An implementation of effective homotopy of fibrations

Ana Romero

University of La Rioja, Spain

Julio Rubio

University of La Rioja, Spain

Francis Sergeraert

Université Joseph Fourier, France

Abstract

In this paper, we present a new module for the Kenzo system which makes it possible to compute the effective homotopy of the total space of a fibration, using the well-known long exact sequence of homotopy of a fibration defined by Jean-Pierre Serre. The programs are written in Common Lisp and require the implementation of new classes and functions corresponding to the definitions of *setoid group (SG)* and *effective setoid group (ESG)*. Moreover, we have included a new module for working with finitely generated abelian groups, choosing the representation of a free presentation by means of a matrix in canonical form. These tools are then used to implement the long exact homotopy sequence of a fibration. We illustrate with examples some applications of our results.

Keywords: Constructive algebraic topology, effective homotopy, fibrations, homotopy groups, Serre exact sequence, finitely generated groups, central extensions, setoid groups, effective setoid groups.

1. Introduction

The problem of computing homology groups of topological spaces is well-known to be difficult, for example when loop spaces or classifying spaces are involved. In particular, knowing the homology groups of a topological group or space does not imply that the homology groups of its classifying space or loop space can also be determined. In the same way, given a fibration $F \hookrightarrow E \to B$, there does not exist a general algorithm computing the homology groups of E from the homology groups of B and F.

The *effective homology* method (introduced in Sergeraert (1994) and deeply explained in Rubio and Sergeraert (2002) and Rubio and Sergeraert (2006)) solves the previous problem when

Email addresses: `ana.romero@unirioja.es` (Ana Romero), `julio.rubio@unirioja.es` (Julio Rubio), `francis.sergeraert@univ-grenoble-alpes.fr` (Francis Sergeraert)

the input spaces belong to the large class of *spaces with effective homology* and produces *algorithms* for computing for example the homology groups of the total space of a fibration, of an arbitrarily iterated loop space (Adams' problem), of a classifying space, etc. One of the main ideas in this method is the notion of *a solution for the homological problem* of a space (Sergeraert (2009)), which provides in particular its homology groups and gives also some additional information which is necessary if we want to use the space inside more complicated constructions. The effective homology method has been concretely implemented in the Kenzo system (Dousson et al. (1999)), a Common Lisp program developed by the third author and some coworkers which has made it possible to compute some complicated homology groups so far unreachable.

The computation of homotopy groups is even harder than homology and is in fact one of the most challenging problems in the field of Algebraic Topology. In 1953 Serre obtained a general *finiteness result* (Serre (1953)) which states that, if X is a simply connected space such that the homology groups $H_n(X;\mathbb{Z})$ are of *finite type*, then the homotopy groups $\pi_n(X)$ for $n > 1$ are also abelian groups of finite type. In 1957, Edgar Brown published in Brown (1957) a *theoretical* algorithm for the computation of these groups, based on the Postnikov tower and making use of finite approximations of infinite simplicial sets, transforming in this way the finiteness results of Serre into a computability result. However, Brown's algorithm is out of practical use and has never been implemented. Rolf Schön's work (Schön (1991)) is an interesting systematic reorganization of Edgar Brown's algorithm but no implementation has yet been tried. Other theoretical methods have been also designed trying to determine homotopy groups, but up to our knowledge there does not exist a real implementation in a computer of a general algorithm producing the homotopy groups of an *arbitrary* simply connected space.

Inspired by the fundamental ideas of the effective homology method, in Romero and Sergeraert (2012) a new *effective homotopy* theory was introduced trying to allow the computation of *homotopy* groups of spaces. As in the case of effective homology, the idea consists in considering first some spaces whose effective homotopy can be directly determined, and then different constructors of Algebraic Topology (mainly limits and colimits, for example total space of a fibration, homogeneous space, etc.) should produce new spaces with effective homotopy. As a first result in this research, in Romero and Sergeraert (2012) we developed a *theoretical* algorithm to determine the effective homotopy of the total space of a fibration from the effective homotopies of the fiber and the base space.

In this work we present an *implementation* of the algorithm in Romero and Sergeraert (2012) by means of a new module for the Kenzo system, consisting of about 5000 lines of Common Lisp code available at `http://www.unirioja.es/cu/anromero/research2.html` (together with the most recent version of Kenzo). In a work in progress by Gerd Heber (HDF Group), a new Kenzo version easily loadable from and compatible with any Common Lisp system can be reachable by any GitHub user in the Jupyter page `https://lisp.style`. The new implementation requires the mathematical notions of *setoid group* (SG) (one of the ideas in *dependent type theories*, see Univalent Foundations Program (2013)) and *effective setoid group* (ESG), notions which carefully distinguish which is only *mathematically defined* but however not yet *computable*, from which is *constructively* reachable. Moreover, it includes new structures and functions for working with finitely generated abelian groups and computing kernels, cokernels and central extensions which are then used to implement our effective version of the classical long exact homotopy sequence of a fibration, so far not constructive. The chosen representation for groups is that of a free presentation by means of a matrix in canonical form.

Figure 1: Simplicial set Δ^2 and its horns Λ^2_0, Λ^2_1 and Λ^2_2

2. Preliminaries

In this section we introduce some elementary ideas about simplicial sets, which can be considered a useful combinatorial model for topological spaces. See May (1967) for more details.

Definition 1. *A* simplicial set *K is a simplicial object in the category of sets, that is to say, K consists of:*

- *a set K_q for each integer $q \geq 0$;*
- *for every pair of integers (i,q) such that $0 \leq i \leq q$,* face *and* degeneracy *maps $\partial_i : K_q \to K_{q-1}$ and $\eta_i : K_q \to K_{q+1}$ satisfying the* simplicial identities:

$$\begin{array}{lll} \partial_i\partial_j = \partial_{j-1}\partial_i & & \text{if } i < j \\ \eta_i\eta_j = \eta_{j+1}\eta_i & & \text{if } i \leq j \\ \partial_i\eta_j = \eta_{j-1}\partial_i & & \text{if } i < j \\ \partial_i\eta_j = \mathrm{Id} & & \text{if } i = j, j+1 \\ \partial_i\eta_j = \eta_j\partial_{i-1} & & \text{if } i > j+1 \end{array} \tag{1}$$

Definition 2. *For $n \geq 0$, the* standard n-simplex Δ^n *is a simplicial set built as follows. A q-simplex of Δ^n is any $(q+1)$-tuple $(a_0,\ldots,a_q)$ of integers such that $0 \leq a_0 \leq \cdots \leq a_q \leq n$, and the face and degeneracy operators are defined as*

$$\begin{array}{l} \partial_i(a_0,\ldots,a_q) = (a_0,\ldots,a_{i-1},a_{i+1},\ldots,a_q) \\ \eta_i(a_0,\ldots,a_q) = (a_0,\ldots,a_i,a_i,a_{i+1}\ldots,a_q) \end{array} \tag{2}$$

Note 3. *An interesting subcomplex of Δ^n is the "horn" Λ^n_k which is given by the boundary of the simplex Δ^n with the k-th face removed (see an example for $n = 2$ in Figure 1).*

Definition 4. *A simplicial set K is called a* Kan simplicial set *if it satisfies the following* extension condition*: for every collection of q simplices $x_0,x_1,\ldots,x_{k-1}$, $x_{k+1},\ldots,x_q \in K_{q-1}$, therefore of dimension $q-1$, which satisfy the compatibility condition $\partial_i x_j = \partial_{j-1}x_i$ for all $i < j, i \neq k$, and $j \neq k$, there exists a q-simplex $x \in K_q$ such that $\partial_i x = x_i$ for every $i \neq k$. In other words, K is a Kan simplicial set if every map from the horn $\Lambda^q_k \to K$ can be extended to a map from the simplex $\Delta^q \to K$. It is also said that K is a* Kan complex *or a* fibrant simplicial set.

Let us observe that the existence of the q-simplex x for each collection of $(q-1)$-simplices $x_0,x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_q$ satisfying the compatibility condition does not imply it is always possible to determine it. We say that the Kan simplicial set K is a *constructive Kan simplicial set* if the desired x can be produced by an algorithm σ_K. The constructive Kan property of a simplicial set will be needed later in this paper.

Definition 5. *Let K be a simplicial set. Two q-simplices x and y of K are said to be* homotopic*, written $x \sim y$, if $\partial_i x = \partial_i y$ for $0 \leq i \leq q$ and there exists a $(q+1)$-simplex z such that $\partial_q z = x, \partial_{q+1} z = y$, and $\partial_i z = \eta_{q-1}\partial_i x = \eta_{q-1}\partial_i y$ for $0 \leq i < q$.*

If K is a Kan simplicial set, then $\sim$ is an equivalence relation on the set of q-simplices of K for every $q \geq 0$.

Let $\star \in K_0$ be a base point; we also denote by $\star$ the degeneracies $\eta_{n-1} \dots \eta_0 \star \in K_n$ for every n. We define $S_n(K)$ as the set of all $x \in K_n$ such that $\partial_i x = \star$ for every $0 \leq i \leq n$. An element $x \in S_n(K)$ is called an *n-sphere* of K.

Definition 6. *Given a Kan simplicial set K and a base point $\star \in K_0$, we define*

$$\pi_n(K,\star) \equiv \pi_n(K) := S_n(K)/\!\sim$$

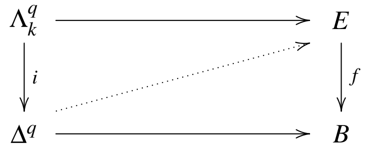
The set $\pi_n(K,\star)$ admits a group structure for $n \geq 1$ and it is abelian for $n \geq 2$ (see May (1967) for details). It is called the n-th homotopy group *of K.*

The previous definition of the homotopy groups $\pi_n(K)$ of a Kan simplicial set K is well-known to be *equivalent* to the notion of homotopy groups of a topological space; more precisely, the group $\pi_n(K)$ just combinatorially defined is canonically isomorphic to the standard homotopy group $\pi_n(|K|)$ of the realization $|K|$ of the simplicial set K. However, let us observe that the definition of $\pi_n(K)$ does not give any method computing it.

Since we aim to work with homotopy groups of simplicial sets in a constructive way, we only consider Kan simplicial sets whose homotopy groups $\pi_n(K)$ are abelian groups of finite type. In some specific cases, the results produced here could be generalized to simplicial sets K with non abelian π_1 (see for example the context of the paper Romero and Rubio (2013)), but we prefer at the current stage of our work to limit ourselves to the abelian case, so avoiding painful technicalities.

