Indian J. pure appl. Math., 35(3) : 333-345, March 2004

ENUMERATION OF 1-TRUNCATED SIMPLICIAL GROUPS OF LOW ORDER

MURAT ALP*

Dumlupinar University, Mathematics Department, Art and Science Faculty, Kütahya - Turkey
e-mail: malp@dumlupinar.edu.tr

(*Received* 28 *November* 2002; *after final revision* 30 *July* 2003; *accepted* 18 *November* 2003)

In this paper we describe a share package of functions for computing with finite, permutation crossed modules, 1-cat groups, 1-truncated simplicial groups and their morphisms, written using the GAP[6] group theory programming language. The category **XMod** of crossed modules is equivalent to the category Cat1 of 1-cat groups and the category **Simp** of 1-truncated simplicial groups. We include functions emulating the functors between these categories. We include a table of the 350 isomorphism classes of 1-truncated simplicial structures on groups of order at most 30.

Key Words: Crossed Module; 1-Truncated Simplicial Groups; 1-Cat Group; Moore Complex

INTRODUCTION

In this paper we describe a share package XMOD[2] for the GAP[1] (ref. 6) group theory language which enables computations with the equivalent notions of finite, permutation crossed modules, 1-cat groups and SIMPACTN[1] to compute 1-truncated simplicial groups. We also present the results of the computation of all isomorphism classes of 1-truncated simplicial group structures on groups of order at most 30.

The term crossed module was introduced by Whitehead[7]. We shall use right actions since this is the convention used by most computational group packages.

In[5] Loday reformulated the notion of a crossed module as a 1-cat group and a simplicial group whose Moore complex is of length one. Also showed that the category XMod of crossed modules is equivalent to the category Cat1 of 1-cat groups and the category Simp of simplicial group which was called 1-truncated simplicial group whose Moore complex is of length one.

In section 2 we recall the basic properties of crossed modules, their morphisms, 1-truncated simplicial group, their morphisms and 1-cat groups and their morphisms. In Section 3 we describe the implementation of these structures in GAP. In Section 4, we presented some of the algorithms for 1-truncated simplicial groups. In Section 5, we tabulate, for groups G of order at most 30, the order of Eng(G); the number of idempotent endomorphisms; the number of 1-truncated simplicial structures on G; and the number of isomorphism classes of these structures.

*Author was supported by TUBITAK BAYG (NATO-B2)
†Gap Group home page: http://www-history.mcs.st-and.ac.uk/~ gap/

2. Crossed Modules, 1-Truncated Simplicial Groups and 1-Cat Groups

In this section we recall the descriptions of three equivalent categories: The category XMod of crossed modules, the category **Simp** of 1-truncated simplicial groups whose Moore complex is of length one and the category Cat1 of 1-cat groups. We also describe functors between these categories which exhibit the equivalences.

A crossed module $\chi = (\partial : S \to R)$ consists of a group homomorphism ∂, called the boundary map of χ , together with an action $\alpha : R \to \mathrm{Aut}(S)$ satisfying, for all $s, s' \in S$ and $r \in R$,

$$\text{XMod 1:}\quad \partial(s^r) = r^{-1}(\partial s)\, r$$

$$\text{XMod 2:}\quad s^{\partial s'} = s^{1-1}\, ss'.$$

One can deduce from the axioms that the kernel of ∂ is abelian.

Standard constructions for crossed modules include the following:

1. Let S be abelian group and any homomorphism $\partial : S \to R$ provides a crossed module with R acting trivially on S.

2. A conjugation crossed module is an inclusion of a normal subgroup $S \trianglelefteq R$, where R acts on S by conjugation.

3. A central extension crossed module has as boundary map a surjection $\partial : S \to R$ with central kernel, where $r \in R$ acts on S by conjugation with $\partial^{-1} r$.

4. An automorphism crossed module has as range a subgroup R of the automorphism group $\mathrm{Aut}(S)$ of S which contains the inner automorphism group of S. The boundary maps $s \in S$ to the inner automorphism of S by s.

5. An R-Module crossed module has an R-module as source and ∂ is the zero map.

6. The direct product $\chi_1 \times \chi_2$ of two crossed modules has source $S_1 \times S_2$, range $R_1 \times R_2$ and boundary map $\partial_1 \times \partial_2$ with R_1, R_2 acting trivially on S_2, S_1 respectively.

