sum(markers) < 11 do
  for kk in [1..Length(subs)] do
    gp := subs[kk];

    if tobig=1 then

```

```

        genlisto := List(gp,
                        x -> MappedWord(x,subs64gens,Firstparto));
else
    genlisto := List(gp.generators,
                    x -> MappedWord(x,FpInt.generators,Firstparto));
fi;
gpo := Subgroup(Int,genlisto);

## -----
## Compute \widetilde{gpo} that is all the elements d in
## D for which [d,x] \in gpo and d^q \in gpo.
## If we find an element d in \widetilde{gpo} which is not
## in PCenD then stop looking go to the next subgroup gp
## else we have found a candidate for G = HashD/gp.
## -----
fact := D/PCenD;
elmlist := List(Elements(fact),
                x->MappedWord(x,fact.generators,D.generators));

nextgp := 0;
for d in elmlist do
    ok := false;
    if not d^q in gpo then
        ok := true;
    fi;
    for x in elmlist do
        if not Comm(d,x) in gpo then
            ok := true;
        fi;
    od;
    if ok = false then
        if not d in PCenD then
            nextgp := 1;          ## That is go to next subgroup
        fi;
    fi;
od;

## -----
## Found a candidate Print out details.
## -----
if nextgp = 0 then

    if tobig=1 then
        genlist := List(gp,
                        x -> MappedWord(x,subs64gens,Firstpart));
    else
        genlist := List(gp.generators,
                        x -> MappedWord(x,FpInt.generators,Firstpart));
    fi;

    subgp := Subgroup(FpHashD,genlist);
    Fact := FpHashD/subgp;
    Print("Size of Factor = ",Size(Fact)," ");
    Print("Is abelian? ",IsAbelian(Fact)," \n");
    # I don't use the GroupId function - it tends to get confused.

```

```

# Can identify group from the IsAbelian and orders.
# gpid := GroupId(Fact);
# Print(" name = ",gpid.names,"\n");
elord := List(Elements(Fact),x -> Order(Fact,x));
Sort(elord);
Print("Order of elements are\n",elord,"\n");

if tobig=2 then
  markers[kk] := 1;
fi;
od;

if tobig=2 then
  markers[1] := 1;
  markers[11] := 1;

## Pick a new bunch
for j in [2,3,4,5,6,7,8,9,10] do
  subgens := []; k := 0; pkgp := TrivialSubgroup(FpInt);
  while k < (j-1) do
    pick := RandomList([1..Size(FpInt)]);
    if not Elements(FpInt)[pick] in pkgp then
      Append(subgens,[ Elements(FpInt)[pick] ]);
      pkgp := Subgroup(FpInt,subgens);
      k := k + 1;
    fi;
  od;
  subs[j] := pkgp;
od;
else
  ## Set all markers to 1 so we don't go down the loop again.
  for j in [1..11] do
    markers[j] := 1;
  od;
fi;
od;
Print(" =====\n");
od;
od;

```

Program: comms.g

```

# -----
# @(#) comms.g 1.1 97/12/16
# @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.comms.g
# -----

```

```

# Author: A.~McDermott
# Email: aidan.mcdermott@ucg.ie
# -----
# Computes the Commutator subgroup of a finitely presented group
# -----
CommfpGroup := function(G)
#
local M,N,          # Subgroups
      g, h,         # elements
      commlist;     # list of elements
#
commlist := [];
for g in G.generators do
  for h in G.generators do
    Append(commlist,[g*h*g^-1*h^-1]);
  od;
od;
M := Subgroup(G,commlist);
repeat
  N := M;
  for g in G.generators do
    Append(commlist,Generators( M^g ));
  od;
  M := Subgroup(G,commlist);
until M = N;
return M;
#
end;
# -----
# Computes the Centre of a finitely presented group
# -----
CentrefpGroup := function(G)
#
local c,           # return centre
      g, h,        # elements
      clist,       # list of elements
      TG,          # Trivial subgroup of G
      isin;        # logical is in centre
#
TG := TrivialSubgroup(G);
#
clist := [];
for g in Elements(G) do
  isin := true;
  for h in G.generators do
    isin := isin and (Comm(g,h) in TG);
  od;
  if isin then
    Append(clist,[g]);
  fi;
od;
c := Subgroup(G,clist);
return c;
#
end;

```

```

# -----
# Computes the p Cover of a finitely presented group
# -----
CoverfpGroup := function(arg)
#
local crel,      # relation list
      r,g,      # elements
      nargs,    # number of arguments
      F,        # Free group used to define G
      rels,     # relations defining G
      p,        # prime
      C;        # return cover

