2001 VOL. XIV(XXII)

**MURAT ALP**

# SECTIONS IN GAP

**Abstract**

*In this paper we describe a share package XMOD [13] of functions for computing with finite, permutation crossed modules, their morphisms and derivations; cat$^1$-groups, their morphisms and their sections, written using the GAP [12] group theory programming language. We also give the implementation method of sections to the GAP.*

**1991 A. M. S. C.:** **13D99, 16A99, 17B99, 17D99, 18D35.**
**Keywords:** **Crossed modules, derivation whitehead multiplication, cat$^1$-groups, sections.**

## 1. Introduction.

A starting point for this paper was to consider the possibility of implementing functions for doing calculations with crossed modules, derivations, actor crossed modules, cat$^1$-groups, sections, induced crossed modules and induced cat$^1$-groups in GAP [12].

Our aim in this paper is to describe a share package XMOD [13] the GAP group theory language which enables computations with the equivalent notions of finite, permutation *crossed modules* and *cat$^1$-groups*.

The term crossed module was introduced by J.H.C.Whitehead in [14]. Most references of crossed module state the axioms of a crossed module using left actions, but we shall use right actions since this is the convention used by most computational group packages.

In [8] Loday reformulated the notion of a crossed module as a 1-cat group and showed that the category XMod is equivalent to the category Cat1 of cat-groups. Loday also generalized the notion of cat$^1$-group to that of cat $n$-group, for all $n \geq 1$ (although he used the term $n$-cat-group). Crossed modules and their higher analogues were considered by Ellis in [6].

In section 2 we recall the basic properties of crossed modules and their derivations and of cat$^1$-groups and their sections. We also gave the implementation method and implementation algorithms in section 3.

## 2. Crossed Modules and Cat$^1$-Groups.

In this section we recall the descriptions of three equivalent categories: XMod, the category of crossed modules and their morphisms; Cat 1, the category of cat$^1$-groups and their morphisms; and **GpGpd**, the subcategory of groups in the category **Gpd** of groupoids. We also describe functors between these categories which exhibit the equivalences.

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

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

$$XMod\ 2 : s^{\partial s'} = s'^{-1} s s'.$$

The kernel of $\partial$ is abelian.

The standard examples of crossed modules are:

1. Any homomorphism $\partial : S \to R$ of abelian groups with $R$ acting trivially on $S$ may be regarded as a crossed module.
2. A conjugation crossed module is an inclusion of a normal subgroup $S \leq R$,, where $R$ acts on $S$ by conjugation.
3. A central extension crossed module has as boundary 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 $Aut(S)$ 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 $\partial$ 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 $\partial_1 \times \partial_2$, with $R_1, R_2$ acting trivially on $S_2, S_1$ respectively.
7. An important motivating topological example of crossed module due to Whitehead [15] is the boundary $\partial : \pi_2(X, A, x) \to \pi_1(A, x)$ from the second relative homotopy group of a based pair $(X, A, x)$ of topological spaces, with the usual action of the fundamental group $\pi_1(A, x)$.

A morphism between two crossed modules $\chi_1$ and $\chi_2$ is a pair $(\sigma, \rho)$ where $\sigma : S_1 \to S_2$ and $\rho : R_1 \to R_2$ are homomorphisms satisfying

$$\partial_2 \sigma = \rho \partial_1, \ \sigma(s^r) = (\sigma s)^{\rho r}.$$

When $\chi_2 = \chi_1$ and $\sigma, \rho$ are automorphisms then $(\sigma, \rho)$ is an automorphism of $\chi_1$. The group of automorphisms is denoted by $Aut(\chi_1)$.

The Whitehead monoid $Der(\chi)$ of $\chi$ was defined in [15] to be the monoid of all derivations from $R$ to $S$, that is the set of all maps $R \to S$, with composition o, satisfying

$$Der1: \quad \chi(qr) = (\chi q)^r (\chi r)$$

$$Der2: \quad (\chi_1 \circ \chi_2)(r) = (\chi_1 r)(\chi_2 r)(\chi_1 \partial \chi_2 r).$$

Inevitable elements in the monoid are called regular. The Whitehead group $W(\chi)$ is the group of $Der(\chi)$. The actor of $\chi$ is a crossed module $(\Delta : W(\chi) \to Aut(\chi))$ which was shown by Lue and Norrie, in [9], [11] and [10], to be the automorhism object of $\chi$ in the category XMod.