A morphism between two crossed modules χ_1 and χ_2 is a pair (σ, ρ), where $\sigma : S_1 \to S_2$ and $\rho : R_1 \to R_2$ are homomorphisms satisfying

$$\partial_i \sigma = \rho\, \partial_1, \quad \sigma(s^r) = (\sigma s)^{\rho r}.$$

When $\chi_2 = \chi_1$ and σ, ρ are automorphisms then (σ, ρ) is an automorphism of χ_1. The group of automorphisms is denoted by $\mathrm{Aut}(\chi_1)$.

Consider a simplicial group

$$(\ldots \Rrightarrow G \overset{s,b}{\rightrightarrows} N)$$

where N is ıdentified with a subgroup of G by the degeneracy map $\sigma : N \to G$. The relations among face and degeneracy maps in a simplicial group imply $S_{|N} = b_{|N} = id_{|N}$. If the Moore complex of this simplicial group is of length one, that is

$$\cdots 1 \to 1 \cdots\cdots \to 1 \to \ker s \to N$$

then the face maps s and b satisfy the following peoperty: the group [ker s, ker b] generated by the commutators $[x, y] = xyx^{-1}y^{-1}, x \in \ker s, y \in \ker b$ is trivial[5].

Again Loday[5] reformulated the notion of a crossed module as a 1-cat group, namely a group G with a pair of homomorphisms $t, h : G \to G$ having a common image R and satisfying certain axioms. We find it convenient to define a 1-cat group $C = (e; t, h : G \to R)$ as having source group G, range group R, and three homomorphisms: two surjections $t, h : G \to R$ and an embedding $e : R \to G$ satisfying:

Cat 1: $te = he = id_R$

Cat 2: $[\ker t, \ker h] = \{1_G\}$.

The maps t, h are usually referred to as the source and target, but we choose to call them the tail and head of C, because source is the GAP term for the domain of a function.

A morphism $C_1 \to C_2$ of 1-cat groups is a pair (γ, ρ) where $\gamma : G_1 \to G_2$ and $\rho : R_1 \to R_2$ are homomorphisms satisfying

$$h_2\gamma = \rho h_1, \quad t_2\gamma = \rho t_1, \quad e_2\rho = \gamma e_1. \qquad \text{... (1)}$$

The crossed module χ associated to C has $S = \ker t$ and $\rho = h_{|S}$. The 1-cat group associated to χ has $G = R \ltimes S$, using the action from χ, and

$$t(r, s) = r, h(r, s) = r(\partial s), er = (r, 1).$$

Loday showed that a 1-cat group and a crossed module are equivalent to a simplicial group whose Moore complex is of length one as follows. Recall from[5] that, if K_* is a simplicial group, the Moore complex of K_* is obtained by taking for each n the subgroup $\bigcap_{i=1}^{n} \ker d_i$ of K_n; the restriction of d_0 to this subgroup is the differential of the complex. The homology groups of the Moore complex are the homotopy groups of the geometric realization $|K_*|$.

Starting from the category we obtain a simplicial set by taking the nerve. In fact this simplicial set is a simplicial group because the category is a group object in the category of categories. Its Moore complex is $\cdots 1 \to 1 \to M \to N$, which is of length 1.

Suppose that the Moore complex of K_* is of length one, that is

$$\cdots 1 \to 1 \to \ker d_1 \to K_0.$$

There is a 1-cat group associated to this situation. Put $G = K_1$ and N = image of K_2 in K_1 by the degeneracy map. The structural morphisms s and b are given by $s = d_1, b = d_0$. Axiom Cat1 follows from the relations between face and degeneracy map. To prove axiom Cat2 it is sufficient to see that for $x \in \ker d_1$ and $y \in \ker d_0$ the element $[s_0(x), s_0(y)\, s_1(y)^{-1}]$ of K_2 is in fact in $\ker d_1 \cap \ker d_2$ and its image by d_0 is $[x, y]$. As $\ker d_1 \cap \ker d_2 = 1$ it follows that $[\ker d_0, \ker d_1] = 1$.

