How the Kenzo program works.

Francis Sergeraert[*]

September 15, 2011

In honour of Guy Wallet[1]

Abstract

Yet another introduction to Constructive Homological Algebra*. Experience shows this domain of Mathematics is rarely well understood. It requires a reasonable understanding of* two *subjects, functional programming and elementary homological algebra. These subjects are not difficult but they are relatively far from each other. This* n*-th introduction tries to achieve the following goal: just a few simple claims in functional programming and in homological algebra are stated and have to be admitted to understand our general organization. A simple Kenzo calculation is used to proceed, hoping this text could be useful for* both *categories of mathematicians: the computer scientists and the topologists.*

1 Introduction

It is a little bewildering for a topologist to be invited at a meeting entitled "Des Nombres et des Mondes" (About Numbers and Worlds); how would it be possible to connect this ambitious title and the relatively esoteric world of topology? But after all, this can lead quickly to the heart of our subject.

The Euclid algorithm is well known. An *automatic* process can be applied to compute, given two positive numbers a and b, their gcd. It is a beginner exercise often given in programming courses. A simple program can, given the *input* $(30, 45)$, return the *output* 15, the gcd of 30 and 45; the same program can be used for every input. In modern language, the *specification* of the Euclid algorithm E is:

$$\mathbb{N}_*^2 \ni (a, b) \overset{E}{\longmapsto} c = \gcd(a, b) \tag{1}$$

[*] `Francis.Sergeraert@ujf-grenoble.fr`

[1] This text is a variant of the oral talk given at the nice meeting "*Des Nombres et des Mondes*" organized at La Rochelle in June 2011, in honour of Guy Wallet. I use this opportunity to express my gratitude to Guy. Working at Poitiers in 72-82 in his neighbourhood was one of the best experiences of my scientific life, in particular when he convinced me of the interest of the *non-standard analysis*. With a consequence: this made obvious that the usual mathematical starting axioms are not necessarily fixed once for all; certainly this played a role in my permanent interest now for *constructive* mathematics. To support this idea, let us observe the recent studies around *constructive non-standard analysis*[13, 14].

Euclid can be here thought of as a designer of algorithm, long before the invention of computers.

A more recent example along the same line. Jean-Pierre Serre obtained in 1954 a Fields Medal; he was rewarded in particular for this result:

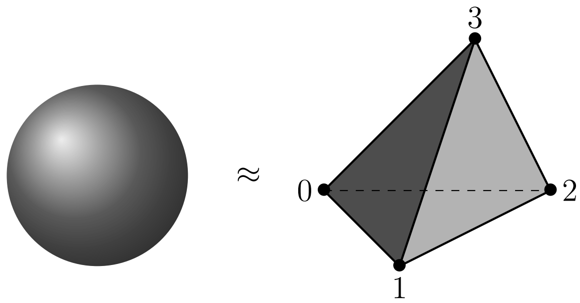
$$(7\ ((1\ 2\ 3)\ (0\ 2\ 3)\ (0\ 1\ 3)\ (0\ 1\ 2))) \stackrel{S}{\longmapsto} (2) \tag{2}$$

What is the nature of the output (2) with respect to the complicated list given as input?

Serre obtained several homotopy groups of spheres, in particular $\pi_7(S^2) = \mathbb{Z}/2$. This means the functional space $\mathcal{C}(S^7, S^2)$ made of the continuous maps between the 7-sphere and the 2-sphere has two connected components. Also this set has a natural group structure which therefore necessarily is isomorphic to $\mathbb{Z}/2$.

In the correspondance $\stackrel{S}{\longmapsto}$ above, the first 7 codes the index 7 of the functor π_7. Any abelian group G of finite type admits a canonical expression with respect to its *divisors* $G = \mathbb{Z}/d_1 \oplus \cdots \oplus \mathbb{Z}/d_k$ where every divisor d_i divides the next one, the last divisors being possibly null. For example the canonical writing of $\mathbb{Z}^2 \oplus \mathbb{Z}/4 \oplus \mathbb{Z}/6$ is $\mathbb{Z}/2 \oplus \mathbb{Z}/12 \oplus \mathbb{Z}/0 \oplus \mathbb{Z}/0$. Such a group can be coded by an integer list, (2 12 0 0) for our example. So that the final (2) in the correspondance $\stackrel{S}{\longmapsto}$ represents the value $\mathbb{Z}/2$ of the Serre result.

