Interoperating between Computer Algebra systems: computing homology of groups with Kenzo and GAP

Ana Romero
Dpto. de Matemáticas y Computación
Universidad de La Rioja
Edif. Vives, c/Luis de Ulloa s/n
26004 Logroño, Spain
ana.romero@unirioja.es

Graham Ellis
Department of Mathematics
National University of Ireland, Galway
University Road, Galway, Ireland
graham.ellis@nuigalway.ie

Julio Rubio
Dpto. de Matemáticas y Computación
Universidad de La Rioja
Edif. Vives, c/Luis de Ulloa s/n
26004 Logroño, Spain
julio.rubio@unirioja.es

ABSTRACT

In this paper we report on an experience communicating two computer algebra systems, namely GAP (and more concretely, its HAP package to compute in Homological Algebra) and Kenzo (to compute in Algebraic Topology). Both systems cooperate through an OpenMath link in computing homology of groups. In addition, once the output from HAP has been integrated in Kenzo, it can be used to compute more complicated algebraic invariants, as homology groups of some 2-types.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Communications Applications; G.4 [**Mathematics of Computing**]: Mathematical software

General Terms

Homology of groups, Interoperability, OpenMath, GAP, Kenzo

1. INTRODUCTION

GAP [GAP] is a Computer Algebra system, well-known for its contributions, in particular, in the area of Computational Group Theory. Kenzo [DRSS99] is a more specific system, developed by Sergeraert to implement his ideas on Constructive Algebraic Topology [RS02]. One topic where Group Theory and Algebraic Topology meet is the definition and calculation of (co)homology of groups. Any textbook on this subject (Brown's book [Bro82] being a good example) stresses this fact: the easier way to define the homology of a group G is identifying it with the homology of a canonical topological space associated with G, usually denoted by $K(G, 1)$. This is also quite common that Algebraic Topology appears only in this first, definitional, step. Textbooks continue often with a more algebraic flavour, usually based on the *resolution* notion.

One of the reasons for this distance from Algebraic Topology could be that, traditionally, this discipline has been considered far from the explicit computations needed (and looked for) when dealing with homology of groups. Nevertheless, the Kenzo system changed dramatically this view in last years. Now, one can deal on a computer with complex simplicial spaces, applying high level constructors (as fibrations, loop spaces, classifying spaces, and so on), and finally ask for some of their homology groups.

Furthermore, Ellis developed some years ago a GAP package, called HAP [Ell] (for Homological Algebra Programming), devoted to the computation of (co)homology of groups. In particular, it includes specific algorithms to compute resolutions for a wide variety of groups. Thus, a natural question comes to the mind: why HAP and Kenzo could not cooperate in computations where homology of groups is needed? This paper gives a positive answer to this question.

The two main contributions of this paper are: (1) from the algorithmic point of view, a program computing from a small resolution of a group G, the *effective homology* of the space $K(G, 1)$; and (2) from a more technical perspective, an OpenMath description of groups and resolutions, which can be *exported* from GAP [SC], and then *imported* by Kenzo.

The important conclusion of both contributions is that, once the effective homology of $K(G, 1)$ is built as a Kenzo object, it can be used for further operations and computations (for instance, it can appear as fiber or base space of fibrations, it can be combined with other spaces and so on). Two examples of such applications are presented in this paper.

The paper is organized as follows. Next section, of preliminaries, briefly introduces homology of groups, the effective homology technique (in which Kenzo is based on) and GAP, HAP and Kenzo. The main algorithm (constructing the effective homology of $K(G, 1)$ from a small resolution of the group G) is described in Section 3. Then, Section 4 deals with OpenMath issues. Section 5 is devoted to applications and examples. The paper ends with a section of conclusions and the bibliography.

2. PRELIMINARIES

2.1 Some fundamental notions about homology of groups

The following definitions and important results about homology of groups can be found in [Mac63] and [Bro82].

Definition 1. Given a ring R, a *chain complex* of R-modules is a pair of sequences $C_* = (C_n, d_n)_{n\in\mathbb{Z}}$ where, for each degree $n \in \mathbb{Z}$, C_n is an R-module and $d_n : C_n \to C_{n-1}$ (*the differential map*) is an R-module morphism such that $d_{n-1} \circ d_n = 0$ for all n.

A particular case is obtained when $R = \mathbb{Z}$, the integer ring. Then each C_n is an Abelian group and the differential maps $d_n : C_n \to C_{n-1}$ are Abelian group morphisms.

Definition 2. Let $C_* = (C_n, d_n)_{n\in\mathbb{Z}}$ be a chain complex of R-modules, being R a general ring. For each degree $n \in \mathbb{Z}$, the *n-homology module* of C_* is defined as the quotient module $H_n(C_*) = \operatorname{Ker} d_n / \operatorname{Im} d_{n+1}$.

Definition 3. A chain complex $C_* = (C_n, d_n)_{n\in\mathbb{Z}}$ is *acyclic* if $H_n(C_*) = 0$ for all n.

Definition 4. Let G be a group and $\mathbb{Z}G$ the free $\mathbb{Z}$-module generated by the elements of G. The multiplication in G extends uniquely to a $\mathbb{Z}$-bilinear product $\mathbb{Z}G \times \mathbb{Z}G \to \mathbb{Z}G$ which makes $\mathbb{Z}G$ a ring. This is called the *integral group ring* of G.