3. GAP IMPLEMENTATION

The group theory program GAP[6] is designed to facilitate the implementation of new structures as record types with their own output form. We implement crossed modules, their morphisms, 1-cat groups, their morphisms and 1-truncated simplicial groups, their morphisms. We implement a crossed module $\chi = (\partial : S \to R)$ as a record with fields:

X.source	the source group S of ∂
X.boundary	the homomorphism ∂
X.range	the range group R of ∂
X.aut	a group of automorphisms of S
X.action	the homomorphism α from R to X.aut
X.isXMod	a boolean flag, normally true
X.isDomain	always true
X.operations	a special set of operations XModOps
X.name	a concatenation of the names of S and R.

Further fields, such as .isConjugationXMod, are added where appropriate.

The operations record XModops includes functions for equality; size; list of elements; a special output form; and various functions for the actor square.

A morphism mor = (σ, ρ) of crossed modules is implemented as a record with fields:

mor.source	the source crossed module X
mor.range	the range crossed module Y
mor.sourceHom	the homomorphism σ from X.source to Y.source
mor.rangeHom	the homomorphism ρ from X.range to Y. range
mor.isXModMorphism	a Boolean flag, normally true
mor.operations	a special set of operations XModMorphismOps
mor.name	a concatenation of the names of X and Y.

The operations record XModMorphismo includes functions for equality; kernel and image; composite and inverse morphism; and tests sucn as IsEpimorphism.

We implement a 1-cat group $C=(e;t,h:G\to R)$ as a record C with fields:

C.source	the source group G
C.range	the range group R
C.tail	the tail homomorphism t
C.head	the head homomorphism h
C.embedRange	the embedding e of R in G
C.kernel	a permutation group S isomorphic to the kernel of t
C.embedKernel	the isomorphism $\varepsilon: S\to \ker t$
C.boundary	the restriction ∂ of h to S
C.isDomain	set true
C.operations	a special set of operations Cat1Ops
C.name	a concatenation of the names of G and R
C.isCat1	a boolean flag, normally true

A morphism mor = (γ,ρ) of 1-cat groups is a record with fields similar to those of a morphism of crossed modules.

We implement a 1-truncated simplicial group $S=(s,b:G\to R)$ as a record S with fields:

S.source	the source G
S.range	the range R
S.faceop01	the first face operator d_0
S.faceop11	the second face operator d_1
S.degop00	the degeneracy map R in G
S.kernel	a permutation group isomorphic to the kernel of t
S.embedKernel	the inclusion of the kernel in G
S.boundary	the restriction of d_1 to the kernel
S.isDomain	set true
S.operations	a special set of operations SimplicialGroupOps
S.name	a concatenation of the names of the source and range
S.isSimplicial	a boolean flag, normally true

Further fields, such as .isConjugationXMod, are added where appropriate. The functors providing the equivalence between the categories Cat1, Simp and XMod are implemented as functions XModCat1(C), XModMorphism-SimpMorphism(mor), Cat1XMod(X), SimpMorphismXModMorphism (mor), XModSimp(S) and SimpXMod(X). The third of these calculates the semidirect product $R \ltimes S$ and then finds a suitable isomorphic permutation group G to act as the source, producing a SemidirectPair. In order to minimise the degree of G it is preferable to start with C when a representation for C is known and then construct χ.

Example 3.1 — Let G be the group $c3^2 \ltimes c2$ and R its normal subgroup s_3. 1-truncated simplicial group $\mathcal{S}$ is given as follows:

3.1. Simplicial Group

This function constructs a 1-truncated simplicial group $\mathcal{S}$ from a group G and a pair of endomorphisms, the first and second face operators of $\mathcal{S}$.

```
gap> G := Group((1,2,3), (4,5,6), (2,3)(5,6));
    Group((1,2,3), (4,5,6), (2,3)(5,6))
    gap> G.name := "c3^2|Xc2";;
    gap> s3 := Subgroup(G, [(4,5,6), (2,3)(5,6)]);;
    gap> s3.name := "s3";;
    gap> t := GroupHomomorphismByImages(G,s3,G.generators,
    [( ), (4,5,6), (2,3)(5,6)]);
    GroupHomomorphismByImages( c3^2|X c2, s3,
    [ (1,2,3), (4,5,6), (2,3(5,6) ], [( ), (4,5,6), (2,3)(5,6) ] )
    gap> S := Simp(G,t,t);
    Simplicial group [c3^2|X c2 ==> s3]
```