The reader guesses the list of lists ((1 2 3) (0 2 3) (0 1 3) (0 1 2)) should represent the 2-sphere S^2, the last missing ingredient of the formula $\pi_7(S^2) = \mathbb{Z}/2$. Why? The standard 2-sphere is homeomorphic to the boundary of a tetrahedron spanned by the vertices $0..3$, and our list of lists is the list of the maximal simplices of this boundary. The 2-sphere is combinatorially represented as a *simplicial complex.*

$S^2 =$ 

(3)

Many topological spaces have analogous *combinatorial* descriptions, and the reader could believe Serre produced an algorithm:

$$\mathbb{N} \times \underline{\mathrm{Top}}_1 \ni (n, X) \stackrel{S}{\longmapsto} \pi_n(X) \in \underline{\mathrm{Ab}}$$

where $\underline{\mathrm{Top}}_1$ is an appropriate[2] category of combinatorial topological spaces and $\underline{\mathrm{Ab}}$ the category of abelian groups. Not at all.

The methods designed by Serre have a large scope, but fail to be such an *algorithm.* For example, in the particular case of $\pi_6(S^2)$, Serre was able to determine

[2]In particular X must be *simply connected*, otherwise the problem becomes totally different.

this group has 12 elements, but did not succeed in choosing between $\mathbb{Z}/12$ and $\mathbb{Z}/2 \oplus \mathbb{Z}/6$; the point is that this ambiguity *could not* be solved by some possible *computation*; no *algorithm* could then be deduced of Serre's beautiful work to decide between both known possible values. Barratt and Paechter [1] proved a little later in fact this group contains an element of order 4 and therefore finally $\pi_6(S^2) = \mathbb{Z}/12$, but this was a consequence of a *new* theoretical study on another subject, the homotopy groups of Stiefel manifolds.

This little story is typical of the general style of "standard" algebraic topology when the *value* of some invariant, typically a homology or a homotopy group, is looked for. The countless exact and spectral sequences which are available allow a topologist to determine many invariants of this sort, but they are rarely *algorithms* with a *general* scope. Most often they did not solve the *extension* problems; for example, using his famous spectral sequence, Serre produced an exact sequence:

$$0 \leftarrow \mathbb{Z}/6 \leftarrow \pi_6(S^2) \leftarrow \mathbb{Z}/2 \leftarrow 0 \tag{4}$$

without being able to solve the extension problem; see [21, Section 3.3.2] for a detailed description of the process generating this obstacle. It is often believed the Bockstein spectral sequence [3] can solve these extension problems, but the problem is then in fact transferred into another one, determining the higher differentials of this spectral sequence. Most often no known algorithm can give them; an example is given here later.

Yet an algorithm was quickly given by Edgar Brown [4] to compute the homotopy groups of the finite simply connected simplicial complexes. Based on a combinatorial and very interesting study of the *Postnikov towers*, Edgar Brown also honestly warned the reader:

> *It must be emphasized that although the procedures developed for solving these problems are finite, they are much too complicated to be considered practical.*

an appreciation which unfortunately remains valid today, even with the help of the most powerful computers now available. Note also Edgar Brown did not use spectral sequences: he knew they cannot be used to produce *algorithms*.

In the eighties, several methods were designed to organize basic homological algebra in such a way on the contrary the standard exact and spectral sequences become algorithmical tools. The *constructive* point of view led this author and his collaborators to a simple solution, the only one so far used to produce *concrete* computer programs. A simple computer example, the computation of the homology group $H_5(\Omega^3 S^3 P^2\mathbb{R})$ is used here to explain the general organization of this solution, logically called *constructive homological algebra*. See for other presentations [22, 18, 19, 20, 2], where the other available methods, all of them very interesting, are also referenced and commented.