#
nargs := Length(arg);
F:=arg[1];
rels := arg[2];
p:=arg[3];
#
crel := [];
for r in rels do
  for g in F.generators do
    Append(crel,[Comm(g,r)]);
  od;
  Append(crel,[r^p]);
od;
crel := Set(crel);
RemoveSet(crel,F.identity);
C := F/crel;
return C;
end;

```

Program: qcapable.g

```

## -----
## @(#) qcapable.g 1.1 97/12/16
## @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.qcapable.g
## -----
## Author: A.~McDermott
## Email:  aidan.mcdermott@ucg.ie
## -----
## This function computes the "q-cover"  $D := F/[R,F]R^q$  for the
## finitely presented group  $F/R$ . It calculates  $R/[R,F]R^q$  and
##  $Z(D,q)$  and tests if they have the same order. If so then
## CapableModq returns true else false.
## -----
CapableModq := function(arg)
local G,          # the group - argument 1
      q,          # the prime q - argument 2

```

```

    F,          # Free group for the presentation of G
    rel,        # relators from G
    D,          # D := F/[R,F]R^q
    rrel,       # rel mapped into D
    RinD,       # R/[R,F]R^q as subgroup of D
    CenD,       # Centre of D
    qcenD,      # List of elements in CenD with x^q = 1.
    qCenD,      # Z(D,q)
    el,iscapable;

G := arg[1];
q := arg[2];

if (( not IsBound(G.isFpGroup) ) or ( G.isFpGroup = false ) ) then
  Print("G should be a finitely generated group\n");
fi;

## -----
## Work out the presentation of G
## -----
F := FreeGroup(Length(G.generators));
rel := List(G.relators,r->MappedWord(r,G.generators,F.generators));

## -----
## compute D := F/[R,F]R^q and RinD := R/[R,F]R^q
## -----
D := CoverfpGroup(F,rel,q);
rrel := List(rel,r->MappedWord(r,F.generators,D.generators));
RinD := Subgroup(D,rrel);

## -----
## Compute qCenD := Z(D,q) the centre of D mod q.
## -----
CenD := CentrefpGroup(D);
qcenD := [];
for el in Elements(CenD) do
  if el^q in TrivialSubgroup(D) then
    Append(qcenD,[el]);
  fi;
od;
qCenD := Subgroup(CenD,qcenD);

## -----
## If qCenD = RinD then capable mod q.
## -----
iscapable := Size(RinD) = Size(qCenD);
return iscapable;
end;

```

Program: subgroups.g

```
LogTo("subgroups.out");

## -----
## @(#) subgroups.g 1.1 97/12/16
## @(#) /usr/users/aidan/thesis/thesis2/SCCS/s.subgroups.g
## -----
## Author: A.~McDermott
## Email: aidan.mcdermott@ucg.ie
## -----
## This program computes all the subgroups of the elementary
## abelian group of order 64.
##
## It is the basis of the file elabelian64.g which is subsequently
## read and used by pcomm2.g.
##
## The program calculates the subgroups of size 1, 2 and 4
## directly and then use LowIndexSubgroupsFpGroup for
## subgroups of index 8 or less.
## -----

Read("comms.g");
F := FreeGroup(6);;
a := F.1;; b := F.2;; c := F.3;; d := F.4;; e := F.5;; f := F.6;;

rel := [ a^2, b^2, c^2, d^2, e^2, f^2,
          Comm(a,b), Comm(a,c), Comm(a,d), Comm(a,e), Comm(a,f),
          Comm(b,c), Comm(b,d), Comm(b,e), Comm(b,f),
          Comm(c,d), Comm(c,e), Comm(c,f),
          Comm(d,e), Comm(d,f),
          Comm(e,f) ];;

G := F/rel;;
Print("Size of G is ",Size(G),"\\n");;

xx := LowIndexSubgroupsFpGroup(G,TrivialSubgroup(G),8);;

## Subgroups of size one.
ellist := Elements(G);
OneSub := List(ellist,x -> Subgroup(G,[x]));

## Subgroups of size two.
TwoSub := [];
Append(TwoSub,[Subgroup(G,[ ellist[2], ellist[3] ])]);
for i in [2..Length(ellist)] do
  tjj := i + 1;
  for j in [tjj..Length(ellist)] do
    already := false;
    count := 1;
    while already = false and count <= Length(TwoSub) do
      gp := TwoSub[count];
      if ellist[i] in gp and ellist[j] in gp then
        already := true;
      fi;
      count := count + 1;
    end while;
  end for;
end for;
```