Definition 5. A *resolution* F_* for a group G is an acyclic chain complex of $\mathbb{Z}G$-modules

$$\cdots \longrightarrow F_2 \xrightarrow{d_2} F_1 \xrightarrow{d_1} F_0 \xrightarrow{\varepsilon} F_{-1} = \mathbb{Z} \longrightarrow 0$$

where $F_{-1} = \mathbb{Z}$ is considered a $\mathbb{Z}G$-module with the trivial action and $F_i = 0$ for $i < -1$. The map $\varepsilon : F_0 \to F_{-1} = \mathbb{Z}$ is called the *augmentation*. If F_i is free for each $i \geq 0$, then F_* is said to be a *free resolution*.

Given a free resolution F_*, one can consider the chain complex $C_* = (C_n, d_{C_n})_{n\in\mathbb{N}}$ of *Abelian groups* defined by

$$C_n = (\mathbb{Z} \otimes_{\mathbb{Z}G} F_*)_n, \quad n \geq 0$$

(where $\mathbb{Z} \equiv C_*(\mathbb{Z}, 0)$ is the chain complex with only one non-null $\mathbb{Z}G$-module in dimension 0) with differential maps $d_{C_n} : C_n \to C_{n-1}$ induced by $d_n : F_n \to F_{n-1}$.

Let us emphasize the difference between both chain complexes F_* and $C_* = \mathbb{Z} \otimes_{\mathbb{Z}G} F_*$. The elements of F_n $(n \geq 0)$, can be seen as *words* $\sum \lambda_i(g_i, z_i)$ where $\lambda_i \in \mathbb{Z}$, $g_i \in G$ and z_i is a generator of F_n (which is a *free* $\mathbb{Z}G$-module). On the other hand, the associated chain complex $C_* = \mathbb{Z} \otimes_{\mathbb{Z}G} F_*$ of Abelian groups has elements in degree n: $\sum \lambda_i z_i$ with $\lambda_i \in \mathbb{Z}$ and z_i is now a generator of the free $\mathbb{Z}$-module C_n.

Although the chain complex of $\mathbb{Z}G$-modules F_* is acyclic, $C_* = \mathbb{Z} \otimes_{\mathbb{Z}G} F_*$ is in general not exact, thus its homology groups are not null. A very important result of homology of groups claims that these homology groups are independent of the chosen resolution for G, which makes it possible to define the *homology of a group*.

THEOREM 1. *[Bro82] Let G be a group and F_*, F'_* two free resolutions of G. Then*

$$H_n(\mathbb{Z} \otimes_{\mathbb{Z}G} F_*) \cong H_n(\mathbb{Z} \otimes_{\mathbb{Z}G} F'_*) \quad \textit{for all } n \in \mathbb{N}$$

The condition of F and F' being free can be in fact *relaxed* and the result is also true for projective resolutions.

Definition 6. Given a group G, the *homology groups* $H_n(G)$ are defined as

$$H_n(G) = H_n(\mathbb{Z} \otimes_{\mathbb{Z}G} F_*), \quad n \in \mathbb{N}$$

where F_* is a projective resolution for G.

Given a group G, how can we determine a resolution F_*? One can always consider the *bar resolution* $B_* = \operatorname{Bar}_*(G)$ [Mac63], whose associated chain complex $\mathbb{Z} \otimes_{\mathbb{Z}G} B_*$ can be seen as the Eilenberg-MacLane space $K(G, 1)$ (see [Bro82] for details). The homology groups of $K(G, 1)$ are those of the group G and this space has a big structural richness, but it has a serious drawback: its size. If $n > 1$, then $K(G, 1)_n = G^n$; in particular, if $G = \mathbb{Z}$, the space $K(G, 1)$ is infinite. This fact seems an important obstacle for using $K(G, 1)$ in order to compute the homology groups of G; it could be convenient to try to construct *smaller* resolutions.

For some particular cases, small (minimal) resolutions can be directly constructed. For instance, let G be the cyclic group of order m with generator t. The resolution F_* for G

$$\cdots \xrightarrow{t-1} \mathbb{Z}G \xrightarrow{N} \mathbb{Z}G \xrightarrow{t-1} \mathbb{Z}G \longrightarrow \mathbb{Z} \longrightarrow 0$$

where N denotes the *norm element* $1 + t + \cdots + t^{m-1}$ of $\mathbb{Z}G$, produces the chain complex of Abelian groups

$$\cdots \xrightarrow{0} \mathbb{Z} \xrightarrow{m} \mathbb{Z} \xrightarrow{0} \mathbb{Z} \longrightarrow 0$$

and therefore

$$H_i(G) = \begin{cases} \mathbb{Z} & \text{if } i = 0 \\ \mathbb{Z}/m\mathbb{Z} & \text{if } i \text{ is odd} \\ 0 & \text{if } i \text{ is even}, i > 0 \end{cases}$$

In general it is not so easy to obtain a resolution for a group G. As we will see in Section 2.3, the GAP package HAP has been designed trying to determine resolutions for a wide variety of groups.

2.2 Effective homology

In the following lines we present the general ideas of the effective homology method, deeply explained in [RS02] and [RS06].

Definition 7. A *reduction* ρ between two chain complexes $C_* = (C_n, d_{C_n})_{n\in\mathbb{N}}$ and $D_* = (D_n, d_{D_n})_{n\in\mathbb{N}}$ (which is denoted $\rho : C_* \Rrightarrow D_*$) is a triple (f, g, h) where: (a) the components f and g are chain complex morphisms $f : C_* \to D_*$

and $g : D_* \to C_*$; (b) the component h is a homotopy operator $h : C_* \to C_{*+1}$ (a graded group morphism of degree +1); (c) the following relations are satisfied: (1) $fg = \mathrm{Id}_D$; (2) $d_C h + h d_C = \mathrm{Id}_C - gf$; (3) $fh = 0$; (4) $hg = 0$; (5) $hh = 0$.