3.2 IsSimplicial Group(S)

This function checks that the axioms of a 1-truncated simplicial group are satisfied and that the main fields of a 1-truncated simplicial group record exist.

```
gap> IsSimplicialGroup(S);
  true

gap> SimplicialGroupPrint(S);

Simplicial Group [c3^2|Xc2 ==> s3] :-
: source group has generators:
    [ (1,2,3), (4,5,6), (2,3)(5,6) ]
: range group has generators:
    [ (4,5,6), (2,3)(5,6) ]
: faceOp01 homomorphism maps source generators to:
    [( ), (4,5,6), (2,3) (5,6) ]
: faceOp011 homomorphism maps source generators to:
    [ ( ), (4,5,6), (2,3) (5,6) ]
: range embedding maps range generators to:
    [ (4, 5, 6), (2, 3) (5, 6) ]
```

```
: kernel has generators:
    [(1, 2, 3) ]
: boundary homomorphism maps generators of kernel to:
    [ ( ) ]
: kernel embedding maps generators of kernel to:
    [ (1, 2, 3) ]
```

3.3. XModSimp

This function acts as the functor from the category of 1-truncated simplicial groups to the category of crossed modules.

```
gap> X := XModSimp(S) ;
Crossed module [ker([c3^2|Xc2 ==> s3]) -> s3]
gap> XModPrint(X);

Crossed module [ker([c3^2|Xc2 ==> s3]) -> s3] :
: Source group has parent (c3^2|Xc2) and has generators:
    [ (1, 2, 3) ]
: Range group has parent (c3^2|Xc2) and has generators:
    [ (4,5,6), (2,3) (5,6) ]
: Boundary homomorphism maps source generators to:
    [ () ]
: Action homomorphism maps range generators to automorphisms:
    (4,5,6) -> {source gens -> [ (1, 2, 3) ] }
    (2,3)(5,6) -> {source gens -> [ (1, 3, 2) ] }
These 2 automorphisms generate the group of automorphisms.
```

3.4. SimpXMod

```
gap> SimpXMod(X);
    Simplicial group [c3^2|Xc2 ==> s3]
    gap> SimplicialGroupPrint(last);

    Simplicial Group [c3^2|Xc2 ==> s3] :
    : source group has generators:

      [ (1,2,3), (4,5,6), (2,3)(5,6) ]
    : range group has generators:
      [ (4,5,6), (2,3)(5,6) ]
    : faceOp01 homomorphism maps source generators to:
      [( ), (4, 5, 6), (2, 3)(5, 6) ]
    : faceOp11 homomorphism maps source genrators to:
      [( ), (4, 5, 6), (2, 3)(5, 6) ]
    : range embedding maps range generators to:
      [(4, 5, 6), (2, 3)(5, 6) ]
    : kernel has generators:

      [(1, 2, 3) ]
    : boundary homomorphism maps generators of kernel to:
      [ () ]
    : kernel embedding maps generators of kernel to:
      [ (1, 2, 3) ]
```

: associated crossed module is Crossed module
[ker([c3^2|Xc2 ==> s3]) → s3]

4. Outline Algorithms

4.1. Algorithm for Simp

The function Simp is called as:

gap> Simp (G, d0, dl [, s0]) ;

The function requires three parameters: a group *D*, and the face operators d_0, d_1. The degeneracy operator *s*0 is an optional fourth parameter which is required when *R* is not a subgroup of *G*. As output, the function returns a 1-truncated simplicial group with fields.

Step 1 : Check that there are three parameters and that the first argument is a permutation group.

Step 2 : Check that *d*0 and *d*1 are homomorphisms with source *G* and with a common range *R*.

Step 3 : Set up the record fields.

Step 4 : Call the IsSimp function to verify the axioms.

4.2. Algorithm for IsSimp

The function IsSimp is called as:

gap> IsSimp (S);

The function returns true when the input parameter *S* is a 1-truncated simplicial group and false otherwise. The function checks that the main fields of a 1-truncated simplicial group exist, and that the axioms of 1-truncated simplicial groups are satisfied.

Step 1 : Check that *S* is a record structure, that fields S.source and S.range exist, and that these are permutation groups.

Step 2 : Check that S.faceOp01 and S.faceOp11 exist, and that these are group homomorphisms.

Step 3 : Check that S.embedKernel, S.kernel and S.boundary exist, with correct range and source.

Step 4 : Check that 1-truncated simplicial group conditions are satisfied.

Step 5 : Add field .isSimp to *S*.