```
    od;

    if already = false then
      Append(TwoSub,[Subgroup(G,[ ellist[i], ellist[j] ])]);
    fi;
  od;
od;
Append(xx,OneSub);
Append(xx,TwoSub);

for i in [1..Length(xx)] do
  Print("i = ",i," ",xx[i].generators,"\n");
od;
```

Bibliography

- [1] Bacon, M.R.: On the nonabelian tensor square of a nilpotent group of class two. *Glasgow Math. J.* **36** no. 3, 291–296 (1994)
- [2] Bacon, M.R., Kappe, L.-C.: The nonabelian tensor square of a 2-generator p -group of class 2. *Arch. Math.* (Basel) **61** no. 6, 508–516 (1993)
- [3] Bacon, M.R., Kappe, L.-C., Morse, R.F.: On the nonabelian tensor square of a 2-Engel group. *Arch. Math.* (to appear).
- [4] Berkovich, Ya.G.: On the order of the commutator subgroup and the Schur multiplier of a finite p -group. *J. Algebra* **144** no. 2, 269–272 (1991)
- [5] Brown, R.: A nonabelian tensor product of groups. *Algebra-Tagung Halle 1986*, 59–72, Wissensch. Beitr., 33, Martin-Luther-Univ. Halle-Wittenberg, Halle (Saale) (1987)
- [6] Brown, R.: Coproducts of crossed P -modules: applications to second homotopy groups and to the homology of groups. *Topology*, **23** 337–345 (1984)
- [7] Brown, R.: q -perfect groups and universal q -central extensions. *Publ. Mat.* **34** 291–297 (1990)
- [8] Brown, R., Johnson, D.L.; Robertson, E.F.: Some computations of non-abelian tensor products of groups. *J. Algebra* **111** no. 1, 177–202 (1987)
- [9] Brown, R., Loday, J.-L.: Van Kampen theorems for diagrams of spaces. *Topology* **26** no.3, 311–335 (1987)
- [10] Burns, J., Ellis, G.: Inequalities for Baer invariants of finite groups. *Canadian Math. J* (to appear).
- [11] Burns, J., Ellis, G., MacHale D., Ó' Murchú, P., Sheehy R., Wiegold, J.: Lower central series of groups with small upper central factors. *Proc. Royal Irish Acad.* (to appear)
- [12] Burnside, W.: Theory of groups of finite order. 1st edn., Cambridge University Press, Cambridge 1897
- [13] Conduch, D., Rodriguez-Fernndez, C.: Nonabelian tensor and exterior products modulo q and universal q -central relative extensions. *J. Pure Appl. Algebra* **78** no. 2, 139–160 (1992)

- [14] Dennis, R.K.: In search of new homology functors having a close relationship to K -theory, *Cornell University preprint* (1976)
- [15] Ellis, G.: A bound for the derived and frattini subgroups of a prime-power group, *Proc. American Math. Soc.*, (to appear).
- [16] Ellis, G.: A nonabelian tensor product of Lie algebras. *Glasgow Math. J.* **33** no. 1, 101–120 (1991)
- [17] Ellis, G.: On groups with a finite nilpotent upper central quotient, *Arch. Math* (to appear).
- [18] Ellis, G.: On Peiffer central series. *Glasgow Math. J.*, (to appear).
- [19] Ellis, G.: On the tensor square of prime power groups. *Arch. Math.*, **66** 467-469 (1996)
- [20] Ellis, G.: Homotopical aspects of Lie algebras. *J. Austral. Math. Soc. Ser. A* **54** no. 3, 393–419 (1993)
- [21] Ellis, G.: Nonabelian exterior products of groups and exact sequences in the homology of groups. *Glasgow Math. j.* **29** 13–19 (1987)
- [22] Ellis, G.: The second homology group of a group with integer coefficients modulo q . *preprint* (1988).
- [23] Ellis, G.: On the higher universal quadratic functors and related computations. *J. Algebra* **140** no. 2, 392–398 (1991)
- [24] Ellis, G.: Tensor products and q -crossed modules. *J. London Math. Soc.* (**2**) **51** no. 2, 243–258 (1995)
- [25] Ellis, G.: The nonabelian tensor product of finite groups is finite. *J. Algebra* **111** no. 1, 203–205 (1987)
- [26] Ellis, G., Rodriguez-Fernndez, C.: An exterior product for the homology of groups with integral coefficients modulo p . *Cahiers Topologie Gom. Diffrentielle Catgoriques* **30** (1989), no. 4, 339–343.
- [27] Ellis, G., Leonard, F.: Computing Schur multipliers and tensor products of finite groups. *Proc. Roy. Irish Acad.* **95A** no. 2, 137–147 (1995)
- [28] Gilbert, N.D.: The nonabelian tensor square of a free product of groups. *Arch. Math. (Basel)* **48** no. 5, 369–375 (1987)