These relations express that C_* is the direct sum of D_* and an acyclic complex. This decomposition is simply $C_* = \mathrm{Ker}\, f \oplus \mathrm{Im}\, g$, with $\mathrm{Im}\, g \cong D_*$ and $H_*(\mathrm{Ker}\, f) = 0$. In particular, this implies that the graded homology groups $H_*(C_*)$ and $H_*(D_*)$ are canonically isomorphic.

A reduction is in fact a particular case of chain equivalence in the classical sense (see [Mac63]), where the homotopy operator on the chain complex D_* is the null map.

Definition 8. A *(strong chain) equivalence* ε between two chain complexes C_* and D_*, denoted by $\varepsilon : C_* \Lleftarrow\!\!\Rrightarrow D_*$, is a triple (B_*, ρ_1, ρ_2) where B_* is a chain complex, and ρ_1 and ρ_2 are reductions $\rho_1 : B_* \Rrightarrow C_*$ and $\rho_2 : B_* \Rrightarrow D_*$.

REMARK 1. *We must use the notion of* effective *chain complex: it is essentially a free chain complex C_* where each group C_n is finitely generated, and a provided algorithm returns a (distinguished) $\mathbb{Z}$-basis in each degree n; in particular, its homology groups are elementarily computable (for details, see [RS02]).*

Definition 9. An *object with effective homology* X is a quadruple $(X, C_*(X), HC_*, \varepsilon)$ where $C_*(X)$ is a chain complex canonically associated with X (which allows us to study the homological nature of X), HC_* is an effective chain complex, and ε is an equivalence $\varepsilon : C_*(X) \Lleftarrow\!\!\Rrightarrow HC_*$.

It is important to understand that in general the HC_* component of an object with effective homology is *not* made of the homology groups of X; this component HC_* is a free $\mathbb{Z}$-chain complex of finite type, in general with a non-null differential, whose homology groups $H_*(HC_*)$ can be determined by means of an elementary algorithm. From the equivalence ε one can deduce the isomorphism $H_*(X) := H_*(C_*(X)) \cong H_*(HC_*)$, which allows one to compute the homology groups of the initial space X.

In this way, the notion of object with effective homology makes it possible to compute homology groups of complicated spaces by means of homology groups of effective complexes. The effective homology technique is based on the following idea: given some topological spaces $X_1, \ldots, X_n$, a topological constructor Φ produces a new topological space X. If effective homology versions of the spaces $X_1, \ldots, X_n$ are known, then one should be able to build an effective homology version of the space X, and this version would allow us to compute the homology groups of X.

A typical example of this kind of situation is the loop space constructor. Given a 1-reduced simplicial set X with effective homology, it is possible to determine the effective homology of the loop space $\Omega(X)$, which in particular allows one to compute the homology groups $H_*(\Omega(X))$. Moreover, if X is m-reduced, this process may be iterated m times, producing an effective homology version of $\Omega^k(X)$, for $k \leq m$. Effective homology versions are also known for classifying spaces or total spaces of fibrations, see [RS06] for more information.

Taking into account these ideas, we are tempted to try to use the effective homology technique to compute also the homology of a group G. To this aim, we consider the Eilenberg-MacLane space $K(G, 1)$, whose homology groups coincide with those of G. The size of this space makes it difficult to calculate the groups in a direct way, but it is possible to operate with this simplicial set making use of the *effective homology* technique: if we construct the effective homology of $K(G, 1)$ then we would be able to *efficiently* compute the homology groups of $K(G, 1)$, which are those of G. Furthermore, it would be possible to use the space $K(G, 1)$ inside different constructions. We can introduce therefore the following definition.

Definition 10. A group G is a *group with effective homology* if $K(G, 1)$ is a simplicial set with effective homology.

The problem is, given a group G, how can we determine the effective homology of $K(G, 1)$? If the group G is finite, the simplicial set $K(G, 1)$ is effective too, so that it has trivial effective homology. However, the enormous size of this space makes it difficult to obtain real calculations, and therefore we will try to obtain an equivalence $C_*(K(G, 1)) \Lleftarrow\!\!\Rrightarrow E_*$ where E_* is effective and (much) smaller than the initial complex. The main result of this paper, presented in Section 3, is an algorithm which computes this equivalence from a resolution of G.

2.3 Kenzo, GAP and HAP

GAP and Kenzo are two different programs devoted to Symbolic Computation, that up to now have followed separate lines and do not have many calculations in common.

One the one hand, Kenzo [DRSS99] is a Common Lisp program devoted to Symbolic Computation in Algebraic Topology. It was developed by Francis Sergeraert and some coworkers, and makes use of the effective homology method to determine homology groups of complicated spaces; it has obtained some results (for example homology groups of iterated loop spaces of a loop space modified by a cell attachment, components of complex Postnikov towers, etc.) which had never been determined before. In principle Kenzo is not intended to compute homology of groups and it does not know what a resolution is, although it implements Eilenberg-MacLane spaces $K(G, n)$ for every n but only for $G = \mathbb{Z}$ and $G = \mathbb{Z}_2$.

On the other hand, GAP [GAP] is a system for computational discrete algebra, with particular emphasis on Computational Group Theory. In our work we focus our attention on HAP, which is a homological algebra library for use with GAP, written by the second author of the paper and still under development. The initial focus of this package is on computations related to the cohomology of groups. A range of finite and infinite groups are handled, with particular emphasis on integral coefficients. It also contains some functions for the integral (co)homology of: Lie rings, Leibniz rings, cat-1-groups and digital topological spaces. In particular, HAP allows one to obtain (small) resolutions of many different groups, but it does not implement the bar resolution nor Eilenberg-MacLane spaces $K(G,1)$.