Definition 7. *Let $f : E \to B$ be a simplicial map. The map f is a* Kan fibration *if for every collection of q $(q-1)$-simplices $x_0, \dots, x_{k-1}, x_{k+1}, \dots, x_q$ of E which satisfy the compatibility condition $\partial_i x_j = \partial_{j-1} x_i, i < j, i \neq k, j \neq k$, and for every q-simplex y of B such that $\partial_i y = f(x_i), i \neq k$, there exists a q-simplex x of E such that $\partial_i x = x_i, i \neq k$, and $f(x) = y$. The simplicial set E (resp. B) is called the* total space *(resp. the* base space*) of the fibration. If Φ denotes the simplicial set generated by a vertex of B (usually the base point $\star$), then $F := f^{-1}(\Phi)$ is called the* fiber space *over Φ.*

In other words, a map f is a Kan fibration if for every commutative diagram of simplicial set morphisms



(3)

there is a map $\Delta^q \to E$ (dotted arrow) making the diagram commute. The map i is the obvious inclusion of Λ^q_k in Δ^q.

Later in this paper we will need the Kan property of a fibration to be constructive; we say that f is a *constructive Kan fibration* if the desired q-simplex x of E is produced by an algorithm σ_f. This idea has also been stated in a cubical framework as one of the main notions in *cubical type theories*, see Cohen et al. (2016).

3. Effective homotopy and setoid groups

As said in Section 2, the homotopy groups $\pi_n(K)$ of a Kan simplicial set K are formally defined as a quotient (see Definition 6) but this definition is not constructive since the nature of the components of this quotient, most often highly infinite, does not allow to deduce an algorithm computing it. The implementation of the effective homotopy theory, which will make it possible to produce algorithms computing the homotopy groups $\pi_n(K)$ for some Kan simplicial sets, needs to consider $\pi_n(K)$ as a *group defined over a setoid* as we explain in this section. The definition of *setoid group* (or group defined over a setoid) appears in the context of *dependent types theories* (see Univalent Foundations Program (2013) or the implementation in the Adga library Bove et al. (2009)). Our new necessary constructions of resolution for a central extension and exact sequence of setoid groups are explained in different subsections.

3.1. Finitely generated abelian groups

In order to develop our effective homotopy theory, we need first to work *effectively* with finitely generated abelian groups.

If G is a finitely generated abelian group, a finite sequence of generators $(g_1,\ldots,g_r)$ defines a surjective map $\varepsilon : \mathbb{Z}^r \to G$. The kernel $\operatorname{Ker}\varepsilon$ is isomorphic to $\mathbb{Z}^s$ with $s \le r$, giving a short resolution:

$$0 \to \mathbb{Z}^s \stackrel{\rho}{\to} \mathbb{Z}^r \stackrel{\varepsilon}{\to} G \to O \tag{4}$$

The *Smith reduction* (Kaczynski et al., 2004, Ch. 3) of the (r,s)-matrix ρ produces a Smith form for $\rho = \operatorname{Diag}(d_1,\ldots,d_s)$ for a sequence $(d_1,\ldots,d_s)$ of positive natural numbers satisfying the *divisor condition*: d_i divides d_{i+1} for $1 \le i < s$. Getting rid of the initial 1's of the divisor sequence gives the *minimal resolution*, where every divisor is > 1. So that $G \cong \mathbb{Z}/d_1 \oplus \cdots \oplus \mathbb{Z}/d_s \oplus \mathbb{Z}^{r-s}$ with $1 < d_1 \,|\, d_2 \,|\, \cdots \,|\, d_s$. These divisors and the rank $r-s$ define the isomorphism class of G.

As we will see in Section 5, a finitely generated abelian group will be represented in Kenzo by means of a matrix

$$M_G = \begin{bmatrix} d_1 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & d_s \\ 0 & \cdots & 0 \\ & \vdots & \\ 0 & \cdots & 0 \end{bmatrix} \tag{5}$$

where the number of rows with all entries equal to zero at the bottom of the matrix is equal to $r-s$. A group could also be represented by any general matrix or by means of the factors in the diagonal and the number for the block of 0's to add below, but the restriction of being a matrix in canonical form (which will be represented as a sparse matrix, therefore without any significant extra cost) is the option we have chosen to facilitate calculations on groups which will be necessary later (we will need to use the canonical matrix associated with a group several times and in this way we only construct it once).

3.2. Setoid groups

Definition 8. *A* setoid group (SG) *is a 5-tuple* $(A, \sim, *, 0, \mathrm{inv})$ *where:*

- *A is a set.*
- *'$\sim$' is an equivalence relation defined on A.*
- *'$*$' is a binary operation defined on A, compatible with the equivalence relation: if $a \sim a'$ and $b \sim b'$, then $a * b \sim a' * b'$.*
- *'$*$' is $\sim$-associative, that is, $(a * b) * c \sim a * (b * c)$.*
- *'0' is a $\sim$-neutral element: $a * 0 \sim a \sim 0 * a$.*
- *'*inv*' is a unary function producing a $\sim$-inverse of its argument:* $\mathrm{inv}(a) * a \sim 0 \sim a * \mathrm{inv}(a)$.

The quotient set $A/\sim$, setoid, is therefore provided with a group structure. In this text we consider only the *commutative case*: in other words, for every $a, b \in A$, the relation $a * b \sim b * a$ is satisfied and $A/\sim$ is abelian.

We are interested by situations where for some reason we *know* the quotient group $A/\sim =: G_A$ is of finite type, but we do not have a priori any mean to determine its isomorphism class. For example, when looking for the *effective homotopy* of a Kan simplicial set K, a setoid group can be the set $S^n(K)$ of the n-spheres of K for the Kan composition of spheres (May, 1967, §4), the n-th homotopy group being the quotient $\pi_n(K) = S^n(K)/\sim$ by the homotopy relation between spheres (Definition 6).

The space K and the set of spheres $S^n(K)$ most often are far from being of finite type, and also in general no algorithm can *decide* whether two given n-spheres are homotopic. The definition of $\pi_n(K)$ is therefore not at all *constructive*. In this *initial* situation, we are a priori unable to *compute* $\pi_n(K)$.

Sometimes we *know* that two particular n-spheres s and s' are homotopic, the proof being an $(n+1)$-simplex h describing the homotopy, as explained in Definition 5. If this is the case, we write $s \approx s'$. In other words, the relation $s \sim s'$ is a (hypothetical) predicate, true or false, while $s \approx s'$ is a "theorem". We use the same distinction for an arbitrary SG, the relation $a \approx a'$ meaning a proof of $a \sim a'$ is provided. Regarding to type theory, relations like $\approx$ are the default, and by a process of *propositional truncation* the relation $\sim$ is deduced (see Univalent Foundations Program (2013) for details).

We assume the properties required for the relation '$\sim$' for a SG are also satisfied for '$\approx$: for example, we are always able to produce a certificate for $(a * b) * c \approx a * (b * c)$ and, given certificates for $a \approx a'$ and $b \approx b'$, we can produce a certificate for $a * b \approx a' * b'$, for $a * b \approx b * a$, etc.

3.3. Effective Setoid Groups

Definition 9. *An* effective setoid group (ESG) *A is a tuple:*

$$A = (A, \sim, *, 0, \mathrm{inv}, G_A, f, g, h) \tag{6}$$

where:

- $(A, \sim, *, 0, \mathrm{inv})$ *is a setoid group.*
- *G_A is a finitely generated abelian group provided with an explicit minimal resolution via a Smith matrix (as explained in Subsection 3.1).*
- *$f : A \to G_A$ is a morphism with respect to the respective structures of A and G_A.*

- *$g : G_A \to A$ is a $\approx$-morphism: for every $a, b \in G_A$, the relation $g(a+b) \approx g(a) * g(b)$ is satisfied.*
- *The composition $fg : G_A \to G_A$ is the identity of G_A.*
- *The component h is a process which, given an arbitrary $a \in A$, produces a proof of $a \approx gf(a)$.*

The reader understands that an effective setoid group is a SG combined with an explicit isomorphism $A/\!\sim\, \cong G_A$, which group is in particular *known*. The map f gives *the* equivalence class $f(a) \in G_A$ of $a \in A$. The map g produces a representative $g(a) \in A$ for every element $a \in G_A$. If f and g are coherent, the relation $a \sim gf(a)$ must be satisfied for every a of A, and the component h produces a proof of $a \approx gf(a)$.

In other words, G_A is basically just the data needed to make $\sim$ into a decidable equivalence relation, so that an ESG is a group over a *decidable setoid*. The conditions are then that composition in one direction is *intensional equality*, while in the other direction is *propositional equality*. One can also think that an ESG is an explicit isomorphism between a group over an arbitrary setoid to a group with decidable equality, which is furthermore strict (in the categorical sense).

The definitions of SG and ESG make it possible to introduce now the main notion for the implementation of the effective homotopy theory.

3.4. Effective homotopy

Definition 10. *A* Kan simplicial set with effective homotopy *is a constructive Kan simplicial set K such that every homotopy group $\pi_n(K)$ is an ESG.*

As the reader can guess, the interesting point of this definition is the fact that, if a Kan simplicial set K has effective homotopy, then the homotopy groups $\pi_n(K)$ are given with the relatively sophisticated extra information allowing the user to work with them *effectively*. This can be used for example to compute the homotopy groups of the total space of a fibration as it will be explained in the following section.

Let us observe that this definition of the effective homotopy of a Kan simplicial is slightly different (but equivalent) to the definition introduced in Romero and Sergeraert (2012). The notions of setoid group and effective setoid group have been included now to make the definition clearer and more general, so that our Kenzo module can be used in other situations different from the effective homotopy context.

In the next two subsections we explain the constructions of resolution for a central extension and exact sequence of setoid groups, which are also necessary to determine the effective homotopy of the total space of a fibration.

3.5. Resolution for a central extension

If a short exact sequence of groups $0 \to A \to B \to C \to 0$ is given, it is common in various frameworks to deduce a resolution of B from resolutions of A and C. In particular, if A, B and C are abelian groups of finite type, a simple process produces a short resolution of B from short resolutions of A and C. This is explained for example in detail in (Knapp, 2008, Lemma 4.23), but it happens this proof is inoperative in our context. It is interesting to explain exactly where the obstacle is located, and to modify the proof accordingly.