2 Constructive Homological Algebra.

We recall in this section the general organization of *Constructive Homological Algebra*. With respect to previous texts, consider "Constructive" and "Effective" are perfect synonymous words. From a theoretical point of view, the matter at issue is the following: many "methods" in Algebraic Topology are presented as methods of *computation*, with a rather ambiguous status: because they often involve infinite objects, in fact they are not *algorithms*. The so-called *Constructive* Homological Algebra overcomes this essential obstacle by a simple strict organization needing other tools.

Two main tools are used. The first one is purely mathematical; the *Basic Perturbation Lemma* [5] is an elementary process often allowing its user to appropriately connect a chain complex[3] *defining* homology groups to another chain complex *computing* the same homology groups; most often the first one is not of finite type, or has a giant size forbidding its implementation on a concrete computer, even a very powerful one.

The second tool comes from Computer Science: *Functional Programming* does finally allow a user to implement these infinite or giant chain complexes on a computer, but in a *functional way*. No algorithm can compute the homology groups of a chain complex so implemented. Fortunately, the basic perturbation lemma is available and can often be used to connect this strange type of chain complex with an ordinary chain complex of finite type where on the contrary the homology groups can be elementarily computed. Functional programming is important in computer science, it goes back to Church and his brilliant invention, the λ-*calculus* [7]; a difficult problem for the scope of identifiers is met here for the practical programming languages, elegantly solved now thanks to the wonderful notion of *lexical closure*. For instructive explanations, see [23], in particular for the essential obstacle when trying to use such a technique in Java or C_{++}.

Using these tools, the general *finiteness* results obtained long ago by Serre [15] are easily transformed into *computability* results. The Kenzo program [8] proves this is not only a theoretical result: the algorithms so defined can be concretely implemented and used.

For example, if you are interested[4] by the homology group $H_5(\Omega^3\Sigma^3P^2\mathbb{R})$, using the Kenzo program, you could proceed as follows. First, the real projective plane $P^2\mathbb{R}$ is constructed:

---

[3]A *chain complex* is an algebraic object made of abelian groups called *chain groups* connected by *differentials* satisfying simple properties; a chain complex is an intermediate algebraic object between a topological space and the homology groups of this space. The chain complex *associated* to some object *defines* the homology groups of this object.

[4]We like this example, because in principle the result should be deduced from [12]. Two excellent topologists were questionned about this group; the first one successively proposed two different results, both incorrect; the second one observed he was not able to apply the Bockstein spectral sequence to this particular case, necessary to obtain the final result. At least they *tried* to compute this group, sincere thanks!

```
> (setf P2R (r-proj-space 3)) ✠
[K1 Simplicial-Set]
```

The Lisp prompt is the greater character '`>`' and the user then *enters* a Lisp *statement* to be *evaluated*, here the statement `(setf P2R (r-proj-space 3))`, 3 because it is the first dimension without any simplex, a more convenient convention for the general case. On this display, the end of the Lisp statement is marked by the maltese character '✠', in fact not visible on the user's screen; the end of the Lisp statement is automatically detected by the Lisp interpreter, which then *evaluates* the given statement and *returns* the result of the evaluation, here the Kenzo object #1 (`K1`), which happens to be a simplicial set[5]. Only a simple external reference to this object is displayed, the internal object, a package of rather sophisticated algorithms, cannot be properly displayed. Also the simplicial set so constructed is *assigned* to the symbol `P2R` for future reference.

Then the *suspension*[6] *functor* Σ is applied three times to our simplicial set `P2R`. The result is again a simplicial set assigned to the symbol `S3P2R`:

```
CL-USER(2): (setf S3P2R (suspension P2R 3)) ✠
[K16 Simplicial-Set]
```

The result is the Kenzo object #16, a simplicial set; the Kenzo program had to construct several other intermediate invisible objects numbered from 2 to 15. The *loop-space*[7] *functor* is in turn applied three times:

```
CL-USER(3): (setf O3S3P2R (loop-space S3P2R 3)) ✠
[K45 Simplicial-Group]
```

The Kan model for the loop space functor, see [11], constructs a simplicial *group*. Finally the desired homology group can be calculated:

```
CL-USER(4): (homology O3S3P2R 5) ✠
Computing boundary-matrix in dimension 5.
Rank of the source-module : 23.
Computing boundary-matrix in dimension 6.
Rank of the source-module : 53.
Homology in dimension 5 :
Component Z/2Z
Component Z/2Z
Component Z/2Z
Component Z/2Z
Component Z/2Z
```

[5]Roughly analogous to a simplicial complex, but with incidence relations between simplices more sophisticated.