In this work, we try to relate both systems: we implement the spaces $K(G,1)$ in Kenzo for other groups G and then take a resolution from HAP to determine its effective homology, which will make it possible to determine the homology groups of G and make use of $K(G,1)$ (in an *effective* way) in other constructions.

3. AN ALGORITHM CONSTRUCTING THE EFFECTIVE HOMOLOGY OF A GROUP FROM A RESOLUTION

Let us suppose that G is a group and a (small) free resolution

$$\cdots \longrightarrow F_2 \xrightarrow{d_2} F_1 \xrightarrow{d_1} F_0 \xrightarrow{\varepsilon} F_{-1} = \mathbb{Z} \longrightarrow 0$$

is provided. Furthermore, let us assume that the resolution F_* is given with a *contracting homotopy*, that is to say, Abelian group morphisms $h_n : F_n \rightarrow F_{n+1}$ for each $n \geq -1$ (in general not compatible with the G-action), such that

$$\begin{aligned} \varepsilon h_{-1} &= \mathrm{Id}_{\mathbb{Z}} \\ h_{-1}\varepsilon + d_1 h_0 &= \mathrm{Id}_{R_0} \\ h_{i-1}d_i + d_{i+1}h_i &= \mathrm{Id}_{R_i}, \quad i > 0 \end{aligned}$$

On the other hand, we can also consider the bar resolution of G, $B_* = \mathrm{Bar}_*(G)$, with augmentation ε' and contracting homotopy h' (see [Bro82] for details about the definition of these maps).

As B_* and F_* are free resolutions for G, it is well known [Bro82] that one can construct a morphism of chain complexes of $\mathbb{Z}G$-modules $f : B_* \rightarrow F_*$ (compatible with the augmentations ε and ε'), and such that f is a homotopy equivalence. The explicit definition of this morphism can be found in [Bro82] and is included in the following lines.

For degree -1 we consider $f_{-1} = \mathrm{Id} : \mathbb{Z} \rightarrow \mathbb{Z}$. For each $n \geq 0$ we take $\{u_\alpha^n\}_\alpha$ a $\mathbb{Z}G$-basis of B_n (which is a free $\mathbb{Z}G$-module), and then we give a definition of f_n over each generator u_α^n. This definition is then extended by linearity over all elements of B_n, which implies that each f_n is a morphism of $\mathbb{Z}G$-modules.

First of all, f_0 is given by

$$f_0(u_\alpha^0) = h_{-1}\varepsilon'(u_\alpha^0)$$

Once we have defined $f_{n-1} : B_{n-1} \rightarrow F_{n-1}$, we consider

$$f_n(u_\alpha^n) = h_{n-1}f_{n-1}d_n(u_\alpha^n)$$

In a similar way, one can construct an augmentation-preserving morphism of chain complexes of $\mathbb{Z}G$-modules $g : F_* \rightarrow B_*$ given by

$$\begin{aligned} g_{-1} &= \mathrm{Id} : \mathbb{Z} \rightarrow \mathbb{Z} \\ g_0(v_\alpha^0) &= h'_{-1}\varepsilon(v_\alpha^0) \\ g_n(v_\alpha^n) &= h'_{n-1}g_{n-1}d_n(v_\alpha^n), \quad n \geq 1 \end{aligned}$$

where $\{v_\alpha^n\}_\alpha$ is a basis of the $\mathbb{Z}G$-module F_n.

In order to prove that f and g are homotopy equivalences, we construct graded morphisms of $\mathbb{Z}G$-modules

$$k : F_* \rightarrow F_{*+1}, \qquad k' : B_* \rightarrow B_{*+1}$$

such that $d_F k + k d_F = \mathrm{Id}_F - fg$ and $d_B k' + k' d_B = \mathrm{Id}_B - gf$.

The explicit expressions are not included in the classical texts about this subject but are not difficult to deduce. For degree -1, $k_{-1} : \mathbb{Z} \rightarrow F_0$ is the null map. For $n \geq 0$, the homotopy operator k can be defined inductively (over the generators of each $\mathbb{Z}G$-module F_n) as

$$\begin{aligned} k_0(v_\alpha^0) &= h_0(v_\alpha^0) - h_0 f_0 g_0(v_\alpha^0) \\ k_n(v_\alpha^n) &= h_n(\mathrm{Id}_{F_n} - f_n g_n - k_{n-1}d_n)(v_\alpha^n) \end{aligned}$$

It is not difficult to prove then that $d_{n+1}k_n + k_{n-1}d_n = \mathrm{Id}_{F_n} - f_n g_n$ for every $n \geq 0$. Analogously we can define $k' : B_* \rightarrow B_{*+1}$ satisfying $d_B k' + k' d_B = \mathrm{Id}_B - gf$.

We have therefore a homotopy equivalence (in the classical sense):

$$k' \circlearrowright B_* \underset{g}{\overset{f}{\rightleftarrows}} F_* \circlearrowleft k$$

where the four components f, g, k and k' are compatible with the action of the group G.

If we apply now the functor $\mathbb{Z} \otimes_{\mathbb{Z}G} -$, which is additive, we obtain an equivalence of chain complexes (of $\mathbb{Z}$-modules):

$$k' \circlearrowright \mathbb{Z} \otimes_{\mathbb{Z}G} B_* \underset{g}{\overset{f}{\rightleftarrows}} \mathbb{Z} \otimes_{\mathbb{Z}G} F_* \circlearrowleft k$$

and both chain complexes provide us the homology of the initial group G, that is,

$$H_*(\mathbb{Z} \otimes_{\mathbb{Z}G} B_*) \cong H_*(\mathbb{Z} \otimes_{\mathbb{Z}G} F_*) \equiv H_*(G)$$

In order to obtain a strong chain equivalence (in other words, a pair of reductions, following the framework of effective homology), we make use of the mapping cylinder construction.