4.3. Algorithm for SimpMorphism

The function SimpMorphism is called as:

gap> SimpMorphism(S, T, L);

The function SimpMorphism requires as parameters two 1-truncated simplicial groups and a two-element list containing the source and range homomorphisms. As output, it sets up the required fields for a morphism mor.

4.4. Algorithm for SimpXMod

The function SimpXMod is called as:

gap> SimpXMod(X);

This function implements the functor XMod → Simp.

Step 1 : Call IsXMod on the argument.

Step 2 : Call Cat1XMod to construct 1-cat group.

Step 3 : If X.action is trivial then the source group is constructed using the direct product $G = R \times S$. If X.action is not trivial the source group is constructed as a permutation representation G of $R \ltimes S$ using the function SemidirectPair(C).

Step 4 : The face operators and degeneracy operator are defined.

Step 5 : Call Simp(G, d0, d1, s0);

Step 6 : Add fields X.simp := S and S.xmod := X.

The procedure for XModSimp is similar.

4.5. All Simps

We[3,4] enumerated Whitehead groups and 1-cat groups of low order in respectively. Using the same technique we enumerated 1-truncated simplicial groups of low order and found same results using the function AllSimps[1]. A list L of 1-truncated simplicial groups with source S is initialized and a list of representatives of the nontrivial conjugacy classes of subgroup of S is selected as $\{L_1, L_2, ..., L_n\}$. For each L_i all idempotent endomorphism $\pi : S \to L_i$ are constructed and the image of the generators of S stored in a list K_i. These π are face operators which determine a 1-truncated simplicial group $\mathcal{S}$. This $\mathcal{S}$ is compared with L.

4.6. Timed Runs

We now present average times, in seconds, for the calculation of 1-truncated simplicial groups and crossed modules. Computations were performed on a DEC3000 Model 300LX Digital alpha 64-bit workstation running xgap with 20M memory.

1-truncated Simplicial group	time	crossed module	time
k4 → c2	124	c2 → c2	59
a4a4 → a4	9429	a4 → a4	1705
c4c4 → c4	733	c4 → c4	1777
d8d8 → d8	4131	d8 → d8	756
c4c2c4c2 → c4c2	723	c4c2 → c4c2	532

5. Results

In the following table the 92 groups of size ≤ 30 are ordered by their GAP number. For each group G we list the size of End(G); the size of the set IE(G) of idempotents in End(G), which are candidates for face and degeneracy operators s and b respectively; the size of $\mathcal{S}(G)$, the set of 1-truncated simplicial structures on G; and the number of isomorphism classes of 1-truncated simplicial structures. For each G the first 1-truncated simplicial structure is (id; id, id : $G \to G$) and we omit this from the list. For each of the remaining isomorphism classes we list the names of S, R and, when $\partial \neq 0$, the kernel of the boundary map. Just 50 of these structures have $\partial \neq 0$.

GAP	G	End(G)	IE(G)	$S(G)$	$S/\cong$	Names of S, R and ker ∂
1/1	1	1	1	1	1	
2/1	c2	2	2	2	2	[c2, I]
3/1	c3	3	2	2	2	[c3,I]
4/1	k4	16	8	14	4	[k4,I], [c2,c2], [c2,c2,I]
4/2	c4	4	2	2	2	[c4,I]
5/1	c5	5	2	2	2	[c5,I]
6/1	c6	6	4	4	4	[c6,I][c3,c2], [c2,c3]
6/2	s3	10	5	4	4	[c3,c2]
7/1	c7	7	2	2	2	[c7,I]
8/1	$c2^3$	512	58	226	6	[$c2^3$,I][k4,c2][k4,c2,c2], [c2,k4], [c2k4,I]
8/2	c4c2	32	10	18	6	[c4c2,I], [c4c2], [c4,c2,c2], [c2,c4], [c2,c4,I]
8/3	c8	8	2	2	2	[c8,I]
8/4	d8	36	10	9	3	[c4,c2], [k4,c2]
8/5	q8	28	2	1	1	
9/1	$c3^2$	81	14	38	4	[$c3^2$,I], [c3,c3], [c3,c3,I]
9/2	c9	9	2	2	2	[c9,I]
10/1	c10	10	4	4	4	[c10,I], [c5,c2], [c2,c5]
10/2	d10	26	7	6	2	[c5,c2]
11/1	c11	11	2	2	2	[c11,I]
12/1	c6c2	48	16	28	8	[c6,c2,I][c6,c2], [c6,c2,c3], [k4,c3], [c3,k4], [c2,c6], [c2,c6,I]
12/2	c12	12	4	4	4	[c12,cI], [c4,c3], [c3,c4]
12/3	d12	64	21	12	4	[c6,c2], [c3,k4], [c2,s3]
12/4	q12	20	5	4	2	[c3,c4]
12/5	a4	33	6	5	2	[k4,c3]
13/1	c13	13	2	2	2	[c13,I]
14/1	c14	14	4	4	4	[c14,I], [c7,c2], [c2,c7]
14/2	d14	50	9	8	2	[c7,c2]
15/1	c15	15	4	4	4	[c15,I], [c5,c3], [c3,c5]
16/1	$c2^4$	65536	382	4162	9	[$c2^4$,I], [$c2^3$,c2], [$c2^3$,c2,k4], [k4,k4], [k4,k4,c2], [k4,k4,I], [c2c23], [$c2c2^3$,I]