The standard of Whitehead groups [7] are:

1. If $S$ is a $R$ - module, then the trivial homomorphism $S \to R$ is a crossed module and $Der(\chi) = W(\chi)$ is the usual abelian group of derivations.
2. Together with the conjugation action of a group $R$ on itself, the identity map $\chi = (id : R \to R)$ is a crossed module. An automorphism $\alpha$ of $R$ determines its displacement derivation $\delta_\alpha \in W(\chi)$ given by $\delta_\alpha(r) = \alpha(r) r^{-1}$, and the correspondence $\alpha \mapsto \delta_\alpha$ is an isomorphism $\delta : AutR \to W(\chi)$.
3. Generalizing (ii) we have the inclusion $S \to R$ of a normal subgroup $S$ of a group $R$, with $R$ acting by conjugation. Then $W(\chi)$ is isomorphic to the subgroup of $AutR$ consisting of all those $\alpha$ whose displacement derivations take values in $S$,

$$W(\chi) \cong \{\alpha \in AutR \mid for\ all\ r \in R, \alpha(r)r^{-1} \in S\}$$

In particular, if $S$ is a characteristic subgroup of $R$, then $W(\chi)$ is the kernel of the canonical map from $AutR$ to $Aut(R/S)$.

4. A crossed module $\chi = (\partial : S \to R)$ with surjective boundary map amounts to a central extension of $\ker\partial$ by $R$; so let $E$ be a group and $K$ a central subgroup of $E$. Let $Aut_k E$ be the subgroup of $AutE$ consisting of those automorphism of $E$ that act trivially on $K$. The natural map $N = (v : E \to E/K)$ is a crossed module and $W(N)$ is isomorphic to $Aut_k E$.

In [8] Loday reformulated the notion of a crossed module as a cat$^1$-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 cat$^1$-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:

$$Cat1:\ te = he = id_R,$$
$$Cat2:\ [\ker t, \ker h] = \{1_G\}.$$

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 cat$^1$-groups is a pair $(\gamma, \rho)$ where $\gamma : G_1 \to G_2$ and $\rho : R_1 \to R_2$ are homomorphisms satisfying

$$h_2\gamma = \rho h_1,\quad t_2\gamma = \rho t_1,\quad e_2\rho = \gamma e_1. \tag{1}$$

The construction for cat$^1$-groups equivalent to the derivation of a crossed module is the section. The monoid of sections of $C$ is the set of group homomorphisms $\xi : R \to G$, with composition o, satisfying:

$$Sect1:\ t\xi = id_R,$$
$$Sect2:\ (\xi_1 \circ \xi_2)(r) = (\xi_2 r)(eh\xi_2 r)^{-1}(\xi_1 h\xi_2 r).$$

The embedding $e$ is the identity for this composition, and $h(\xi_1 \circ \xi_2) = (h\xi_1)(h\xi_2)$. A section is regular when $h\xi$ is an automorphism and, of course, the group of regular sections is isomorphic to the Whitehead group.

The crossed module $\chi$ associated to $C$ has $S = \ker t$ and $\partial = h|_s$. The cat$^l$-group associated to $\chi$ has $G = R \times S$, using the action from $\chi$, and

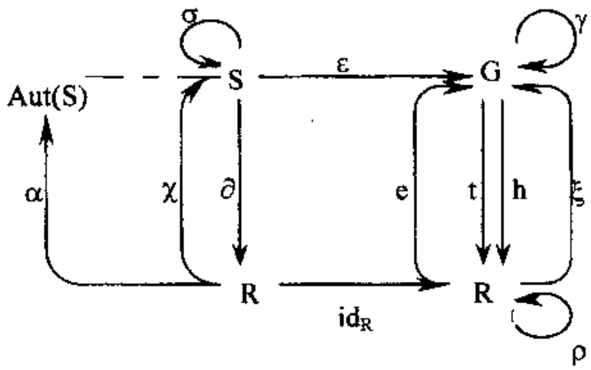
$$t(r,s) = r,\ h(r,s) = r(\partial s),\ er = (r,1).$$

We denote by $\in$ the inclusion of $S$ in $G$.