Let us consider now a "general" chain equivalence of $\mathbb{Z}$-modules

$$h \circlearrowright A_* \underset{g}{\overset{f}{\rightleftarrows}} B_* \circlearrowleft k$$

where $f : A_* \rightarrow B_*$ and $g : B_* \rightarrow A_*$ are chain complex morphisms and $h : A_* \rightarrow A_{*+1}$ and $k : B_* \rightarrow B_{*+1}$ are graded group morphisms such that

$$\begin{aligned} gf &= \mathrm{Id}_A - d_A h - h d_A \\ fg &= \mathrm{Id}_B - d_B k - k d_B \end{aligned}$$

The mapping cylinder Cylinder$(f)_* \equiv C_*$ is a chain complex given by

$$C_n = A_{n-1} \oplus B_n \oplus A_n$$

with differential map given by the matrix

$$D_C = \begin{bmatrix} -d_A & 0 & 0 \\ f & d_B & 0 \\ -1 & 0 & d_A \end{bmatrix}$$

that is to say, $d_C(a_{n-1}, b_n, a_n) = (-d_A(a_{n-1}), f(a_{n-1}) + d_B(b_n), -a_{n-1} + d_A(a_n))$.

A reduction $\rho_B : \text{Cylinder}(f)_* \Rrightarrow B_*$ can be constructed for every chain map f (not necessarily a homotopy equivalence), where $\rho_B = (F_B, G_B, H_B)$ with

$$F_B(a_{n-1}, b_n, a_n) = b_n + f(a_n)$$
$$G_B(b_n) = (0, b_n, 0)$$
$$H_B(a_{n-1}, b_n, a_n) = (-a_n, 0, 0)$$

The difficult part of the wanted strong equivalence is the construction of a reduction $\rho_A : \text{Cylinder}(f)_* \Rrightarrow A_*$, where we should use the fact that f is a homotopy equivalence, in other words, we should take into account the components g, h and k. The formulas for the three elements of the reduction $\rho_A = (F_A, G_A, H_A)$ can be deduced from [NL91]; they are given concretely by the matrices

$$F_A = \begin{bmatrix} -h & g & 1 \end{bmatrix}$$

$$G_A = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H_A = \begin{bmatrix} -h - gkf + gfh & g & 0 \\ -kkf + kfh & k & 0 \\ -hh - hgkf + hgfh + gkkf - gkfh & hg - gk & 0 \end{bmatrix}$$

One can prove that these maps satisfy the equations $F_A G_A = \text{Id}_A$ and $G_A F_A = \text{Id}_C - d_C H_A - H_A d_C$, so that we obtain a reduction $\rho_A : C_* = \text{Cylinder}(f)_* \Rrightarrow A_*$.

Considering now our reductions $\rho_B : \text{Cylinder}(f)_* \Rrightarrow B_*$ and $\rho_A : \text{Cylinder}(f)_* \Rrightarrow A_*$ we obtain a strong chain equivalence

$$A_* \overset{\rho_A}{\Lleftarrow} \text{Cylinder}(f)_* \overset{\rho_B}{\Rrightarrow} B_*$$

In our particular case, we have the (classical) equivalence

$$\overset{k'}{\curvearrowright}\ \mathbb{Z} \otimes_{\mathbb{Z}G} B_* \underset{g}{\overset{f}{\rightleftarrows}} \mathbb{Z} \otimes_{\mathbb{Z}G} F_*\ \overset{k}{\curvearrowleft}$$

so that we can construct a strong equivalence

$$\mathbb{Z} \otimes_{\mathbb{Z}G} B_* \overset{\rho'}{\Lleftarrow} \text{Cylinder}(f)_* \overset{\rho}{\Rrightarrow} \mathbb{Z} \otimes_{\mathbb{Z}G} F_*$$

Now we recall that the left chain complex $\mathbb{Z} \otimes_{\mathbb{Z}G} B_*$ is equal to $C_*(K(G,1))$. If we suppose that the initial resolution F_* is of finite type (and small), then the right chain complex $\mathbb{Z} \otimes_{\mathbb{Z}G} F_* \equiv E_*$ is effective (and small too), so that we have obtained the searched effective homology of $K(G,1)$.

ALGORITHM 1. Input: *a group G and a free resolution F_* of finite type with contracting homotopy.*
Output: *the effective homology of $K(G,1)$, that is, a (strong chain) equivalence $C_*(K(G,1)) \Lleftarrow\!\!\Rrightarrow E_*$ where E_* is an effective chain complex.*

The effective homology of $K(G,1)$ makes it possible to determine the homology groups of G, and, what is more interesting, once we have $K(G,1)$ with its effective homology we could apply different constructors and obtain the effective homology of the results. For instance, if G is Abelian, one can apply the classifying space constructor and obtain the effective homology of $\overline{\mathcal{W}}(K(G,1)) = K(G,2)$, which could be useful in order to compute the homology of a 2-type, as we will see later.

Algorithm 1 has been implemented in Common Lisp enhancing the Kenzo system. The first step has been to create a new class `GROUP` with a slot `resolution`. This resolution is then used to compute the effective homology of the simplicial set $K(G,1)$, as seen in the following example.