16/2	c4k4	1024	82	322	12	[c4k4,I], [c4c2,c2], [c4c2,c2,c4], [c4c2,c2,k4], [c4,k4], [c4,k4,c2], [k4,c4], [k4,c4,c2], [c2,c4c2], [c2,c4c2,I], [c2,c4c2,I]
16/3	c8c2	64	10	18	6	[c8c2,I], [c8,c2], [c8,c2,c4], [c2,c8], [c2,c8,I]
16/4	$c4^2$	256	26	98	5	[$c4^2$,I], [c4,c4], [c4,c4,c2], [c4,c4,I],
16/5	c16	16	2	2	2	[c16,I]
16/6	d8c2	1088	82	97	9	[c4c2,c2], [$c2^3$,c2], [c4,k4], [c4,k4,c2], [k4,k4], [k4,k4,c2], [c2,d8], [c2,d8,1]
16/7	q8c2	448	18	17	3	[c2,q8], [c2,q8,1]
16/8	d8y4	224	26	13	2	[c4c2,c2]
16/9	c4c2 ⋉ c2	128	18	25	4	[c4c2,c2], [k4,c4], [k4,c4,c2]
16/10	c4 ⋉ c4	96	10	17	3	[c4,c4], [c4,c4,c2]
16/11	c2 ⋉ c8	48	6	5	2	[c8,c2]
16/12	d16	100	18	9	2	[c8,c2]
16/13	dq16	52	10	5	2	[c8,c2]
16/14	q16	36	2	1	1	
17/1	c17	17	2	2	2	[c17,1I
18/1	c6c3	162	28	76	8	[c6c3,I], [$c3^2$,c2], [c6,c3], [c6,c3,c2], [c3,c6], [c3,c6,I], [c2,$c3^2$]
18/2	c18	18	4	4	4	[c18,I], [c9,c2], [c2,c9]
18/3	d18	82	11	10	2	[c9,c2]
18/4	s3c3	36	12	8	4	[$c3^2$,c2], [c3,c6,c3,s3]
18/5	$c3^2$ ⋊ c2	730	47	118	4	[$c3^2$,c2],[c3,s3],[c3,s3,I]
19/1	c19	19	2	2	2	[c19,I]
20/1	c10c2	80	16	28	8	[c10c2,I], [c10,c2], [c10,c2,c5], [c5,k4], [k4,c5], [c2,c10], [c2,c10,I]
20/2	c20	20	4	4	4	[c20,I], [c5,c4], [c4,c5]
20/3	d20	144	31	18	4	[c10,c2], [c5,k4], [c2,d10]
20/4	q20	52	7	6	2	[c5,c4]
20/5	c4 ⋉ c5	36	7	6	2	[c5,c4]
21/1	c21	21	4	4	4	[c21,I], [c7,c3], [c3,c7]
21/2	c7 ⋊ c3	57	9	8	2	[c7,c3]
22/1	c22	22	4	4	4	[c22,I], [c11,c2], [c2,c11]
22/2	d22	122	13	12	2	[c11,c2]
23/1	c23	23	2	2	2	[c23,I]