[6]If X is a topological space with a *base point* $*$, the suspension ΣX of X is the quotient $(X \times I)/((X \times 0) \cup (X \times 1) \cup (* \times I))$; for example $\Sigma S^n = S^{n+1}$.

[7]If X is a topological space with a base point $*$, the loop space ΩX is the functional space of the continuous functions $\gamma : I \to X$ satisfying $\gamma(0) = \gamma(1) = *$, with the usual topology.

which implies $H_5(\Omega^3\Sigma^3 P^2\mathbb{R}) = (\mathbb{Z}/2)^5$.

3 Object with Effective Homology.

How the Kenzo program successfully processes the computation of the previous section? The Kenzo program implements the key notions of *reduction*, of (homological) *equivalence* and of *object with effective homology*. The definitions are recalled here.

Definition 1 — A *reduction* $\rho = (f, g, h) : \widehat{C}_* \Rightarrow\!\!\!\!\Rightarrow C_*$ is a diagram:

$$\rho = \boxed{h \circlearrowright \widehat{C}_* \underset{f}{\overset{g}{\leftrightarrows}} C_*} \tag{5}$$

where:

- The nodes $\widehat{C}_*$ and C_* are chain complexes, the first one $\widehat{C}_*$ being the *big* one, the second one C_* the *small* one;
- The arrows f and g are two chain complex morphisms;
- The self-arrow h is a homotopy operator (degree +1);
- The following relations are satisfied:

$$\begin{aligned} fg &= \mathrm{id}_{C_*} \\ gf + dh + hd &= \mathrm{id}_{\widehat{C}_*} \\ fh &= 0 \\ hg &= 0 \\ hh &= 0 \end{aligned}$$

This reduction describes the big chain complex $\widehat{C}_*$ as the direct sum of the small one $C_* \cong g(C_*)$ and an *acyclic* complement $\ker(f)$. This implies the homological natures of both complexes C_* and $\widehat{C}_*$ are canonically isomorphic.

Definition 2 — An *equivalence* $\varepsilon : C_* \Leftarrow\!\!\!\!\Leftarrow\!\!\Rightarrow\!\!\!\!\Rightarrow C'_*$ between two chain complexes is an extra chain complex $\widehat{C}_*$ and a pair of reductions $\rho = (f, g, h) : \widehat{C}_* \Rightarrow\!\!\!\!\Rightarrow C_*$ and $\rho' = (f', g', h') : \widehat{C}_* \Rightarrow\!\!\!\!\Rightarrow C'_*$.

Definition 3 — An object *with effective homology* is a 4-tuple $(X, C_*X, EC_*, \varepsilon)$ where:

- X is some object studied from a homological point of view, thanks to the canonical chain complex C_*X associated to it in the current context: simplicial homology, homology of groups, Hochschild homology, cyclic homology... This chain complex *defines* the homology groups of X with respect to some homological theory, but most often, the *computation* of these groups is out of scope if only this information is available.

- EC_* is a chain complex of finite type whose homology is therefore elementarily computable (E for effective);
- ε is an equivalence $\varepsilon : C_*X \Lleftarrow\!\!\Rrightarrow EC_*$

The equivalence ε defines in particular an ordinary homology equivalence between C_*X and EC_*; a canonical isomorphism is defined $H_*X := H_*C_*X \cong H_*EC_*$: the homology groups of X are finally *computable*, thanks to the extra information given by EC_* and ε.

Much more importantly, this data type is *stable*, which is explained now.

Meta-Theorem 4 — *Let* χ *be a constructor:*

$$\chi : (X_1, \ldots, X_n) \mapsto X$$

producing an object X *from various objects* $X_1, \ldots, X_n$. *Then, under appropriate conditions, an* algorithm χ^{EH}*:*

$$\chi^{EH} : (X_1^{EH}, \ldots, X_n^{EH}) \mapsto X^{EH}$$

can be written down. This algorithm χ^{EH} *is called a* version with effective homology *of the constructor* χ.