Knapp's proof. In Figure 2, the bottom row $0 \to A \to B \to C \to 0$ is a given short exact sequence of finitely generated abelian groups, the left and right columns being given short resolutions of A and C. We must construct the central resolution of B. The $\mathbb{Z}$-module C_0 is free, allowing us to

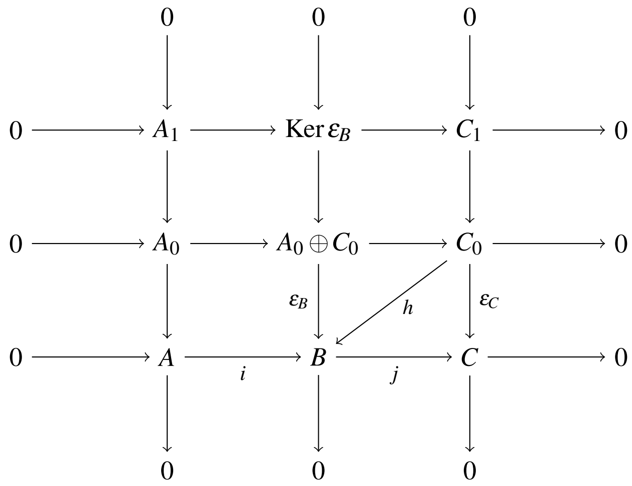


Figure 2: Knapp's proof

lift the map ε_C along j, producing the map h. Using this h, it is easy to define ε_B making both bottom squares commutative. Also, ε_B is surjective. Therefore $0 \to \operatorname{Ker}\varepsilon_B \to A_0 \oplus C_0 \to B \to 0$ is the looked-for resolution.

Well, but if B is in fact an *unknown* group G_B, the abelian group associated to a setoid group B, the knowledge of the arrow $\varepsilon_B : A_0 \oplus C_0 \to B$ does not allow to produce the kernel of ε_B. The equivalence relation '$\sim$' defining G_B from B is not effective, and examining whether some element $b = \varepsilon_B(a_0, c_0)$ of the setoid group B is equivalent to 0 or not *in general cannot be decided.*

Knapp's proof revisited. Considering first the classical situation where the group B is known, we give now a slightly different construction for the resolution of B. And we will see later this proof is again efficient in our contorted situation where the group B is an *unknown* group, defined as the quotient of a *setoid group*.

Consider Figure 3 with the same initial data. The map h is constructed as before and the same for the projection $\varepsilon_B := \bar{\varepsilon}_A \oplus \bar{\varepsilon}_C$ with $\bar{\varepsilon}_A := i\varepsilon_A$ and $\bar{\varepsilon}_C := h$. Now we describe how we can in fact *compute* the kernel of ε_B. It happens this kernel is necessarily isomorphic to $A_1 \oplus C_1$, the isomorphism being an injective map $A_1 \oplus C_1 \to A_0 \oplus C_0$ to be determined.

All the elements of our diagram are known except a map $\beta : C_1 \to A_0$. The key point is the following: the composition $\varepsilon_C\alpha_C$ is null but the composition $\bar{\varepsilon}_C\alpha_C$ in general is not.

To understand how this happens, please consider the simplest case of a non-trivial exact sequence, when $A = C = \mathbb{Z}/2$, $B = \mathbb{Z}/4$, $A_1 = A_0 = C_1 = C_0 = \mathbb{Z}$, the maps ε_A and ε_C are the canonical projections $\mathbb{Z} \to \mathbb{Z}/2$, the maps α_A and α_C are the multiplication by 2. Then the lift of the generator of C is the generator of B, so that h is also the canonical projection $\mathbb{Z} \to \mathbb{Z}/4$. This implies $\bar{\varepsilon}_C\alpha_C(1) = 2 \in \mathbb{Z}/4$, non null.

We will *twist* the direct sum $\alpha_A \oplus \alpha_C$ by the introduction of an appropriate $\beta : C_1 \to A_0$. Given a generator c_1 of C_1, the relation $j\bar{\varepsilon}_C\alpha_C = jh\alpha_C = \varepsilon_C\alpha_C = 0$ implies $b := \bar{\varepsilon}_C\alpha_C(c_1)$ is in $\operatorname{Ker} j = \operatorname{Im} i$. An i-preimage a of b can be lifted as $a_0 \in A_0$, and we decide $\beta(c_1) = -a_0$. It is clear this a_0 is constructed to obtain $(\beta(c_1), \alpha_C(c_1)) \in \operatorname{Ker}\varepsilon_B$. Doing the same work for the elements of a $\mathbb{Z}$-basis of C_1 produces a map $\beta \oplus \alpha_C : C_1 \to A_0 \oplus C_0$ having its image in $\operatorname{Ker}\varepsilon_B$.

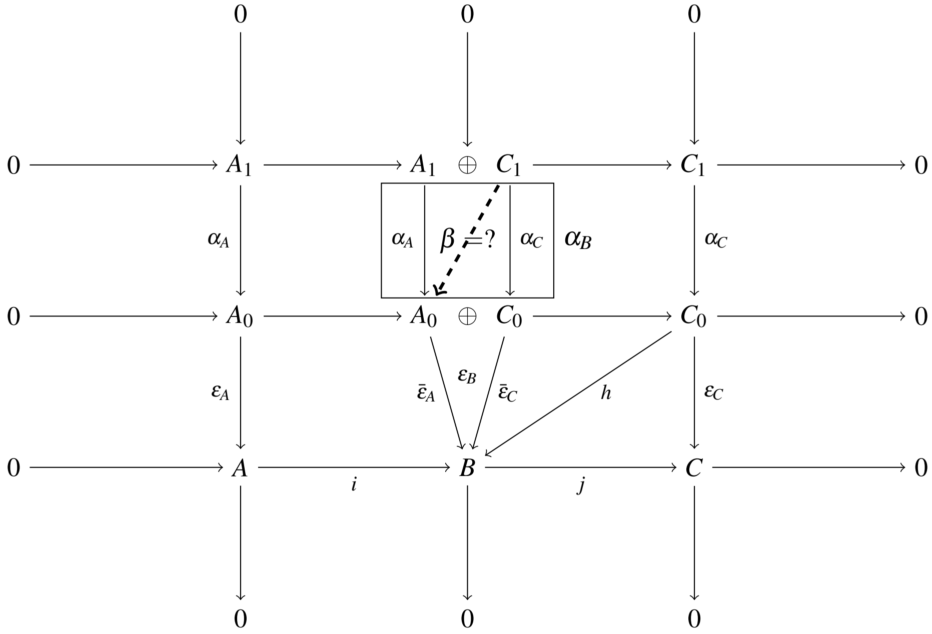


Figure 3: Knapp's proof revisited

Let us define $\alpha_B := \alpha_A \oplus (\beta \oplus \alpha_C) = \begin{pmatrix} \alpha_A & \beta \\ 0 & \alpha_C \end{pmatrix}$; it is clear all the squares of the diagram are commutative.

We claim that:

$$0 \to A_1 \oplus C_1 \xrightarrow{\alpha_B} A_0 \oplus C_0 \xrightarrow{\varepsilon_B} B \to 0 \tag{7}$$

is exact. The map β has been constructed to ensure this sequence is at least a chain complex; mainly the composition of both main maps is null, this is the reason of the choice of β. But the columns left and right of our diagram in Figure 3 are exact, all the horizontals are exact, so that the central column is exact as well. The sequence (7) is therefore a resolution of B which, if desired, can be made minimal by a Smith reduction of α_B.

3.6. Exact sequence of SGs

Definition 11. An (effectively) *exact sequence of SGs* is a sequence:

$$0 \to A \xrightarrow{i} B \xrightarrow{j} C \to 0 \tag{8}$$

where A, B and C are SGs, and i and j are $\approx$-morphisms. The composition ji is $\approx$-null; if $b \in B$ satisfies $jb \approx 0$, then an algorithm produces $a \in A$ satisfying $ia \approx b$. The map i is $\approx$-injective, that is, if $a \in A$ satisfies $i(a) \approx 0$, then an algorithm produces a proof of $a \approx 0$. Finally if $c \in C$, an algorithm produces $b \in B$ and a proof of $j(b) \approx c$.

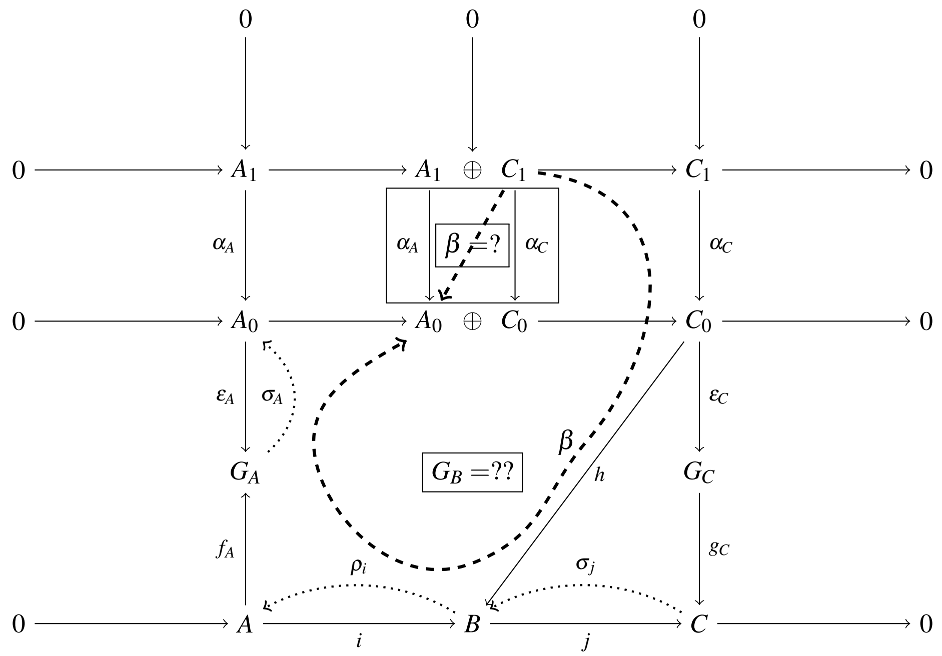


Figure 4: Theorem 12

In particular all the equivalence relations mentioned in this definition have the strong form '$\approx$', in other words they are assumed *with certificates*. For example, if $a \in A$, then we can produce a proof of $ji(a) \approx 0$. If $a, a' \in A$, we can produce a proof of $i(a * a') \approx i(a) * i(a')$, etc.

For such an exact sequence of SGs, the corresponding sequence of groups:

$$0 \to G_A \xrightarrow{i} G_B \xrightarrow{j} G_C \to 0 \tag{9}$$

is exact, but with these weak hypotheses, these groups are in general unknown.

Theorem 12. *Let:*

$$0 \to A \xrightarrow{i} B \xrightarrow{j} C \to 0 \tag{10}$$

be an exact sequence of SGs where A and C are ESGs. Then a canonical process produces an ESG structure for B. In particular the group G_B is so computed*, provided with a short resolution which is a twisted sum of the given resolutions of G_A and G_C.*

The appropriate diagram to prove this result is now Figure 4. The SG A being an *ESG*, a map $f_A : A \to G_A$ associating to every element of the setoid A its equivalence class in G_A is available. In the same way, C being an ESG, a map $g_C : G_C \to C$ is available, associating to every element of the "abstract" group G_C a "concrete" representative in the setoid C.