We consider the cyclic group of order 5. We construct it with our Lisp function `CyclicGroup` and store it in the variable `C5`. In this case, at the same time the group is built, a (small) resolution of it is automatically stored in a slot of `C5` called `resolution`. It is a reduction from the $\mathbb{Z}G$-chain complex $K2$ to the trivial chain complex $K5 \equiv \mathbb{Z}$. This resolution allows us to compute the homology groups of $G = C5$:

```
> (setf C5 (CyclicGroup 5))
[K1 Abelian-Group]
> (setf F (resolution C5))
[K10 Reduction K2 => K5]
> (orgn (k 2))
(zg-chain complex for [K1 Abelian-Group])
> (orgn (k 5))
(z-chcm)
> (homology C5 3)
Homology in dimension 3 :
Component Z/5Z
---done---
```

Let us emphasize that in this particular case the homology of the group `C5` could also be determined using the Bar resolution $B_* = \text{Bar}_*(C5)$, which in this case produces an effective chain complex $\mathbb{Z} \otimes_{\mathbb{Z}G} B_* \equiv E_*$. However, the size of this space is much bigger than the resolution used, and therefore computations could only be done for low dimensions.

This resolution will be also used in the computation of the effective homology of $K(G,1)$. With the following instruction we construct this space; since `C5` is Abelian, we obtain a simplicial Abelian group.

```
> (setf KG1 (K-G-1 C5))
[K17 Abelian-Simplicial-Group]
```

The effective homology of a space is obtained with the function `efhm`. In our case we observe that the right chain complex $K11$ is produced from $K2$ by applying the functor $\mathbb{Z} \otimes_{\mathbb{Z}G} -$, while the top *big* chain complex is obtained making use of the cylinder construction, as explained in the previous lines.

```
> (efhm KG1)
[K54 Homotopy-Equivalence K17 <= K44 => K11]
> (orgn (k 11))
(tensor-with-integers [K2 zg-chain complex for [K1 Abelian-Group]])
> (orgn (k 44))
(cylinder [K37 Morphism (degree 0): K17 -> K11])
```

The small (effective) resolution associated with the group $G = C5$ is the one explained at the end of Section 2.1. It could be exported from HAP (see the following sections) or, in this particular case, it could be directly implemented in Kenzo. For more complicated groups it is not so easy to deduce and implement such a resolution; instead of programming it directly in Kenzo, we will try the first method: to obtain it from the system HAP.

4. EXPORTING RESOLUTIONS FROM HAP

As explained in Section 2.3, the GAP package HAP allows one to obtain resolutions of many different groups, making it possible to compute their homology. Our goal consists in using these resolutions in Kenzo: we want to use HAP to produce resolutions of some groups and import them into Kenzo to construct the effective homology of spaces $K(G, 1)$, which will be later involved in new constructions.

In order to export resolutions from HAP, we use OpenMath [OM], an XML standard for representing mathematical objects. There exist OpenMath translators from several Computer Algebra systems, and in particular GAP includes a package [SC] which produces OpenMath code from some GAP elements (lists, groups...). We have extended this package in order to represent resolutions. A resolution in OpenMath will be given by 5 elements: group, highest degree, list of ranks of each $\mathbb{Z}G$-module, boundary map and contracting homotopy.

First of all, for the group we use the representation already defined in the OpenMath package. For instance, the permutation group generated by the elements $(3, 2, 1)$ and $(1, 3, 5, 4, 2)$ will be given by:

```
<OMA>
     <OMS cd="group1" name="Group"/>
     <OMA>
          <OMS cd="permut1" name="Permutation"/>
          <OMI> 3</OMI>
          <OMI> 2</OMI>
          <OMI> 1</OMI>
     </OMA>
     <OMA>
          <OMS cd="permut1" name="Permutation"/>
          <OMI> 1</OMI>
          <OMI> 3</OMI>
          <OMI> 5</OMI>
          <OMI> 4</OMI>
          <OMI> 2</OMI>
     </OMA>
</OMA>
```

The highest degree of the resolution is simply an integer number, therefore will be denoted inside `<OMI>` and `</OMI>`.

```
<OMI> 6</OMI>
```

The next element of the resolution is the list of ranks of each $\mathbb{Z}G$-module, that is, a list of integers, one for each degree from 0 to the highest one. For the permutation group already constructed, we obtain the following list:

```
<OMA>
     <OMS cd="list1" name="list"/>
     <OMI> 1</OMI>
     <OMI> 3</OMI>
     <OMI> 6</OMI>
     <OMI> 10</OMI>
     <OMI> 15</OMI>
     <OMI> 20</OMI>
     <OMI> 26</OMI>
</OMA>
```

The description of the $\mathbb{Z}G$-boundary and the contracting homotopy is not so easy. These two maps are represented as lists containing the images of the generators of each module F_i, which are $\mathbb{Z}G$-combinations. For instance, in degree 1 $F_1 = (\mathbb{Z}G)^3$ has three generators. For the first one, its boundary is the combination $1 * (g_2, z_1) - 1 * (g_1, z_1)$ where g_i is the i-element of the group G and z_i is the i-generator of $F_0 = \mathbb{Z}G$. It is represented in OpenMath as:

```
<OMA>
     <OMS cd="resolutions" name="zgcombination"/>
     <OMA>
          <OMS cd="resolutions" name="zgterm"/>
          <OMI> 1</OMI>
          <OMA>
               <OMS cd="resolutions" name="zggnrt"/>
               <OMI> 2</OMI>
               <OMI> 1</OMI>
          </OMA>
     </OMA>
     <OMA>
          <OMS cd="resolutions" name="zgterm"/>
          <OMI> -1</OMI>
          <OMA>
               <OMS cd="resolutions" name="zggnrt"/>
               <OMI> 1</OMI>
               <OMI> 1</OMI>
          </OMA>
     </OMA>
</OMA>
```

Similar $\mathbb{Z}G$-combinations are obtained for the generators 2 and 3 of $F_1 = (\mathbb{Z}G)^3$, and the same process is applied for each degree. Some examples of OpenMath representations of resolutions written by our methods can be found in [Rom]. In the same site one can read the Content Dictionary formalizing all the OpenMath tags involved in our description.