The equation $\xi r = (er)(\chi r)$ defines a section $\xi$ of $C$, given a derivation $\chi$ of $\chi$, and conversely. Each $\chi$ or $\xi$ determines endomorphisms of $R, S, G, \chi$ and $C$ namely

$$\rho : R \to R,\quad r \mapsto r(\partial\chi r) = h\xi r,$$
$$\sigma : S \to S,\quad s \mapsto s(\chi\partial s),$$
$$\gamma : G \to G,\ g \mapsto (eh\xi tg)(\xi tg^{-1})g(ehg^{-1})(\xi hg),$$
$$(\sigma, \rho) :\ \chi \to \chi,$$
$$(\gamma, \rho) :\ C \to C.$$

The accompanying diagram shows the relationship between the various groups and homomorphisms.




**Figure**

### 3. GAP Implementation.

In order to represent crossed modules, their derivations and actors, cat[1]-groups and their sections within GAP we utilise the record structure in GAP3. The standard method in GAP is to represent a new algebraic structure as a record with a set of fields. All records with the same structure are allocated an operations record, namely a record whose fields are functions which operate on the structure. Thus the operations record XModOps contains functions such as DirectProduct, IsSimplyConnected and Actor. Similarly the operations record XModMorphismOps contains functions such as Kernel and CompositeMorphism. We descride algorithms for constructing crossed modules and their derivations in [1, 2, 3, 4, 5]. We also describe algorithms for constructing sections of cat[1]-groups in section 4.

### 4. Algorithms for Sections.

Sections are group homomorphism which satisfy the section conditions Sect1 and Sect2. In the implementation a section is stored as a GroupHomomorphismByImages. However, sections are provided with a modified set of operations, Cat1SectionByImagesOps, which includes a special. Print function to display the section. There are two more functions, SectionDerivation and DerivationSection, which convert derivations to sections and vice-versa.

#### 4.1. Record structures for Sections.

A section $\xi : R \to G$ is stored as a record with fields:

| | |
|---|---|
| **xi.source** | the range group $R$ of $C$ |
| **xi.range** | the source group $G$ of $C$ |
| **xi.generators** | a fixed generating set for $R$, |
| **xi.genimages** | the chosen images of the generators, |
| **xi.cat1** | the cat[1]-group $C$ |
| **xi.operations** | special set of operations Cat1SectionByImagesOps, |
| **xi.isSection** | a boolean flag normally true. |

There are two functions to calculate sections, RegularSections and AllSections. Both create or modify a record Sec = C.sections with fields:

| | |
|---|---|
| **Sec.areSections,** | a boolean flag, normally true, |
| **Sec.isReg,** | true when only the regular sections are known, |
| **Sec.isAll,** | true when all the sections have been found, |

| | |
|---|---|
| **Sec.generators,** | a copy of R.generators, |
| **Sec.genimageList,** | a list of genimages lists for the sections, |
| **Sec.regular,** | the number of regular sections (if known), |
| **Sec.cat$^1$,** | the cat$^1$-group $C$ , |
| **Sec.operations,** | a special set of operations Cat1SectionsOps. |

**4.2. Algorithm for Cat1SectionByImages.**

The function Cat1SectionByImages is called as:
gap>Cat1SectionByImages (C, im);
The input parameters are a cat$^1$-group and a list of images. As output, the function returns a record as described in section 4.1.

| | |
|---|---|
| Step 1 | Check that the given arguments are of the correct form. |
| Step 2 | Construct a map $\xi : R \to G$ using GroupHomomorphismByImages (R, G, genR, im); |
| Step 3 | Set up the record fields as described in section 4.1. |
| Step 4 | Call IsSection(xi); to verify axioms SECT1 and SECT2. |

**5. RegularSections and AllSections.**

It is easier to test that a prospective $\xi$ is a homomorphism than that a map $\chi$ satisfies axiom Der 1. However, since $\xi r = (er)(\chi r)$, a section $\xi$ is determined by a choice of $\chi r_i$ for each $r_i$ in a generating set $\{r_1, r_2, ...\}$ for R.