Because of the $\approx$-exactness of the bottom row, we can construct a set-theoretic section σ_j of the $\approx$-onto map j. And a set-theoretic retration ρ_i of the $\approx$-into map i. Also the provided resolution of G_A allows us to produce a set-theoretic section σ_A of ε_A.

The group G_B is not yet known, so that the map h, essential in Knapp's proof, can be defined here only as a map $h : C_0 \to B$. It is tempting to define h as the composition $\sigma_j g_C \varepsilon_C$ but this would be *erroneous*, for the h to be defined *must be* $\approx$*-linear*, which $\sigma_j g_C \varepsilon_C$ in general is not! The map h is first to be defined on the elements of a $\mathbb{Z}$-basis of C_0, using this time the composition $\sigma_j g_C \varepsilon_C$, and once this is done, h is the obvious linear extension to the whole C_0.

Now we can use the following path starting from the right component of $A_1 \oplus C_1$ up to the left component of $A_0 \oplus C_0$:

$$C_1 \xrightarrow{\alpha_C} C_0 \xrightarrow{h} B \xrightarrow{\rho_i} A \xrightarrow{f_A} G_A \xrightarrow{\sigma_A} A_0 \tag{11}$$

Following this path for every element of a $\mathbb{Z}$-basis of C_1, then linearly extending to C_0 defines a β map twisting the direct sum $\alpha_A \oplus \alpha_C$.

Now we *define* G_B as the cokernel of $\alpha_B := \begin{pmatrix} \alpha_A & \beta \\ 0 & \alpha_C \end{pmatrix}$; in particular the isomorphism class of G_B is a direct consequence of the Smith reduction of α_B. It is routine work to complete the diagram and produce all the missing maps $G_A \to G_B \to G_C$, also the maps $B \to G_B$ and $G_B \to B$ defining the ESG-structure of B, our main goal.

4. Effective homotopy of fibrations

In this section we present some new Kenzo functions for the computation of the effective homotopy of the total space of a fibration. The functions implement the following algorithm.

Algorithm 1. *(Romero and Sergeraert (2012))*

- ***Input:***
  - *A constructive Kan fibration $p : E \to B$ where B, the base space, is a constructive Kan complex (which implies the fiber space F and the total space E are also constructive Kan simplicial sets), and F or B is simply connected.*
  - *Respective effective homotopies of the simplicial sets F and B.*
- ***Output:*** *An effective homotopy for the Kan simplicial set E.*

The theoretical algorithm was presented in Romero and Sergeraert (2012) although the definition of *object with effective homotopy* given in Romero and Sergeraert (2012) is slightly different from the one introduced in Section 3 by means of the notions of *setoid group* and *effective setoid group*. The implementation uses auxiliary structures already present in Kenzo (simplicial sets, morphisms, etc.) and requires 5000 lines of new code including a representation for finitely generated abelian groups, setoid groups and effective setoid groups; we deal with this issue in the next section. In Table 1 the reader can find a *dictionary* with the different Kenzo classes introduced in the module and the corresponding mathematical objects.

First of all, new classes `KAN-WITH-EFHMT` (Kan simplicial set with effective homotopy), `SMGR-WITH-EFHMT` (simplicial group with effective homotopy) and `ABSMGR-WITH-EFHMT` (abelian simplicial group with effective homotopy) have been considered. These classes inherit respectively from the Kenzo classes `KAN` (representing Kan simplicial sets), `SIMPLICIAL-GROUP` and `AB-SIMPLICIAL-GROUP` (abelian simplicial group) and they include a new slot providing the effective homotopy of the object.

Kenzo class	Mathematical object
ABSMGR-WITH-EFHMT	Abelian simplicial group with effective homotopy
EFFECTIVE-SETOID-GROUP	Effective setoid group
ESG-MRPH	Morphism of effective setoid groups
FNGN-ABGROUP	Finitely generated abelian group
FNGN-ABGROUP-MRPH	Morphism of finitely generated abelian groups
KAN-FIBRATION	Kan fibration
KAN-WITH-EFHMT	Kan simplicial set with effective homotopy
SETOID-GROUP	Setoid group
SG-MRPH	Morphism of setoid groups
SMGR-WITH-EFHMT	Simplicial group with effective homotopy

Table 1: New Kenzo classes

For example, the class `KAN-WITH-EFHMT` has one slot[1]:

`efhmt` A function which inputs a degree n and returns an ESG.

The homotopy groups of an element of the class `KAN-WITH-EFHMT` (or of its subclasses `SMGR-WITH-EFHMT` and `ABSMGR-WITH-EFHMT`) can be determined by means of the new function `homotopy-group`.

Then, a *constructive Kan fibration* $F \hookrightarrow E \xrightarrow{p} B$ is implemented as an instance of the new class `KAN-FIBRATION` with the following slots:

`incl` A simplicial morphism representing the inclusion $i : F \hookrightarrow E$.
`incl-1` A morphism $i^{-1} : E \supset p^{-1}(*) \to F$, a partial inverse of i.
`fibr` The fibration $p : E \to B$.
`kfll` A function providing the constructive Kan property of the fibration.

The composition $\texttt{p} \circ \texttt{incl}$ is supposed to be the null simplicial morphism which sends every simplex $x \in F$ to the base point in B.

The main function of the new Kenzo module for computing the effective homotopy of the total space of a fibration is the function:

`kfbr-tot-efhmt` *fib n*

which inputs an object of the class `KAN-FIBRATION` and a non-negative integer and computes the effective homotopy of the total space E in that degree from the effective homotopies of B and F and the Kan properties of F, E, B and the fibration p. The result is an object of the class `EFFECTIVE-SETOID-GROUP`. The construction requires about 2000 lines of code and follows the idea of the proof of Algorithm 1 explained in Romero and Sergeraert (2012), although for the implementation our new structures of SG and ESG are used. Here we include a general sketch summarizing the main ideas of the theoretical proof and then we include some details about the implementation.

Given a fibration $F \hookrightarrow E \xrightarrow{p} B$ and $n \geq 0$, the proof starts with the long exact sequence of homotopy (May, 1967, Ch.II):

$$\begin{aligned} \cdots \xrightarrow{p_*} \pi_{n+1}(B) \xrightarrow{\partial} \pi_n(F) \xrightarrow{\text{inc}_*} \pi_n(E) \xrightarrow{p_*} \pi_n(B) \\ \xrightarrow{\partial} \pi_{n-1}(F) \xrightarrow{\text{inc}_*} \cdots \end{aligned} \tag{12}$$

[1]Several Lisp technical components without any interest here have been omitted.

where the maps p_* and inc_* are the morphisms between the corresponding homotopy groups induced respectively by the fibration $p : E \to B$ and the inclusion $F \hookrightarrow E$, and $\partial : \pi_*(B) \to \pi_{*-1}(F)$ is the *connecting morphism* (see (May, 1967, Ch.II) for the definition of this map).

From this one can deduce a short exact sequence

$$0 \longrightarrow \text{Coker} \xrightarrow{i} \pi_n(E) \xrightarrow{j} \text{Ker} \longrightarrow 0 \tag{13}$$

where $\text{Coker} \equiv \text{Coker}[\pi_{n+1}(B) \xrightarrow{\partial} \pi_n(F)]$ and $\text{Ker} \equiv \text{Ker}[\pi_n(B) \xrightarrow{\partial} \pi_{n-1}(F)]$, which implies the desired group $\pi_n(E)$ can be expressed as an extension of Ker by Coker, $\pi_n(E) \cong \text{Coker} \times_\gamma \text{Ker}$, for a cocycle $\gamma : \text{Ker} \times \text{Ker} \to \text{Coker}$ classifying the extension (see (Brown, 1982, Ch. IV.3)). Unfortunately, this cocycle is *a priori* unknown; so that the desired group $\pi_n(E)$ then appears as an unknown quotient of $S_n(E)$, a *setoid group*. Our task consists in determining the missing components f, g, h and $G = \pi_n(E)$ connecting $S_n(E)$ and $\pi_n(E)$, making of $S_n(E)$ an *effective setoid group*.

In order to compute the correct extension (and then the components G, f, g and h of the effective homotopy of E in degree n), one needs to make the short exact sequence *effective* (see Definition 11). This can be done by defining a set-theoretic section $\sigma : \text{Ker} \to \pi_n(E)$ and a set-theoretic retraction $\rho : \pi_n(E) \to \text{Coker}$ such that $\rho i = \text{Id}_{\text{Coker}}$, $i\rho + \sigma j = \text{Id}_{\pi_n(E)}$ and $j\sigma = \text{Id}_{\text{Ker}}$. Both maps can be defined regarding to the spheres $S_n(E)$ by means of a suitable game of successive applications of the Kan properties of B, F, E and the fibration p and the effective homotopies of B and F as explained in Romero and Sergeraert (2012).

The implementation of Algorithm 1 is divided in the following steps:

- For each integer n, the groups $\pi_{n+1}(B)$, $\pi_n(F)$, $\pi_n(B)$ and $\pi_{n-1}(F)$ are determined by means of the component G of the effective homotopies of B and F in the corresponding degrees (which are given as ESGs).

- The connecting morphisms $\pi_{n+1}(B) \xrightarrow{\partial} \pi_n(F)$ and $\pi_n(B) \xrightarrow{\partial} \pi_{n-1}(F)$ are computed following the definition of (May, 1967, Ch.II) and using again the effective homotopies of F and B and the constructive Kan property of the fibration p. Both of them are implemented as morphisms of ESGs.

- Next, we compute the kernel $\text{Ker} \equiv \text{Ker}[\pi_n(B) \xrightarrow{\partial} \pi_{n-1}(F)]$ and the cokernel $\text{Coker} = \text{Coker}[\pi_{n+1}(B) \xrightarrow{\partial} \pi_n(F)]$ by means of two new Kenzo functions called `mrph-kernel` and `mrph-cokernel` respectively. As we will explain in Section 5, the results of these functions produce also ESGs and their associated finitely generated abelian groups will be represented by means of finite matrices between free $\mathbb{Z}$-modules of finite type $M_K : K_1 \to K_0$ and $M_C : C_1 \to C_0$ respectively.

  The algorithm must now determine the correct extension of Ker by Coker to determine the group $\pi_n(E)$ which appears in the middle of the short exact sequence (13); the key point is the following: even if we do not know the isomorphism class of the group $\pi_n(E)$, the algorithm is able to determine the extension by making the short exact sequence effective and thanks to the effective setoid group structure of Ker and Coker .