Each input object X_i^{EH} is assumed to be an object with effective homology $X_i^{EH} = (X_i, C_*X_i, EC_{i,*}, \varepsilon_i)$ and the algorithm χ^{EH} produces an object $X^{EH} = (X, C_*X, EC_*, \varepsilon)$, also an object with effective homology.

So that, if interested in the homology groups of X, you can use the effective chain complex EC_* to elementarily compute them. More important, if you intend to use the output object X as input for another constructor χ', the same process can be applied in turn to χ'^{EH} and X^{EH}; in particular, *iterations* become easy.

4 Object with Effective Homology in Kenzo.

The Kenzo program is written in the programming language *Common-Lisp*. Let us try to understand how the theoretical notions quickly explained in the previous section look like in the Kenzo environment. We reexamine the objects produced by the short Kenzo session of Section 2.

The final space `O3S3P2R` $= \Omega^3 S^3 P^2\mathbb{R}$ was the Kenzo object #45, a simplicial group:

```
> O3S3P2R ✠
[K45 Simplicial-Group]
```

The Lisp statement asking for $H_5(\Omega^3 S^3 P^2\mathbb{R})$ generated a silent work in the Kenzo program, producing a version *with effective homology* of our loop space. We can ask for this effective homology:

```
CL-USER(9): (efhm O3S3P2R) ✠
[K431 Equivalence K45 <= K421 => K417]
```

Understand that `efhm` = <u>E</u>ffective <u>H</u>o<u>m</u>ology. The equivalence, the Kenzo object #431, is an equivalence between `K45` and `K417` through the auxiliary chain complex `K421`. Initially, the object `K45` was our space `O3S3P2R`, but thanks to the powerful *object oriented programming* available in Common Lisp, a simplicial set in general, a simplicial group in particular, implicitly contains[8] the associated chain complex defining its homology groups; this is the reason why the simplicial set and the associated chain complex are located through the same reference.

With respect to Definition 3, the components $(X_6, C_*X_6, EC_{6,*}, \varepsilon_6)$ of the version with effective homology of our space `O3S3P2R` are:

$$(X_6, C_*X_6, EC_{6,*}, \varepsilon_6) = (\texttt{K45}, \texttt{K45}, \texttt{K417}, \texttt{K431}) \tag{6}$$

for, as explained above, the object `K45` contains simultaneously the simplicial group and the associated chain complex.

Why this notation X_6? Our program successively constructed:

$$\begin{gathered} X_0 = P^2\mathbb{R} \quad X_1 = \Sigma P^2\mathbb{R} \quad X_2 = \Sigma^2 P^2\mathbb{R} \quad X_3 = \Sigma^3 P^2\mathbb{R} \\ X_4 = \Omega\Sigma^3 P^2\mathbb{R} \quad X_5 = \Omega^2\Sigma^3 P^2\mathbb{R} \quad X_6 = \Omega^3\Sigma^3 P^2\mathbb{R} \end{gathered} \tag{7}$$

each one as an object with effective homology. Let us examine for example the object X_5 and its effective homology:

```
> (setf O2S3P2R (loop-space S3P2R 2)) ✠
[K33 Simplicial-Group]
CL-USER(6): (efhm O2S3P2R) ✠
[K383 Equivalence K33 <= K373 => K369]
```

Kenzo knows how to locate this object already constructed, it is the object #33 and you understand its effective homology is:

$$(X_5, C_*X_5, EC_{5,*}, \varepsilon_5) = (\texttt{K33}, \texttt{K33}, \texttt{K369}, \texttt{K383}) \tag{8}$$