Since $r^{-1}(\rho r) = \partial \chi r$ it follows that $\chi r \in \partial^{-1}(r^{-1}(\rho r))$. In order to find the regular sections, we use the standard GAP function AutomorphismGroup(R) to obtain $Aut(R)$. Then, for each $\rho \in Aut(R)$, we make a list of preimages

$$\left[\partial^{-1}\left(r_1^{-1}(\rho r_1)\right), \partial^{-1}\left(r_2^{-1}(\rho r_2)\right), ...\right].$$

A backtrack procedure is then used to select $\chi r_1$, $\chi r_2$,... from these preim-age lists, with each selection being tested to see whether it provides a partial homomorphism $R \to G$.
A similar strategy is used to find all the sections, replacing $Aut(R)$ by the endomorphism monoid $End(R)$. Since no standard GAP function yet exists for computing $End(R)$, we have added a function EndomorphismClasses(R). An endomorphism of R is determined by

- a normal subgroup $N$ of $R$, a permutation representation of the quotient $\theta : R/N \to Q$, giving a projection $\theta \circ v : R \to Q$, where $v : R \to R/N$ is the natural homomorphism;
- an automorphism $\alpha$ of Q;
- a subgroup $H'$ in a conjugacy class [H] of subgroups of $R$ isomorphic to $Q$ having representative $H$, an isomorphism $\phi : Q \cong H$, and a conjugating element $r \in R$ such that $H^r = H'$.

Endomorphisms are placed in the same class if they have the same choice of $N$ and *[H]*, so number of endomorphisms is

$$|End(R)| = \sum_{classes} |Aut(Q)||[H]|.$$

The function returns a record E =R.endomorphismClasses.classes with fields

| | |
|---|---|
| **E.quotient,** | the group $Q \cong R/N$, |
| **E.autoGroup,** | the automorphism group of $Q$, |
| **E.Isomorphism,** | the isomorphism $\phi \circ \theta \circ v$, |
| **E.representative,** | the subgroup H, |

**E.conj,** the list of conjugating elements $r'$.

Functions RegularSections and ALLSections are called as:
gap>RegularSections (C [, method]);
gap>AllSections (C, [, method]);
where method is one of «endo» or «xmod». The default method is «endo» uses the method described in the previous section. When «xmod» is specified the following procedure is used.

**Step 1** Call X: = XModCat1(C);

**Step 2** Call D: = RegularDerivation(X);

**Step 3** For each image im in D.genimageList call SectionDerivation(C, im); to construct the corresponding section.

**6. Cat1SectionByImages.**

Cat1SectionByImages(C,im)

This function takes a list of images in G = C.source for the generations of R = C.range and constructs a homomorphism $\xi: R \to G$ which is then tested to see whether the axioms of a section are satisfied.

gap>SC;
Cat1-group [c3^2|xc2==> s3]
gap>im|xi : = [(1,2,3), (1,2) (4,6)];
gap>xi : = Cat1SectionByImages (SC, imxi);
Cat1SectionByImages (s3, c3^2|xc2, [(4,5,6), (2,3) (5,6)], [(1,2,3), (1,2) (4,6)])

**7. IsSection.**

IsSection (C, im)
IsSection (xi)
This function may be called in two ways, and tests that the section given by the images of its generators is well-defined.
gap> im0 : = [(1,2,3), (2,3) (4,5)];
gap> IsSection (SC, im0);
false

**8. RegularSections.**

RegularSections (C[, «endo» or «xmod»]).