- We construct the section $\sigma : \text{Ker} \to \pi_n(E)$ and the retraction $\rho : \pi_n(E) \to \text{Coker}$ following the definitions of Romero and Sergeraert (2012) (which make use of the Kan properties of F, E, B and p and the effective homotopies of B and F). Let us observe that $\pi_n(E)$ cannot

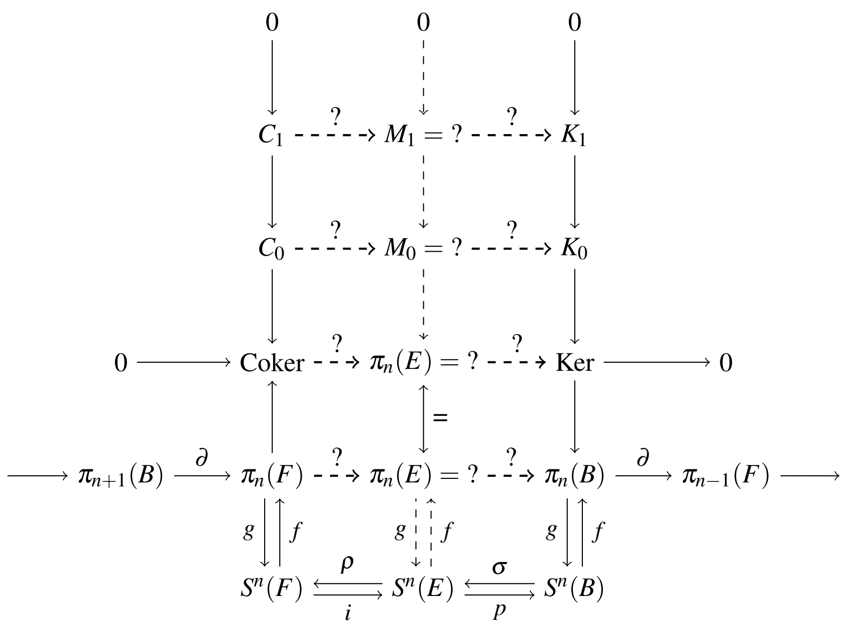


Figure 5: Algorithm 1

be represented yet as an ESG and then the SG structure of $S_n(E)$ introduced in Subsection 3.2 is used instead. In other words, σ and ρ are defined only with respect to the *spheres* representing the (unknown) elements of the group.

- Given the effectively exact sequence of SGs

$$0 \longrightarrow \text{Coker} \underset{\rho}{\overset{i}{\rightleftarrows}} S_n(E) \underset{\sigma}{\overset{j}{\rightleftarrows}} \text{Ker} \longrightarrow 1 \tag{14}$$

  the algorithm presented in Theorem 12 produces an ESG structure on $S_n(E)$, providing the looked-for effective homotopy of E in degree n.

The different steps of Algorithm 1 can be represented by means of the diagram of Figure 5, which in particular shows the relation Ker $\rightarrow \pi_n(B) \rightarrow S^n(B) \rightarrow S^n(E) \rightarrow S^n(F) \rightarrow \pi_n(F) \rightarrow$ Coker, which is then used to determine the free presentation $M_1 \rightarrow M_0$ for $\pi_n(E)$.

5. A Kenzo module for setoid grups and effective setoid groups

As a necessary ingredient for the computation of the effective homotopy of the total space of a fibration, a new Kenzo module has been developed dealing with finitely generated abelian groups, setoid groups and effective setoid groups. It consists of about 1300 lines of code containing functions to construct groups, SGs and ESGs and morphisms of these elements and computing kernels, cokernels and central extensions. The module also includes auxiliary functions dealing with matrices and computing their *canonical form* (see Subsection 5.1).

5.1. Finitely generated abelian groups

A finitely generated abelian group is implemented in Kenzo as an instance of a CLOS class, the class `FNGN-ABGROUP`, with the following main slot:

`mtrx` A matrix in canonical form corresponding to the minimal resolution of the group.

This relevant slot is stored in Kenzo by means of a sparse matrix of the form:

$$M_G = \begin{bmatrix} d_1 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & d_s \\ 0 & \cdots & 0 \\ & \vdots & \\ 0 & \cdots & 0 \end{bmatrix} \tag{15}$$

As said before, a group could also be represented by any general matrix (as for instance in Barakat and Bremer (2008), where a representation of finitely generated modules which works for any computable ring is presented) or by means of the factors in the diagonal and the number for the block of 0's to add below, but the restriction of `mtrx` being a sparse matrix in canonical form is included to facilitate calculations on groups which will be necessary later as explained in Subsection 3.1. To this aim, a function `canonical-representation` is provided computing the canonical form of a general matrix $N : N_1 \to N_0$ and returning a new matrix $N' : N'_1 \to N'_0$ as the one in (15). The canonical matrix N' is obtained by computing the Smith normal form of N and removing rows and lines corresponding to 1's and columns corresponding to 0's in the diagonal. The function `canonical-representation` also returns two matrices $R : N'_0 \to N_0$ and $R' : N_0 \to N'_0$ describing the relations between the original and the new generators (obtained also by means of the Smith normal form algorithm).

To facilitate the construction of instances of the class `FNGN-ABGROUP` we have provided the function

`build-fngn-abgroup :mtrx` *mtrx* `:divs` *divs*

defined with keyword parameters which allows one to construct a group either by means of a matrix or by its list of *divisors* of the form $(d_1,\ldots,d_s,0,\ldots,0)$. If the argument `mtrx` is not present, a matrix is constructed by using the parameter `divs`. The returned value is an instance of the class `FNGN-ABGROUP`.

Moreover, our module for finitely generated abelian groups includes some useful functions as for example the following ones.

`check-fngn-abgroup` *group*

Inform if the slot `mtrx` of the group *group* corresponds to a matrix in canonical form.

`divisors` *group*

Display on the screen the components $\mathbb{Z}/d_1,\ldots,\mathbb{Z}/d_s$ or $\mathbb{Z}$ of *group*.

`cyclic-group` *n*

Construct the cyclic group $\mathbb{Z}/n$ for $n \geq 2$. For $n = 0$, the infinite cyclic group $\mathbb{Z}$ is built.

`trivial-group`

Construct the trivial group $\mathbb{Z}/1 \equiv 0$.

In order to make the program more efficient, when constructing a finitely generated abelian group by means of the function `build-fngn-abgroup` the fact that the matrix `mtrx` is in canonical form is not checked. We follow in this way the general philosophy of Kenzo of *building* objects without checking if the inputs satisfy the necessary conditions (although in this case one can verify that the matrix is in canonical form in a later step by means of the function `check-fngn-abgroup`); in particular, let us remark that this kind of checks are not possible when working with objects of infinite nature, a frequent situation in the Kenzo program.

A morphism of finitely generated abelian groups is then defined by means of two matrices which must commute with the matrices defining source and target groups. In this case the matrices can have as entries any integer number and no restriction is considered.

The class `FNGN-ABGROUP-MRPH` has the following slots:

`sorc`	The source of the morphism, represented by means of a matrix $M_A : A_1 \to A_0$.
`trgt`	The target of the morphism, represented by means of a matrix $M_B : B_1 \to B_0$.
`mtrx0`	A matrix $M_0 : A_0 \to B_0$.
`mtrx1`	A matrix $M_1 : A_1 \to B_1$.

A morphism could in fact be defined only by a matrix M_0 such that there exists an M_1 which makes the diagram commute. A (non-unique) M_1, in case it exists, could be computed a posteriori to verify that M_0 defines a morphism. In our case, the matrix M_1 is also included in the construction of the morphism to facilitate later calculations.

The new function

`build-fngn-abgroup-mrph :sorc` *sorc* `:trgt` *trgt* `:mtrx0` *mtrx0* `:mtrx1` *mtrx1*

inputs two groups and two matrices and constructs an object of the class `FNGN-ABGROUP-MRPH`.

The main functions to deal with morphisms of finitely generated abelian groups are the following ones:

`check-fngn-abgroup-mrph` *mrph*

Inform if the morphism *mrph* is well defined, verifying if the groups `sorc` and `trgt` are well defined and the matrices `mtrx0` and `mtrx1` are compatible with matrices of source and target groups.

`mrph-cmps` *mrph1 mrph2*

Compute the composition of the morphisms *mrph1* and *mrph2*.

`mrph-zerop` *mrph*

Decide if the morphism *mrph* is the null morphism.

For example, let us consider the finitely generated abelian groups $A = \mathbb{Z}/6 \oplus \mathbb{Z}$ and $B = \mathbb{Z}/2$ and the morphism $f : A \to B$ given by the sum of the canonical projections $\mathbb{Z}/6 \to \mathbb{Z}/2$ and $\mathbb{Z} \to \mathbb{Z}/2$. This morphism is defined by means of the matrices $M_0 = [1\ 1]$ and $M_1 = [3]$.

$$
\begin{array}{ccc}
\mathbb{Z} & \xrightarrow{\begin{bmatrix} 3 \end{bmatrix}} & \mathbb{Z} \\
{\scriptstyle \begin{bmatrix} 6 \\ 0 \end{bmatrix}} \downarrow & & \downarrow {\scriptstyle \begin{bmatrix} 2 \end{bmatrix}} \\
\mathbb{Z}^2 & \xrightarrow{\begin{bmatrix} 1 & 1 \end{bmatrix}} & \mathbb{Z}
\end{array}
$$

The morphism is constructed in Kenzo with the following statements.

```
>(progn
  (setf a (build-fngn-abgroup :divs '(6 0)))
  (setf b (cyclic-group 2))
  (setf m0 (build-mtrx :nrow 1 :ncol 2
                       :entries '(1 1)))
  (setf m1 (build-mtrx :nrow 1 :ncol 1
                       :entries '(3)))
  (setf gr-mrph1 (build-fngn-abgroup-mrph :sorc a
       :trgt b :mtrx0 m0 :mtrx1 m1)))
```

```
[K3 Fngn-abgroup-mrph]
>(check-fngn-abgroup-mrph mrph)
T
```

The result is the finitely generated abelian group morphism $K3$ (the Kenzo object #3) which is well defined.

5.2. Setoid groups and effective setoid groups

A new class `SETOID-GROUP` has been defined with the following slots:

`elms`	A function which determines the elements of the SG.
`cmps`	A function which inputs two elements x and y of the setoid and returns the element $x * y$.
`nullel`	The 0 element of the setoid group.
`inv`	The inverse function.

Let us observe that the component $\sim$ of Definition 8, although is *mathematically* defined for the user, is not yet usable by the program: it is not included in the implementation.