The loop space functor Ω had been applied to X_5 = `O2S3P2R` producing $X_6 = \Omega X_5$ = `O3S3P2R` and, more important, because of Meta-Theorem 4 applied to the constructor Ω, a version with effective homology Ω^{EH} of this functor has been used to produce a version with effective homology of X_6:

$$\begin{gathered} (X_5, C_*X_5, EC_{5,*}, \varepsilon_5) \longmapsto (X_6, C_*X_6, EC_{6,*}, \varepsilon_6) \\ (\texttt{K33}, \texttt{K33}, \texttt{K369}, \texttt{K383}) \overset{\Omega^{EH}}{\longmapsto} (\texttt{K45}, \texttt{K45}, \texttt{K417}, \texttt{K431}) \end{gathered} \tag{9}$$

[8] More precisely, following the right point of view of Eilenberg and MacLane [9], a chain complex can sometimes be endowed with an additional structure defining a simplicial set. In other words, Eilenberg and MacLane had already understood in 1950 what *object oriented programming* is.

The chain complexes `K33` $= C_*X_5$ and `K45` $= C_*X_6$ are “terribly” not of finite type: in `K45`, a chain group is the free $\mathbb{Z}$-module generated by the *elements* of the free *non-commutative* group generated by the *elements* of another free *non-commutative* group in turn generated by the *elements* of another free *non-commutative* group of finite type, text carefully checked by the author. Kan in his landmark article [11] probably did not imagine his hyper-giant theoretical model for iterated loop spaces could some day be concretely and usefully installed on a computer; no problem to do it with the functional languages now available.

Unfortunately, an avatar of the incompleteness theorems of Gödel, Turing, Church and Post excludes some algorithm can compute the homology groups for example of `K45` *alone* available in your computer environment.

On the contrary the chain complexes `K369` and `K417` are of finite type and their homology groups can be elementarily computed. For example, because of the connection `K431` between `K45` and `K417`, the desired homology group $H_5(\Omega^3\Sigma_3 P^2\mathbb{R}) = H_5(\texttt{K45})$ is canonically isomorphic to $H_5(\texttt{K417})$; and the chain complex `K417` is made of free $\mathbb{Z}$-modules of finite type of rank 10, 23 and 53 in degrees 4, 5 and 6. The Smith reduction can be applied to the 10×23 and 23×53 boundary matrices, the homology group H_5 being a direct consequence.

To explain the heart of the constructive process, let us repeat the critical construction step:

$$(\texttt{K33}, \texttt{K33}, \texttt{K369}, \texttt{K383}) \overset{\Omega^{EH}}{\longmapsto} (\texttt{K45}, \texttt{K45}, \texttt{K417}, \texttt{K431}) \tag{10}$$

In this diagram, you can deduce `K45` from `K33`, easy in functional programming, but you definitively *cannot deduce* `K417` *from* `K45` and *you cannot deduce* `K417` *from* `K369` either. The *whole* information given in the object $(\texttt{K33}, \texttt{K33}, \texttt{K369}, \texttt{K383})$ is necessary to be able to construct `K417`. The same for `K431`, necessary also if you intend to use in turn $X_6 = \Omega^3\Sigma^3 P^2 R$ for another construction.

In the next diagram, understand each arrow means the source of the arrow is *in particular* necessary to construct its target.

$$\begin{tikzcd}
(X_5 \arrow[r] \arrow[d, mapsto, "\Omega"'] \arrow[drr] \arrow[drrr] & C_*X_5 & EC_{*,5} \arrow[d] \arrow[dr] & \varepsilon_5) \arrow[dl] \arrow[d] & = & X_5^{EH} \arrow[d, mapsto, "\Omega^{EH}"] \\
(X_6 \arrow[r] & C_*X_6 & EC_{*,6} & \varepsilon_6) & = & X_6^{EH}
\end{tikzcd} \tag{11}$$

Note also some redundancy, just to help reading and understanding: the simplicial set X_i defines the associated chain complex C_*X_i; the equivalence ε_i contains in particular both equivalent chain complexes C_*X_i and $EC_{*,i}$ connected by this equivalence.

Analogous comments are valid for the connections between X_0^{EH} and X_1^{EH}, X_1^{EH} and X_2^{EH}, …, X_4^{EH} and X_5^{EH}.

5 The starting point.

5.1 Trivial starting point.

The previous section does not explain the nature of the starting point: how the Kenzo program determines the necessary version with effective homology $X_0^{EH} = (X_0, C_*X_0, EH_{*,0}, \varepsilon_0)$ of the initial space, the real projective plane $X_0 = P^2$?