By default, this function computes the set of idempotent automorphisms from $R \to R$ and takes these as possible choices for $h\xi$. A backtrack procedure then calculates possible images for such a section. The result is stored in a sections record C.sections with fields similar to those of a serivations record. The alternative strategy, for which «xmod» option should be specified is to calculate the regular derivations of the associated crossed module first, and convert the resulting derivations to sections.
gap> Unbind (XSC.derivations);
gap> regSC : RegularSections(SC);
RegularSections record for cat1-group [c3^2|Xc2==>s3],
: 6 regular sections, others not found.

**9. AllSections.**

AllSections (C [, «endo» or «xmod»])

By default, this function computes the set of idempotent endomorphisms from $R \to R$ and these takes as possible choices for the composite homomorphism $h\xi$. A backtrack procedure then calculates possible images for such a section. This function calculates all the sections of *C* and overwrites any existing subfields of *C.sections.*

gap> allSC : = AllSections (SC);
AllSections record for cat1-group [c3^2|Xc2==> s3],
: 6 regular sections, 3 irregular ones found.
gap>RecFields (allSC);
[«areSections», «isReg», «isAll», «regular», «genimageList», «genirators», «cat1», «operations»]
gap> PrintList (allSC. genimageList);
[(4, 5, 6), (2, 3) (5, 6)]
[(4, 5, 6), (1, 3) (4, 5)]
[(4, 5, 6), (1, 2) (4, 6)]
[(1, 3, 2) (4, 6, 5), (2, 3) (5, 6)]
[(1, 3, 2) (4, 6, 5), (1, 3) (4, 5)]
[(1, 3, 2) (4, 6, 5), (1, 2) (4, 6)]
[(1, 2, 3), (2, 3) (5, 6)]
[(1, 2, 3), (1, 2) (4, 6)]
[(1, 2, 3), (1, 3) (4, 5)]
gap> allXSC : = AllDerivations (XSC, «cat1»);
AllDerivations record for crossed module [c3->s3],
: 6 regular derivations, 3 irregular ones found.

**10. AreSections.**

AreSections (S)

This function checks that the record S has the correct fields for a sections record (regular or all).
gap> AreSections (allSC);
true

**11. SectionDerivation.**

SectionDerivation (D,i)

This function converts a derivation of X to a section of the associated cat1-group C. This function is inverse to DerivationSection. In the following examples we note that allXSC has been obtained using allSC, so the derivations and sections correspond in the same order.
gap> chi8 : = XModDerivationByImages (XSC, allXSC.genimageList[8]);
XmodDerivationByImages(s3, c3, [(4, 5, 6), (2,3) (5,6)],
[(1, 2, 3) (4, 6, 5), (1, 2, 3) (4, 6, 5)])
gap> xi8 : = SectionDerivation(chi8);
GroupHomomorphismByImages (s3, c3^2/Xc2,
[(4, 5, 6), (2,3) (5,6)], [(1, 2, 3), (1, 2) (4, 6)])

**12. DerivationSection.**

DerivationSection (C, xi)

This function converts a section of C to a derivation of the associated crossed module X. This function is inverse to SectionDerivation.
gap>xi4 : = Cat1SectionByImages (SC, allSC.genimageLict[4]);
Cat1SectionByImages (s3, c3^2/Xc2, [(4, 5, 6), (2, 3) (5, 6)],
[(1, 3, 2) (4, 6, 5), (2, 3) (5, 6)])
gap>chi4 : = DerivationSection (xi4);
XModDerivationByImages (s3, c3, [(4, 5, 6), (2, 3) (5, 6)],
[(1, 3, 2) (4, 5, 6), ( )])

**13. CompositeSection.**

CompositeSection (xi, xj)

This function applies the Whitehead composition to two sections and returns the composite.

gap>xi48 : = CompositeSection (xi4, xi8);

Cat1SectionByImages (s3, c3^2|Xc2, [(4, 5, 6), (2,3) (5, 6)],

[(1, 2, 3), (1,3) (4,5)])

gap true>SectionDerivation(chi48) = xi48;