This OpenMath code for resolutions should be imported by Kenzo in order to use them directly without the need of constructing them. Once the resolution is defined in Kenzo, we could use it to determine the effective homology of $K(G, 1)$ as explained in Section 3. Some examples of application of these constructions are presented in the next section.

5. APPLICATIONS AND EXAMPLES

5.1 Homology of cyclic groups

Let $G = C_m$ be the cyclic group of order m. As seen before, it is not difficult to construct a resolution F_* of G. This allows one to compute some homology groups of every cyclic group C_m. For instance, for $m = 7$:

```
> (setf C7 (cyclicgroup 7))
[K55 Abelian-Group]
> (resolution C7)
[K62 Reduction K56 => K5]
> (homology C7 3 6)
Homology in dimension 3 :
Component Z/7Z
---done---
Homology in dimension 4 :
---done---
Homology in dimension 5 :
Component Z/7Z
---done---
```

The same resolution can also be imported from HAP. To this aim, we make HAP to write the OpenMath code in a file "resolutionC7.txt" (see [Rom]) and then import it into Kenzo with the instruction `OmparseNextObject`.

```
> (setf rsltnC7 (OmparseNextObject
                 (filetostring "resolutionC7.txt")))
[K75 Reduction K69 => K5]
> (orgn rsltnC7)
(resolution of [K55 Abelian-Group] obtained from hap)
```

If we assign it to the slot `resolution` of `C7`, then this resolution will be used to compute the homology of the group. As expected, we obtain the same result.

```
> (setf (slot-value C7 'resolution) rsltnC7)
[K75 Reduction K69 => K5]
> (homology C7 3 6)
Homology in dimension 3 :
Component Z/7Z
---done---
Homology in dimension 4 :
---done---
Homology in dimension 5 :
Component Z/7Z
---done---
```

These examples are very simple, but the case presented in Section 4 is already more interesting: to construct small resolutions for permutation groups is challenging, and we need the expert knowledge implemented in HAP to import it into Kenzo. But even the simpler case of cyclic groups can give interesting applications when combining HAP with all the power of Kenzo, as we will show in the two following subsections.

5.2 Computations with K(G,n)'s

The first real application of our results is that we have allowed Kenzo to compute the effective homology of the spaces $K(G, n)$ for every Abelian group G and all $n \geq 1$, provided that HAP knows how to compute a resolution of G. In particular, it is the case for the cyclic groups C_m of order m.

Given $n = 1$, our Algorithm 1 provides us the effective homology of $K(G, 1)$. We have already seen an example of computation in Section 3. Let us consider now $G = C_7$.

```
> (setf KC71 (K-G-1 C7))
[K82 Abelian-Simplicial-Group]
> (efhm KC71)
[K119 Homotopy-Equivalence K82 <= K109 => K76]
```

Since $G = C_7$ is Abelian, $K(G, 1)$ is a simplicial Abelian group, and we can apply the classifying space constructor $\overline{\mathcal{W}}$ which gives us $\overline{\mathcal{W}}(K(G, 1)) = K(G, 2)$, which is also a simplicial Abelian group *with effective homology.*

```
> (setf KC72 (classifying-space KC71))
[K120 Abelian-Simplicial-Group]
> (efhm KC72)
[K259 Homotopy-Equivalence K120 <= K249 => K245]
> (homology KC72 3 6)
Homology in dimension 3 :
---done---
Homology in dimension 4 :
Component Z/7Z
---done---
Homology in dimension 5 :
---done---
```

Iterating the process, $K(G, n) = \overline{\mathcal{W}}(K(G, n-1))$ has effective homology for every $n \in \mathbb{N}$. Our new Kenzo function `K-Cm-n` allows us to construct $K(C_m, n)$; we observe that the slot `efhm` is directly constructed.

```
> (setf KC42 (K-Cm-n 4 2))
[K555 Abelian-Simplicial-Group]
> (efhm KC42)
[K729 Homotopy-Equivalence K555 <= K719 => K715]
> (homology KC42 4)
Homology in dimension 4 :
Component Z/8Z
---done---
```

This same technique allows to compute the effective homology of spaces $K(G, n)$, where G is a finitely generated Abelian group. In this case, the homology of $K(G, n)$ is one of the main ingredients to compute homotopy groups of spaces (see [RS02] and [RS06] for details).

5.3 An example of homology of a 2-type

Let us consider now $G = C_3$ the cyclic group of order 3. Let $A = \mathbb{Z}/3\mathbb{Z}$ be the Abelian group of three elements with trivial G-action (the groups G and A are in fact isomorphic; different notations are used to distinguish multiplicative and additive operations). Then the third cohomology group of G with coefficients in A is

$$H^3(G, A) = \mathbb{Z}/3\mathbb{Z}.$$

The elements of this cohomology group correspond to 2-types [Ell92] with $\pi_1 = G$ and $\pi_2 = A$. One such 2-type X corresponding to a non-trivial cohomology class $[f]$ in $H^3(G, A)$ can be seen as a twisted Cartesian product (simplicial version of a fibration, see [May67]) $X = K(A, 2) \times_f K(G, 1)$. It can be constructed by Kenzo in the following way:

```
> (setf K-C3-1 (K-Cm-n 3 1))
[K261 Abelian-Simplicial-Group]
> (setf chml-clss (chml-clss K-C3-1 3))
[K308 Cohomology-Class on K288 of degree 3]
> (setf tau (zp-whitehead 3 K-C3-1 chml-clss))
[K323 Fibration K261 -> K309]
> (setf x (fibration-total tau))
[K329 Kan-Simplicial-Set]
```