24/1	c6k4	1536	116	452	12	[c6k4,I], [c6c2,c2], [c6c2,c2,c6], [$c2^3$,c3], [c6,k4], [c6,k4,c3], [k4,c6], [k4,c6,c2], [c3,$c2^3$], [c2,c6c2], [c2,c6c2,I]
24 2	c12c2	96	20	36	12	[c12c2,I], [c12,c2], [c12,c2,c6], [c4c2,c3], [c6,c4], [c6,c4,c3], [c4,c6], [c4,c6,c2], [c3,c4c2], [c2,c12], [c2,c12,I]
24/3	c24	4	4	4	4	[c24,I], [c8,c3], [c3,c8]
24/4	d8c3	108	20	18	6	[c12,c2], [c6c2,c2], [c4,c6], [k4,c6], [c3,d8]
24/5	q8c3	84	4	2	2	[c3,q8]
24/6	s3k4	1792	157	116	8	[c6c2,c2], [c6,k4], [c6,k4,c3], [k4,s3], [c3,$c2^3$], [c2,d12], [c2,d12,I]
24/7	s3c4	128	27	12	4	[c12,c2], [c4,s3], [c4,c4c2]
24/8	q12c2	160	25	36	6	[c6,c4], [c6,c4,c3], [c3,c4c2], [c2,q12], [c2,12,I]
24/9	c3 ⋉ c8	40	5	4	2	[c3,c8]
24/10	a4c2	72	15	10	4	[$c2^3$,c3], [k4,c6], [c2,a4]
24/11	d8 ⋉ c3	124	23	12	4	[c6c2,c2], [k4,s3], [c3,d8]
24/12	d24	196	33	20	4	[c12,c2], [c4,s3], [c3,d8]
24/13	q24	124	5	4	2	[c3,q8]
24/14	sl(2.3)	33	6	1	1	
24/15	s4	58	12	5	2	[k4,s3]
25/1	$c5^2$	625	32	152	4	[$c5^2$,I], [c5,c5], [c5,c5,I]
25/2	c25	25	2	2	2	[c25,I]
26/1	c26	26	4	4	4	[c26,I], [c13,c2], [c2,c13]
26/2	d26	170	15	14	2	[c13,c2]
27/1	$c3^3$	19683	236	2108	6	[$c3^3$,I], [$c3^2$,c3], [$c3^2$,c3,c3], [c3,$c3^2$], [c3,$c3^2$,I]
27/2	c9c3	243	20	56	6	[c9c3,I], [c9,c3], [c9,c3,c3], [c3,c9], [c3,c9,I]
27/3	c27	27	2	2	2	[c27,I]
27/4	$c3^2$ ⋉ c3	729	38	37	2	[$c3^2$,c3]
27/5	c9 ⋉ c3	135	11	10	2	[c9,c3]
28/1	c14,c2	112	16	28	8	[c14c2,I], [c14,c2], [c14,c2,c7], [c7,k4], [k4,c7], [c2,c14], [c2,c14,I]
28/2	c28	28	4	4	4	[c28,I], [c7,c4], [c4,c7]
28/3	d28	256	41	24	4	[c14,c2], [c7,k4], [c2,d14]
28/4	q28	100	9	8	2	[c7,c4]

29/1	c29	29	2	2	2	[c29,I]
30/1	c30	30	8	8	8	[c30,I], [c15,c2], [c10,c3], [c6,c5], [c5,c6], [c3,c10], [c2,c15]
30/2	d10c3	78	14	12	4	[c15,c2], [c5,c6], [c3,d10]
30/3	d6c5	50	10	8	4	[c5,c2], [c5,s3], [c3,c10]
30/4	d30	226	25	24	4	[c15,c2], [c5,s3], [c3,d10]

REFERENCES

1. Alp M., SIMPCATn-Catn-groups and Simplicial groups in GAP, version 1.1 Manual for the SIMPCAT-n share package, 1-20 (2001) (Submitted).
2. Alp M. and Wensley C. D., XMOD-Crossed modules and Cat1-groups in GAP, Version 1.3, Manual for the XMOD share package, 1-80 (1996).
3. Alp M., *Int. J. Algebra and Computation*, **12**(5) (2002), 645-58.
4. Alp M. and Wensley C. D., *Int. J. Algebra and Computation*, **10**(4) (2000), 407-24.
5. Loday J. L., *J. App. Algebra*, 24 (1982), 179-202.
6. Schönert M. *et al.*, GAP: Groups, Algorithms and Programming, Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany, third edition, 1993.
7. Whitehead J. H. C., *Bull. Amer. Math. Soc.*, 55 (1949), 453-96.