**14. SourceEndomorphismSection.**

SourceEndomorphismSection (xi)

Each section $\xi$ determines an endomorphism $\gamma$ of $G$ such that

$$\gamma g = (eh\xi\, tg)(\xi\, tg^{-1})g(ehg^{-1})(\xi\, hg).$$

gap>gamma4 : = SourceEndomorphismDerivation (xi4);

GroupHomomorphismByImages (c3^2|Xc2, c3^2|Xc2,

[(1, 2, 3), (4, 5, 6), (2, 3) (5, 6)], [(1, 3, 2), (4, 6, 5), (2, 3) (5, 6)])

**15. RangeEndomorphismSection.**

RangeEndomorphismSection (xi)

Each derivation $\xi$ determines an endomorphism $\rho$ of $R$ such that $\rho r = h\xi r$.

gap>rho4 : = RangeEndomorphismDerivation (XSC, 4);

GroupHomomorphismByImages (s3, s3, [(4, 5, 6), (2, 3) (5, 6)],

[(4, 6, 5), (2, 3) (5, 6)])

**16. Cat1EndomorphismSection.**

Cat1EndomorphismSection (xi)

The endomorphisms gamma4, rho4 together determine a pair which may be used to construct an endomorphism of $C$. When the derivation is regular, the resulting morphism is an automorphism, and this construction determines a homomorphism from the Whitehead group to the automorphism group of $C$.

gap>psi4 : = Cat1EndomorphismSection (xi4);

Morphism of cat1-groups <[c3^2|Xc2 ==> s3]→[c3^2|Xc2==>s3]>

**References**


[1]. Alp, M., Algoritms for two equivalent categoties:crossed modules and Cat$^1$-groups, *International Journal of natural Sciences and Engineering..*

[2]. Alp, M., Special Cases of derivation groups, *Documeta Mathematica,* (Incelemede)

[3]. Alp, M., Special Cases of Cat1-groups, *Algebras, Groups and Geometries.* (Vol 15) 1998.

[4]. Alp, M., Enumeration of Cat1-groups of degree 41-47, *Anadolu Universitesi Fen-Fakultesi yayinlari.* (Saqi 3) 1997 79-90.

[5]. Alp, M., Wensley, C.D., Enumeration of Cat1-groups of low order, *U.W.B. preprint,* 96.05, 15pp.

[6]. G. Ellis, *Crossed modules and their higher dimensional analogues,* PHD thesis, University of Wales, Bangor, 1984.

[7]. Gilbert, N,D., Derivations, automorphisms and crossed modules, *Comm. in Algebra* 18 (1990) 2703-2734.

[8]. Loday, J.L., Spaces with finitely many non-trivial homotopy groups, *J.App.Algebra,* 24 (1982) 179-202.

[9]. Lue, A.S.T., The Ganea map for nilpotent groups, *J.London Math. Soc., (2)* 14 (1976) 309-312.

[10]. Norrie, K.J., Crossed module and anologues of group theorems, *Ph.D thesis, Kings College, University of London,* (1987).
[11]. Norrie, K.J., Actions and automorphisms of crossed modules, *Bull. Soc. Math. France,* 118 (1990) 129-146.
[12]. Schönert, M. Et al, GAP: Groups, Algorithms, and Programming, Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany, third edition, 1993.
[13]. Wensley, C.D., and Alp, M., XMOD, Crossed modules and cat1-groups in GAP, version 1.3 *Manumal for the XMOD share package,* (1996) 1-80.
[14]. Whitehead, J.H.C., on operators in relative homotopy group, Ann. Of Math., 49 (1948) 610-640.
[15]. Whitehead, J.H.C., Combinatorial homotopy II, *Bull. A.M.S., 55* (1949) 453-496.

**Alp Murat**
Dumlupinar Üniversitesi, Fen-Edebiyat Fakültesi, Matematik Bölümü
Kütahya, Turkey.

Received February 8, 2000; Revised December 15, 2000.
Translated by author.