As explained in the previous example, $K(A, 2)$ and $K(G, 1)$ are objects with effective homology. From the two equivalences $C_*(K(A, 2)) \Longleftarrow\!\!\!\Rightarrow E_*$ and $C_*(K(G, 1)) \Longleftarrow\!\!\!\Rightarrow E'_*$, Kenzo knows how to construct the effective homology of the twisted Cartesian product $X = K(A,2) \times_f K(G,1)$, which makes it possible to determine its homology groups:

```
> (efhm x)
[K541 Homotopy-Equivalence K329 <= K531 => K527]
> (homology x 5)
Homology in dimension 5 :
Component Z/3Z
---done---
```

In the same way, the homology groups of $X = K(A,2) \times_f K(G,1)$ can be determined for all groups A and G with given (small) resolutions and cohomology class $[f]$ in $H^3(G,A)$. If the group G acts non-trivially on A, we obtain a different 2-type $X' = K(A,2) \times'_f K(G,1)$. In this case, to compute the effective homology of X' it would be necessary to include in Kenzo the construction of *induced* fibration (or induced twisted Cartesian product, in our simplicial setting); it should not be difficult as a further work.

6. CONCLUSIONS AND FURTHER WORK

In this paper we have reported on a successful experience to connect two computer algebra systems: GAP and Kenzo. The first one is devoted to Group Theory (and its package called HAP to homology of groups), and the second one to Algebraic Topology. An OpenMath link allows us to make them work together. Concretely, some resolutions are exported from HAP to Kenzo, allowing our programs to compute the effective homology of Eilenberg-MacLane spaces. Then, these spaces are used as ingredients in other Algebraic Topology constructions (namely, classifying spaces and fibrations), in order to get homology groups of 2-types, an important concept in homotopy theory [Ell92].

Obviously, one could re-program in Kenzo the algorithms already implemented in HAP, since Kenzo is a Common Lisp program which can be easily extended (things would be more difficult the other way around: the effective homology algorithms require higher order functional programming, and it seems that the GAP programming language [GAP] is not specially designed for this kind of task). But it would be more efficient, from the engineering point of view, to apply a separation of concerns principle: each system must be devoted to its own domain of expertise, and then systems should interoperate to get new and challenging results. Fortunately, technique is mature enough at this moment to undertake such a work. Our OpenMath link between HAP and Kenzo can be understood as a demonstration of this claim.

With respect to future research, two big lines are open. In the first one, Computational Group Theory could be applied to Algebraic Topology. This has been briefly evoked at the end of the previous section: a group can act non-trivially on a space, producing new interesting spaces (2-types in our example) where the Kenzo computation of homology groups could increase our knowledge of them.

As a second research line, more information on homology of groups could be extracted from the collaboration between the algebraic techniques in HAP and the topological ones in Kenzo. For instance, to investigate the homology of central extensions a topological approach was provided in [Rub97]; since this kind of extensions has been also dealt with in HAP, to compare *experimentally* both approaches could give a more complete view of it. To this aim it could be instrumental our program to explore spectral sequences of fibrations, explained in [RRS06].

7. ACKNOWLEDGMENTS

Partially supported by Ministerio de Educación y Ciencia (Spain), project MTM2006-06513.

8. REFERENCES

K. S. Brown. *Cohomology of Groups.* Springer-Verlag, 1982.

X. Dousson, J. Rubio, F. Sergeraert, and Y. Siret. The Kenzo program. Institut Fourier, Grenoble, 1999. http://www-fourier.ujf-grenoble.fr/~sergerar/Kenzo.

G. Ellis. HAP package for GAP. http://www.gap-system.org/Packages/hap.html.

G. J. Ellis. Homology of 2-types. *J. London Math. Soc.*, 46(2), pp. 1–27, 1992.

GAP - Groups, Algorithms, Programming - a System for Computational Discrete Algebra. http://www.gap-system.org/.

S. MacLane. *Homology*, volume 114 of *Grundleren der Mathematischen Wissenschaften*. Springer, 1963.

J. P. May. *Simplicial objects in Algebraic Topology*, volume 11 of *Van Nostrand Mathematical Studies*. 1967.

D. W. Narnes and L. A. Lambe. A fixed point approach to homological perturbation theory. *Proceedings of the AMS*, 112(3), pp. 881–892, 1991.

OpenMath. http://www.openmath.org/.

A. Romero. Computing Homology of Groups with the Kenzo System. http://www.unirioja.es/cu/anromero.

A. Romero, J. Rubio, and F. Sergeraert. Computing spectral sequences. *Journal of Symbolic Computation*, 41(10), pp. 1059–1079, 2006.

J. Rubio and F. Sergeraert. Constructive Algebraic Topology. *Bulletin des Sciences Mathématiques*, 126(5), pp. 389–412, 2002.

J. Rubio and F. Sergeraert. Constructive Homological Algebra and Applications, Lecture Notes Summer School on Mathematics, Algorithms, and Proofs. University of Genova, 2006. http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/Genova-Lecture-Notes.pdf.

J. Rubio. Integrating functional programming and symbolic computation. *Mathematics and Computers in Simulation*, 44, pp. 505–511, 1997.

A. Solomon and M. Costantini. GAP package OpenMath. http://www.gap-system.org/Packages/openmath.html.