- [29] Gilbert, N.D., Higgins, P.J.: The nonabelian tensor product of groups and related constructions. *Glasgow Math. J.* **31** no. 1, 17–29 (1989)
- [30] Guin, D.: Cohomologie non-abeliennes des groupes, *J. Pure Appl. Algebra*, **50** 109-138 (1988)

- [31] Guin, D.: Cohomologie des algèbres de Lie croisés et K -théorie de Milnor additive. (French) [Cohomology of crossed Lie algebras and additive Milnor K -theory.] *Ann. Inst. Fourier* (Grenoble) **45** no.1, 93–118 (1995)
- [32] Hall, M., Senior, J.K.: The groups of order 2^n ($n \leq 6$). MacMillian, New York 1964.
- [33] Hannebauer, T.: On nonabelian tensor squares of linear groups. *Arch. Math.* (Basel) **55** no. 1, 30–34 (1990)
- [34] Hartl, M.: The nonabelian tensor square and Schur multiplier of nilpotent groups of class 2. *J. Algebra* **179** no.2, 416–440 (1996)
- [35] Higgs, R.J.: Projective characters of degree one and the inflation-restriction sequence. *J. Australian Math. Soc.* **46** 272–280 (1989)
- [36] Hilton, P., Stammbach, U.: A Course in Homological Algebra. (Graduate Texts in Mathematics 4), Springer-Verlag, Heidelberg - New York - Berlin 1970
- [37] Inassaridze, N.: Finiteness of a non-abelian tensor product of groups. *Theory Appl. Categ.* **2** No. 5, 55–61 (1996)
- [38] Inassaridze, N.: Nonabelian tensor products and nonabelian homology of groups. *J. Pure Appl. Algebra* **112** no. 2, 191–205 (1996)
- [39] Jacobson, N.: Lie Algebras. (Interscience Tracts in Pure and Applied Mathematics 10), Interscience Publishers, New York – London 1963
- [40] Johnson, D. L.: Noncancellation and nonabelian tensor squares. *Group theory (Singapore, 1987)*, 405–408, de Gruyter, Berlin-New York, (1989)
- [41] Johnson, D. L.: The nonabelian tensor square of a finite split metacyclic group. Groups–St. Andrews 1985. *Proc. Edinburgh Math. Soc.* (2) **30** no. 1, 91–95 (1987)
- [42] Kappe, L.-C.: Nonabelian tensor products of groups: the commutator connection. *Proc. Groups-St. Andrews 1997 at Bath* LMS Lecture Notes, (to appear).
- [43] Kappe, L.-C., Sarmin, N.H., Visscher, M.P.: Two-generator two-groups of class two and their nonabelian tensor squares. *Glasgow Math. J.* (to appear).
- [44] Karpilovsky, G.: The Schur multiplier. (LMS monographs New Series 2), Oxford University Press 1987
- [45] MacHale, D., Ó' Murchú, P.: Commutator subgroups of groups with small central factor groups. *Proc. Royal Irish Acad.*, **93A** , 123-129 (1993)
- [46] Mac Lane S.: Categories for the working mathematician. (Graduate Texts in Mathematics, **Vol. 5**), Springer 1971

- [47] Miller, C.: The second homology group of a group. *Proc. Amer. Math Soc.* **3** 588–595 (1952)
- [48] Milnor J.: Introduction to algebraic K -Theory. Princeton University Press 1971
- [49] Robinson, D.J.S.: A course in the theory of groups. (Second Edition, Graduate Texts in Mathematics **80**), Springer 1991
- [50] Rocco, N.R.: A presentation for a crossed embedding of finite solvable groups. *Comm. Algebra* **22** no. 6, 1975–1998 (1994)
- [51] Rocco, N.R.: On a construction related to the nonabelian tensor square of a group. *Bol. Soc. Brasil. Mat. (N.S.)* **22** no. 1, 63–79 (1991)
- [52] Rodriguez-Fernandez, C.; Campo A.M.: The nonabelian tensor product. (Spanish) *Proceedings of the XVth Portuguese-Spanish Conference on Mathematics*, Vol. I (Portuguese) (vora, 1990), 161–166, Univ. vora, vora, (1991)
- [53] Rotman, J.J.: An introduction to algebraic topology. (Graduate Texts in Mathematics 119), Springer-Verlag New York Inc. 1988
- [54] Schönert M., et al.: GAP – groups, algorithms and programming. (manual) Lehrstuhl D für Mathematic, RWTH, Aachen, Germany, 1995
- [55] Steinberg, R.: Générateurs, relations et revêtements de groupes algebriques. *Colloq. Théorie des groupes algebriques*, Bruxelles 113–127 1962.
- [56] Whitehead, G.W.: Elements of homotopy theory. (Graduate Texts in Mathematics 61), Springer-Verlag, Heidelberg - New York - Berlin, 1978