Then, the new class `EFFECTIVE-SETOID-GROUP` inherits from the previous class `SETOID-GROUP` by adding 4 new slots:

`gr`	A finitely generated abelian group $G \cong \mathbb{Z}/d_1 \oplus \cdots \oplus \mathbb{Z}/d_s \oplus \mathbb{Z}^{r-s}$.
`f`	A function which inputs an element x of the setoid and outputs its corresponding element (class) $a = (a_1, \ldots, a_r) \in G$.
`g`	A function which inputs a list $a = (a_1, \ldots, a_r)$ defining an element of the group G and outputs an element x in the setoid representing this class.
`h`	A function which inputs an element x such that $f(x) = 0$ in G and outputs some *proof* of the fact that $x \sim 0$ in the setoid. The proof will depend on the nature of the ESG.

As before, the component $\sim$ is not included in the implementation although in this case it is implicitly defined by f. Moreover, let us remark that the component h does not correspond directly to the one in Definition 9 but it is equivalent and more convenient for our later constructions.

A morphism $f : A \to B$ for A and B SGs if given by the source, the target and the *internal* function defining the image of each element in A, so that the class `SG-MRPH` has the following slots:

`sorc`	The source of the morphism.
`trgt`	The target of the morphism.
`intr`	The internal function defining the morphism.

A morphism of effective setoid groups is represented by means of the new class `ESG-MRPH` which inherits from `SG-MRPH` and has the following new slots:

`mtrx0`	The matrix M_0 of the corresponding morphism of finitely generated abelian groups.
`mtrx1`	The matrix M_1 of the corresponding morphism of finitely generated abelian groups.

These matrices can be introduced directly when building the morphism or can be deduced later from the internal function. Moreover, the slots `sorc` and `trgt` must be ESGs.

In order to construct an example of morphism of ESGs, let us consider Eilenberg-MacLane spaces $K(\pi,n)'s$, which are implemented in our new module of Kenzo as objects with effective homotopy. Let π be a finitely generated abelian group, the simplicial group $K(\pi,n)$ satisfies $\pi_n(K(\pi,n)) \cong \pi$ and $\pi_i(K(\pi,n)) = 0$ for every $i \neq n$. Moreover, one can observe that in fact

$S_n(K(\pi,n)) = K(\pi,n)_n \cong \pi$ and $S_i(K(\pi,n)) = \{\star\} \cong 0$ for $i \neq n$, which makes it possible to define the required components `gr`, `f`, `g` and `h` of the effective homotopy of $K(\pi,n)$ in each degree in an easy way (see Romero and Sergeraert (2012) for details).

For instance, considering again the groups $A = \mathbb{Z}/6 \oplus \mathbb{Z}$ and $B = \mathbb{Z}/2$ built in Subsection 5.1, we construct now the spaces $K(A,1)$ and $K(B,1)$ as follows:

```
> (setf ka1 (k-g a 1))
[K5 Abelian-Simplicial-Group-with-Effective-Homotopy]
> (setf kb1 (k-g b 1))
[K18 Abelian-Simplicial-Group-with-Effective-Homotopy]
```

Their effective homotopies have been built in an automatic way and for each degree n they produce an ESG. Let us consider those in degree 1:

```
>(setf esg1 (funcall (efhmt1 ka1) 1))
[K30 Effective-Setoid-Group]
>(setf esg2 (funcall (efhmt1 kb1) 1))
[K31 Effective-Setoid-Group]
```

The elements of these ESGs are the 1-spheres of $K(A,1)$ and $K(B,1)$, which are canonically isomorphic to the elements of the groups A and B respectively. These elements are coded in Kenzo as *abstract simplices* given by a *degeneracy operator* (coded as a integer) and a *geometric* (non degenerate) simplex. In particular, the null element of both groups correspond to the abstract simplex $\eta_0\star$ coded with the pair $(1,\star)$ and the rest of elements of the groups have degeneracy operator equal to 0. Then, one can construct a morphism of effective setoid groups from `esg1` to `esg2` by defining the internal function, which inputs an abstract simplex of $K(A,1)$ and produces an abstract simplex of $K(B,1)$. In this case, the image of the base point $\eta_0\star$ in $K(A,1)$ is the base point in $K(B,1)$ (the same element) and the image of the two generators of A is the generator 1 in B:

```
>(progn
  (setf intr #'(lambda (absm)
                 (with-absm (dgop gmsm) absm
                   (if (= 0 dgop) (absm 0 (list (list
                      (+ (first (first gmsm)) (second (first gmsm))))))
                    absm))))
  (setf esg-mrph1 (build-esg-mrph :sorc esg1 :trgt esg2 :intr intr)))
[K32 Effective-Setoid-Group-Morphism]
```

The matrices `mtrx0` and `mtrx1` are computed automatically by Kenzo and correspond to the ones defined directly for the morphism of finitely generated abelian groups constructed in Subection 5.1.

```
>(mtrx0 esg-mrph1)
========== MATRIX 1 row(s) + 2 column(s) =====
R1=[C1=1][C2=1]
========== END-MATRIX
>(mtrx1 esg-mrph1)
========== MATRIX 1 row(s) + 1 column(s) =====
R1=[C1=3]
========== END-MATRIX
```

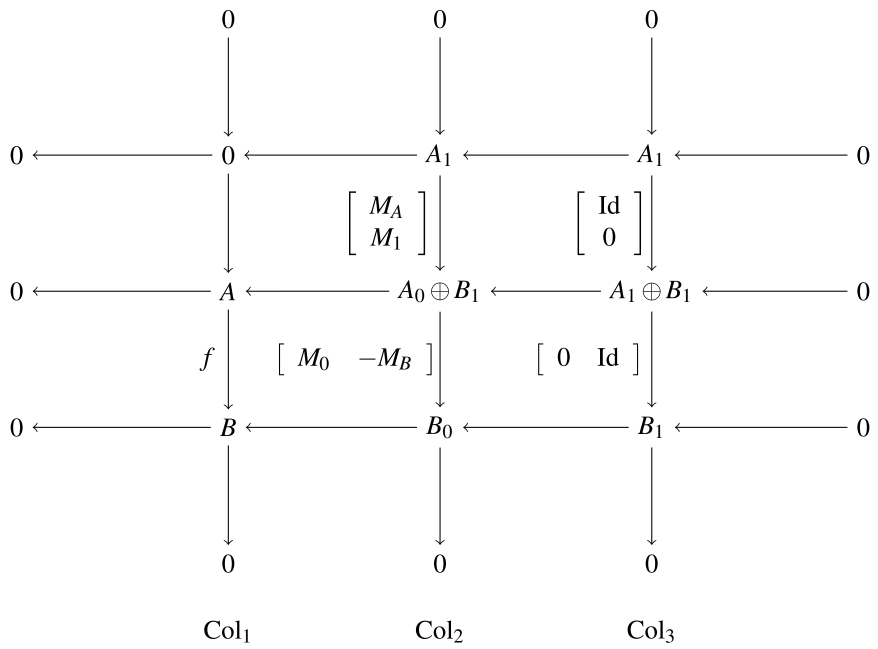


Figure 6: Chain complex for computing the kernel and the cokernel

5.3. Computing cokernels and kernels

Let $f : A \to B$ be a morphism between two ESG. The kernel $\operatorname{Ker} f \equiv K \subset A$ and the cokernel $\operatorname{Coker} f \equiv C = B/\operatorname{Im} f$ are also ESG. In order to implement them in Kenzo we need to define the different slots. The components `elms`, `cmps`, `nullel` and `inv` are induced by those of A and B respectively; it remains to define the slots `gr`, `f`, `g` and `h`. In particular, it is necessary to determine a free presentation of K and C by means of matrices in canonical form.

Let us begin by considering A and B two abelian groups of finite type, represented respectively by matrices $M_A : A_1 \to A_0$ and $M_B : B_1 \to B_0$. Let us suppose also that $f : A \to B$ is now a morphism of finitely generated abelian groups, given by matrices $M_0 : A_0 \to B_0$ and $M_1 : A_1 \to B_1$, and we want to determine free presentations for the finitely generated abelian groups $\operatorname{Ker} f \equiv K$ and $\operatorname{Coker} f \equiv C$.

We consider the diagram in Figure 6 where all the horizontal maps have not been included to simplify the diagram but are defined in the obvious way by using the different matrices. The reader can observe that each row in the diagram is exact, each column is a chain complex and the column on the right Col_3 is also exact. In this way, the homology groups of the two columns on the left Col_1 and Col_2 are canonically isomorphic.

Now, the kernel of f is given by the 1-st homology group of Col_1 and the cokernel of f is the 0-th homology group of Col_1. Thanks to the isomorphism previously explained, we can compute both groups by determining the homology of the second column Col_2, the chain complex:

$$0 \longrightarrow A_1 \overset{\left[\begin{smallmatrix} M_A \\ M_1 \end{smallmatrix}\right]}{\longrightarrow} A_0 \oplus B_1 \overset{\left[\begin{smallmatrix} M_0 & -M_B \end{smallmatrix}\right]}{\longrightarrow} B_0 \longrightarrow 0 \tag{16}$$

The cokernel of f is computed by considering the 1-degree differential matrix of this complex, $[M_0\ -M_B] \equiv D_1$, and then computing its canonical form by means of our new function `canonical-representation`. The new function `mrph-cokernel` inputs an object of the class `FNGN-ABGROUP-MRPH` $f : A \to B$ and outputs an instance C of the class `FNGN-ABGROUP` and two matrices $P : B_0 \to C_0$ and $I : C_0 \to B_0$ where C_0 is the target of the matrix of the returned group C. The matrix P corresponds to the projection $B \to C = \text{Coker}$; the matrix I is an auxiliary matrix expressing the relation between the generators of C and those of B. Let us observe that in general computing the 0-homology of the complex is not sufficient; the matrices P and I can also be necessary for example to compute the effective homotopy of a fibration. In particular, we will use them when determining the cokernel of a morphism of effective setoid groups.

The algorithm for computing the kernel of a morphism $f : A \to B$, which is given by the 1-homology group of the chain complex (16), is a bit more complicated and has several steps:

- We compute the Smith normal form of the matrix $[M_0\ -M_B] \equiv D_1$. This produces a new matrix D_1'.
- We apply a base change over the matrix $\begin{bmatrix} M_A \\ M_1 \end{bmatrix} \equiv D_2$ corresponding to the previous Smith normal form, obtaining a new matrix D_2' of the form $\begin{bmatrix} 0 \\ D_2'' \end{bmatrix}$ since $D_1' \circ D_2' = 0$.
- We compute the canonical form of D_2''.

The new function `mrph-kernel` inputs an object of the class `FNGN-ABGROUP-MRPH` $f : A \to B$ and outputs an instance K of the class `FNGN-ABGROUP` and two matrices $I : K_0 \to A_0$ and $P : A_0 \to K_0$ where K_0 is the target of the matrix of the returned group K. The matrix I corresponds in this case to the inclusion $K = \text{Ker} \to A$ and the matrix $P : A_0 \to K_0$ expresses the relation between the generators of A (being in the kernel) and those of K. As before, these matrices are necessary for computing the effective homotopy of a fibration.

Let us consider again the morphism defined at the end of Subsection 5.1. Its kernel and its cokernel are computed as follows.

```
> (divisors (first (mrph-kernel gr-mrph1)))
Component Z/3Z
Component Z
> (divisors (first (mrph-cokernel gr-mrph1)))
NIL
```

The matrices $m0 = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 2 & 2 \end{bmatrix}$ and $m1 = \begin{bmatrix} 0 & 2 \\ 1 & 2 \end{bmatrix}$ define a morphism between the groups $\mathbb{Z}/2 \oplus \mathbb{Z}/4 \oplus \mathbb{Z}$ and $\mathbb{Z}/2 \oplus \mathbb{Z}/4$. In this case the kernel and the cokernel are respectively the groups $\mathbb{Z}/2 \oplus \mathbb{Z}$ and $\mathbb{Z}/2$.

```
> (progn
  (setf a2 (build-fngn-abgroup :divs '(2 4 0)))
  (setf b2 (build-fngn-abgroup :divs '(2 4)))
  (setf m20 (build-mtrx :nrow 2 :ncol 3
                        :entries '(0 1 0 2 2 2)))
  (setf m21 (build-mtrx :nrow 2 :ncol 2
                        :entries '(0 2 1 2)))
  (setf gr-mrph2 (build-fngn-abgroup-mrph :sorc a2
       :trgt b2 :mtrx0 m20 :mtrx1 m21)))
```

```
[K37 Fngn-abgroup-mrph]
> (check-fngn-abgroup-mrph mrph2)
T
> (divisors (first (mrph-kernel gr-mrph2)))
Component Z/2Z
Component Z
> (divisors (first (mrph-cokernel gr-mrph2)))
Component Z/2Z
```

As said before, the second component in the results of the functions `mrph-kernel` and `mrph-cokernel` are matrices corresponding respectively to the inclusion $K \to A$ and the projection $B \to C$.

```
> (second (mrph-kernel gr-mrph2))
========== MATRIX 3 row(s) + 2 column(s) =====
R1=[C1=-2][C2=1]
R2=[C1=2]
R3=[C2=-1]
========== END-MATRIX
> (second (mrph-cokernel gr-mrph2))
========== MATRIX 1 row(s) + 2 column(s) =====
R1=[C1=-2][C2=1]
========== END-MATRIX
```

Let us suppose now that A and B are ESGs and $f : A \to B$ is a morphism of ESGs. Then the kernel and cokernel of f must be also elements of the class `EFFECTIVE-SETOID-GROUP`. As said before, the components `elms`, `cmps`, `nullel` and `inv` are induced by those of A and B respectively. The component `gr` is computed by using the functions `mrph-kernel` and `mrph-cokernel` defined for morphisms of finitely generated abelian groups. Finally, the components `f`, `g` and `h` are defined as composition of the components `f`, `g` and `h` of A and B and the auxiliary matrices returned by the functions `mrph-kernel` and `mrph-cokernel`.

For the example introduced at the end of Subection 5.2 we obtain the following results:

```
>(setf cok (mrph-cokernel esg-mrph1))
[K42 Effective-setoid-Group]
>(setf ker (mrph-kernel esg-mrph1))
[K45 Effective-setoid-Group]
>(divisors (gr cok))
NIL
>(divisors (gr ker))
Component Z/3Z
Component Z
```

5.4. Exact sequences

Let $0 \to A \to B \to C \to 0$ be a short exact sequence of groups. In Subsection 3.5 we have presented an algorithm to deduce a short resolution of B from short resolutions of A and C. Our algorithm generalises the construction of (Knapp, 2008, Lemma 4.23) and in particular can be applied when the group B is an *unknown* group, defined as the quotient of a setoid group. This algorithm has been implemented in Kenzo by means of a new function called `central-extension`.

Let us consider first the trivial extension of $\mathbb{Z}/3$ by $\mathbb{Z}/3$. It is built in Kenzo by the following statements.

```
> (progn
     (setf z3 (cyclic-group 3))
     (setf cocycle3 #'(lambda (c1 c2)
                       '(0)))
     (setf e (first (central-extension z3 z3 cocycle3))))
[K47 Fngn-abgroup]
```

The new object e is a finitely generated group, and its minimal resolution has been computed by Kenzo from the resolutions of A and C, in this case both equal to $\mathbb{Z}/3$. The resolution of e is stored in the slot mtrx of the group e:

```
> (mtrx e)
========== MATRIX 2 row(s) + 2 column(s) =====
R1=[C1=3]
R2=[C2=3]
========== END-MATRIX
```

The extension e is therefore in this case the group $\mathbb{Z}/3 \oplus \mathbb{Z}/3$:

```
> (divisors e)
Component Z/3Z
Component Z/3Z
```

As a second example, an extension of $\mathbb{Z}/6$ by $\mathbb{Z}/3$ is considered. In this example the cocycle $\gamma : \mathbb{Z}/6 \times \mathbb{Z}/6 \to \mathbb{Z}/3$ is not trivial and the result is the group $\mathbb{Z}/18$.

```
> (progn
   (setf z6 (cyclic-group 6))
   (setf cocycle6 #'(lambda (c1 c2)
                  (let* ((g1 (mod (first c1) 6))
                         (g2 (mod (first c2) 6))
                         (rslt
                          (if (< (+ g1 g2) 6) 0
                            (if (= 6 (+ g1 g2)) 1
                              (let ((g1-1 (1- g1)))
                 (+ (first (funcall cocycle g1-1 g2))
          (- (first (funcall cocycle 1 (+ g1 g2 -1)))
                  (first (funcall cocycle 1 g1-1)))))
                            ))))
                    (list rslt))))
   (setf e2 (first (central-extension z3 z6 cocycle6)))
   (divisors e2))
Component Z/18Z
```

It is worth remarking that in this case the initial block matrix obtained by applying the algorithm explained in Subsection 3.5 is given by

$$\left[\begin{array}{c|c} 3 & 1 \\ \hline 0 & 6 \end{array}\right] \tag{17}$$

whose canonical form is the final matrix [18].

```
> (mtrx e2)
========== MATRIX 1 row(s) + 1 column(s) =====
R1=[C1=18]
========== END-MATRIX
```

For examples of central extensions of ESGs see next section.

6. Outcomes

In this section we present some examples of calculation of the effective homotopy of the total space of a fibration from the effective homotopies of the fiber and the base space. These effective homotopies have been computed by implementing the different steps of our Algorithm 1; in particular, the algorithm for computing a resolution for the central extension of two ESGs presented in Subsection 3.5 has been used.

First of all, as a mechanism for constructing examples of Kan fibrations, a function is provided to build a Kan fibration from a twisted cartesian product; more concretely, the function `twop-kanfibration` inputs a twisting operator $\tau : B \to F$ (see (May, 1967, Ch.IV)) and outputs an object of the class `KAN-FIBRATION`.

Let us consider the Eilenberg-MacLane spaces $K(\mathbb{Z},2)$ and $K(\mathbb{Z},3)$ and a twisting operator $\tau : K(\mathbb{Z},3) \to K(\mathbb{Z},2)$ induced by the multiplication $\mathbb{Z} \stackrel{\times 3}{\to} \mathbb{Z}$. It is built in Kenzo with the following statements:

```
> (progn
    (setf x1 (k-zp 0 3))
    (setf chmlcl1 (chml-clss x1 3))
    (setf chmlcl (n-mrph 3 chmlcl1))
    (setf twop (z-whitehead x1 chmlcl))
    (setf fib1 (twop-kanfibration twop)))
[K186 Kan-fibration]
```

The result is an object of the class `KAN-FIBRATION`. As explained before, Eilenberg-MacLane spaces are automatically built in our new module of Kenzo as objects with effective homotopy. In this way, it is possible to compute the homotopy groups of the fiber and the base space of the fibration.

```
>(setf fiber (sorc (incl1 fib1))
  base (trgt (fibr1 fib1)))
[K74 Abelian-Simplicial-Group-with-Effective-Homotopy]
> (homotopy-group fiber 2)
Component Z
> (homotopy-group base 3)
Component Z
```

Let us remark that thanks to the effective homotopy, which is coded by means of functional programming, the *computation* of the homotopy groups of the base and the fiber in any dimension (which are null except for the previous degrees) is done in constant time and can be directly determined.

```
> (homotopy-group fiber 1000)
NIL
> (homotopy-group base 5000)
NIL
```

The total space of the fibration `fib1` is also built in Kenzo as an object of the class `KAN-WITH-EFHMT` but in this case the slot `efhmt` is in principle unbound.

```
> (setf tot (sorc (fibr1 fib1)))
[K178 Kan-Simplicial-Set-with-Effective-Homotopy]
```

When it is needed, the slot is automatically computed by means of the function `kfbr-tot-efhmt` which inputs an object of the class `KAN-FIBRATION` and a non-negative integer and computes the effective homotopy of the total space E in that degree. This function corresponds to the implementation of Algorithm 1 and in particular uses the construction for determining the ESG structure of a central extension explained in Subsection 3.6.

```
> (homotopy-group tot 1)
NIL
> (homotopy-group tot 2)
Component Z/3Z
> (homotopy-group tot 3)
NIL
```

We observe that the simplicial set `tot` can be seen as a model of the Eilenberg-MacLane space $K(\mathbb{Z}/3,2)$.

The space `tot` can be used now as the base of a new Kan fibration with new fiber $K(\mathbb{Z}/3,2)$ (in this case the space $K(\mathbb{Z}/3,2)$ is directly constructed by means of the function `k-zp`). We consider here the null twisting operator $\tau_2 : \texttt{tot} \to K(\mathbb{Z}/3,2)$.

```
>(progn
  (setf zero-twop2 (zero-twop tot (k-zp 3 2)))
  (setf fib2 (twop-kanfibration zero-twop2))
  (setf tot2 (sorc (fibr1 fib2))))
[K257 Kan-Simplicial-Set-with-Effective-Homotopy]
```

The homotopy groups of the new space are:

```
> (homotopy-group tot2 1)
NIL
> (homotopy-group tot2 2)
Component Z/3Z
Component Z/3Z
> (homotopy-group tot2 3)
NIL
```

The construction can be iterated, producing in this case a new space $\texttt{tot3} = K(\mathbb{Z}/6,1) \times \texttt{tot2}$. Let us observe that this space is not simply connected but π_1 is abelian and can be computed and used by the effective homotopy method.

```
> (progn
    (setf zero-twop3 (zero-twop tot2 (k-zp 6 1)))
    (setf fib3 (twop-kanfibration zero-twop3))
    (setf tot3 (sorc (fibr1 fib3))))
[K339 Kan-Simplicial-Set-with-Effective-Homotopy]
> (homotopy-group tot3 1)
Component Z/6Z
> (homotopy-group tot3 2)
Component Z/3Z
Component Z/3Z
> (homotopy-group tot3 3)
NIL
```

As a second example of calculation of our programs, we consider now the first steps of the Postnikov tower (May, 1967, Ch.V) for the 2-sphere S^2. They can be built in Kenzo by means of the following statements:

```
>(progn
    (setf p2 (k-zp 0 2))
    (setf ch4 (chml-clss p2 4))
    (setf f3 (z-whitehead p2 ch4))
    (setf pfib3(twop-kanfibration f3))
    (setf p3 (sorc (fibr1 pfib3))))
[K426 Kan-Simplicial-Set-with-Effective-Homotopy]
```

The result is a Kan simplicial set with effective homotopy, stored in the variable p3, which corresponds to the total space of a fibration with twisting operator $f3 : K(\mathbb{Z},2) \to K(\mathbb{Z},3)$. The effective homotopy of this space is directly built by Kenzo from the effective homotopies of the base and the fiber, so that the homotopy groups of p3 can be determined. We can observe that they correspond to $\pi_i(S^2)$ for $i \leq 3$.

```
>(homotopy-group p3 1)
NIL
>(homotopy-group p3 2)
Component Z
>(homotopy-group p3 3)
Component Z
```

The process can be iterated and the space p_j of the tower satisfies $\pi_i(p_j) \cong \pi_i(S^2)$ for $i \leq j$. The following Kenzo code shows the homotopy groups $\pi_i(S^2)$ for $i \leq 6$, obtaining in particular $\pi_6(S^2) \cong \mathbb{Z}/12$.

```
>(progn
  (setf ch5 (chml-clss p3 5))
  (setf f4 (zp-whitehead 2 p3 ch5))
  (setf pfib4 (twop-kanfibration f4))
  (setf p4 (sorc (fibr1 pfib4)))
  (setf ch6 (chml-clss p4 6))
  (setf f5 (zp-whitehead 2 p4 ch6))
  (setf pfib5 (twop-kanfibration f5))
  (setf p5 (sorc (fibr1 pfib5)))
  (setf ch7 (chml-clss p5 7))
  (setf f6 (zp-whitehead 12 p5 ch7))
  (setf pfib6 (twop-kanfibration f6))
  (setf p6 (sorc (fibr1 pfib6))))
[K1143 Kan-Simplicial-Set-with-Effective-Homotopy]
>(homotopy-group p6 2)
Component Z
>(homotopy-group p6 3)
Component Z
>(homotopy-group p6 4)
Component Z/2Z
>(homotopy-group p6 5)
Component Z/2Z
>(homotopy-group p6 6)
Component Z/12Z
```

7. Evaluation of other algorithms and programs

The first theoretical result about an algorithm computing the homotopy groups of a simply connected finite simplicial set is due to Edgar Brown (Brown (1957)), but it is well known this method is too complicated to be practically used. A better presentation of this method was given by Rolf Schön (Schön (1991)), also with a larger scope, but was never implemented.

Jean-Pierre Serre's thesis (Serre (1951)) was a revolution in Algebraic Topology, mainly due to the use of *spectral sequences*, the powerful tool invented by Jean Leray and Jean-Louis Koszul. Powerful, but yet this tool does not produce an *algorithm* computing the homology and homotopy groups that are looked for. The main goal of the Kenzo program (Dousson et al. (1999)), based on *effective homology*, was to circumvent this major obstacle, thanks to an intensive use of *functional programming*, now easy with the modern programming languages, Lisp, Haskell, OCaml, for example. The scope of the Kenzo program is large, allowing us for example to compute a few homotopy groups of complicated spaces, so far unknown. Up to our knowledge, no other computer program is capable of computing homotopy groups of arbitrary simply connected finite simplicial sets as Kenzo. The long experience of the algebraic topologists around the homotopy groups of spheres, see tipically Ravenel (2003) and Kochman (1990), shows how the Adams spectral sequence is powerful in this domain; nevertheless up to now without giving an algorithm computing these groups. Also, in the framework of Homotopy Type Theory, several people have proven known results about some homotopy groups of spheres, such as $\pi_3(S^2) = \mathbb{Z}$, $\pi_4(S^3) = \mathbb{Z}/2$, see for example Brunerie (2016). But the Homotopy Type Theory methods do not yet have produced an algorithm computing from scratch the first homotopy groups of arbitrary simply connected simplicial sets.

It was proved by Bousfield and Kan (Bousfield and Kan (1972)) that the Adams spectral sequence is below the so-called Bousfield-Kan spectral sequence, combinatorially more convenient, a spectral sequence which can be proved producing an actual *algorithm* for the desired homotopy groups, see Romero and Sergeraert (2017). The exact Serre sequence for fibrations then play an essential role inside this spectral sequence, and the present work is mainly designed to give a comfortable framework using this exact sequence in the difficult context of setoid simplicial sets.

8. Conclusions and further work

In this work we have presented a new module for the Kenzo system for the computation of the effective homotopy of the total space of a fibration from the effective homotopies of the fiber and the base space. The module consists of about 5000 lines of Common Lisp code containing the definition of new structures and functions. In particular, it includes the notions of setoid group and effective setoid group and a new module for working with finitely generated abelian groups and computing kernels, cokernels and central extensions which is then used to implement the long exact homotopy sequence of a fibration. The chosen representation for groups is that of a free presentation by means of a matrix in canonical form. As examples of calculations, some particular fibrations have been presented showing the computation of their homotopy groups.

The algorithm computing the effective homotopy of the total space of a fibration could be enhanced by obtaining similar programs producing the effective homotopy of the base space (respectively the fiber space) from the effective homotopies of the total space and the fiber (resp. the base). Furthermore, other constructions (loop spaces, classifying spaces, etc) in Algebraic Topology should be studied, as already done in the effective homology framework (see Rubio and Sergeraert (2006)). In other words, given a Kan simplicial set and a solution for its homotopical problem, algorithms should be designed and implemented computing the effective homotopy of its loop space, classifying space, etc. The case of the loop space has been considered in Romero and Sergeraert (2015), where a theoretical (not yet implemented) algorithm producing the effective homotopy of iterated loop spaces has been developed.

On the other hand, our algorithm computing the effective homotopy of a fibration can be considered an important ingredient in the development of a constructive version of the Bousfield-Kan spectral sequence associated with a simplicial set X (Bousfield and Kan (1972)). In Romero and Sergeraert (2017) we developed a theoretical algorithm producing the different components of the spectral sequence when the initial space X is an object with effective homology. As a further work, the implementation of this algorithm should be done in Common Lisp as a new module for the Kenzo system making use of the module for the effective homotopy of fibrations presented in this paper.

Acknowledgments

This work has been partially supported by Ministerio de Economía y Competitividad, Spain, project MTM2017-88804-P.

The authors want to thank the work of different colleagues for their careful reading and their useful constructive recommendations and specially a referee who suggested us to make explicit the relationship of our work with dependent type theory.

References

Barakat, M., Bremer, B., 2008. Higher extension modules and the Yoneda product, preprint. https://arxiv.org/abs/0802.3179.

Bousfield, A., Kan, D., 1972. Homotopy Limits, Completions and Localizations. Lecture Notes in Mathematics vol. 304. Springer-Verlag.

Bove, A., Dybjer, P., Norell, U., 2009. A brief overview of Agda - A functional language with dependent types. In: 22nd International Conference on Theorem Proving in Higher Order Logics (TPHOLs'09), Lecture Notes in Computer Science 5674. pp. 73–78.

Brown, E., 1957. Finite computability of postnikov complexes. Annals of Mathematics 65 (1), 1–20.

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

Brunerie, G., 2016. On the homotopy groups of spheres in homotopy type theory, preprint. https://arxiv.org/abs/1606.05916.

Cohen, C., Coquand, T., Huber, S., Mörtberg, A., 2016. Cubical Type Theory: a constructive interpretation of the univalence axiom, http://arxiv.org/abs/1611.02108.

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

Kaczynski, T., Mischaikow, K., Mrozek, M., 2004. Computational Homology. Vol. 157 of Applied Mathematical Sciences. Springer.

Knapp, A. W., 2008. Advanced algebra. Birkhäuser.

Kochman, S. O., 1990. Stable homotopy groups of spheres. A computer-assisted approach. Vol. 1423 of Lecture Notes in Mathematics. Springer-Verlag.

May, J. P., 1967. Simplicial objects in Algebraic Topology. Van Nostrand Mathematical Studies. University of Chicago Press.

Ravenel, D. C., 2003. Complex Cobordism and stable homotopy groups of spheres. AMS Chelsea, Second Edition.

Romero, A., Rubio, J., 2013. Homotopy groups of suspended classifying spaces: an experimental approach. Mathematics of Computation 82, 2237–2244.

Romero, A., Sergeraert, F., 2012. Effective homotopy of fibrations. Applicable Algebra in Engineering, Communication and Computing 23, 85–100.

Romero, A., Sergeraert, F., 2015. A combinatorial tool for computing the effective homotopy of iterated loop spaces. Discrete & Computational Geometry 53 (1), 1–15.

Romero, A., Sergeraert, F., 2017. A Bousfield-Kan algorithm for computing the effective homotopy of a space. Foundations of Computational Mathematics 17, 1335–1366.

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

Rubio, J., Sergeraert, F., 2006. Constructive Homological Algebra and Applications, preprint. http://arxiv.org/abs/1208.3816.

Schön, R., 1991. Effective algebraic topology. Mem. Amer. Math. Soc. 451.

Sergeraert, F., 1994. The computability problem in Algebraic Topology. Advances in Mathematics 104 (1), 1–29.

Sergeraert, F., 2009. Effective exact couples, preprint. `http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/Exact-Couples-2-2.pdf`.

Serre, J., 1953. Groupes d'homotopie et classes de groupes abéliens. Annals of Mathematics 58 (2), 258–294.

Serre, J.-P., 1951. Homologie singulière des espaces fibrés. applications. Annals of Mathematics, 425–505.

Univalent Foundations Program, T., 2013. Homotopy Type Theory: Univalent Foundations of Mathematics. `https://homotopytypetheory.org/book`, Institute for Advanced Study.