The reader understands the very nature of constructive algebraic topology is recursive, but any recursion must have a starting point. A *particular* study must always be done for the starting point.

Most often, the starting point takes place in a particular situation where the context is simple or even trivial. It is so in the unique example so far considered. The standard simplicial model for the real projective plane $P^2\mathbb{R}$ is very simple: only one vertex s_0, one edge s_1, both ends of which are identified to s_0, producing a circle, and one "triangle" s_2 attached to s_1 by an attaching map of degree 2: $\partial s_2 \to s_1$. The associated chain complex is the following:

$$0 \leftarrow (C_0 = \mathbb{Z}) \xleftarrow{0} (C_1 = \mathbb{Z}) \xleftarrow{\times 2} (C_2 = \mathbb{Z}) \leftarrow 0 \tag{12}$$

For such a very simple chain complex of finite type, the *trivial* effective homology can be chosen: $X_0^{EH} = (X_0, C_*X_0, (EC_{*,0} = C_*X_0), (\varepsilon_0 = \mathrm{id}))$. A trivial reduction $\rho = (f, g, h) : C_* \Rrightarrow C_*$ is one where both chain complexes are the same C_* and $f = g = \mathrm{id}_{C_*}$, and $h = 0$; see Definition 1. A trivial equivalence is made of two trivial reductions $C_* \Lleftarrow C_* \Rrightarrow C_*$.

The Kenzo program knows our initial space X_0 can be so processed:

```
> (efhm P2R) ✠
[K9 Equivalence K1 <= K1 => K1]
```

We can question Kenzo how this equivalence was constructed:

```
> (orgn (efhm P2R))
(TRIVIAL-EQUIVALENCE [K1 Simplicial-Set])
```

Kenzo answers the origin (`orgn`) of our equivalence is simply the trivial equivalence of our simplicial set $P^2\mathbb{R}$, more precisely of the underlying chain complex.

This situation of a trivial starting point is very frequent.

5.2 Non-trivial starting point.

In other cases, a real preliminary and particular study must be undertaken to construct the starting point. A typical example of this sort is for the effective homology of an Eilenberg-MacLane space $K(\pi, n)$. Here the group π is an abelian group of finite type and n is a positive integer. It is a relatively esoteric object absolutely essential in the computation of homotopy groups of a space X via its Postnikov tower, see [4].

The recursive process uses here the *classifying space* constructor usually denoted by B for base space: it is the base space of some universal fibration. The recursive process is defined by the formula $K(\pi, n) := B(K(\pi, n-1))$ and the starting point is $K(\pi, 1)$. The constructor B does admit a version with effective homology B^{EH} and there remains to find the effective homology of $K(\pi, 1)$.

This was implicitly done by Eilenberg and MacLane in Sections 14 and 15 of [10]. The previous paper [9] of the same authors is the first example of a text in fact devoted to a particular case of constructive algebraic topology, without using its terminology. The authors describe there in detail an explicit process to connect the chain complex $C_*K(\pi, n)$ to a chain complex of finite type which they call $A(\pi, n)$, claiming the last one is more *perspicuous*, see the second paragraph of the Introduction.

No mathematical definition for the last adjective. In fact the essential difference is the following: the chain complex $A(\pi, n)$ is of finite type, so that the computability of its homology groups is obvious, while this is often false for $C_*K(\pi, n)$. In the second paper [10] of the series, the authors start to exploit this property and obtain an impressive collection of concrete results. The final result was a simple algorithm obtained by Cartan [6] quickly giving the homology groups $H_*K(\pi, n) = H_*A(\pi, n)$.

These homology groups of course are interesting, but *constructive* algebraic topology explains that if you do not have in your environment an equivalence connecting $K(\pi, n)$ and $A(\pi, n)$, you are unable to use these groups to compute for example homotopy groups in the *general* case. You understand in a sense the first paper [9] is finally more important than the next ones, for this paper explicitly describes such a connection. The most *perspicuous* object is neither $K(\pi, n)$ nor $A(\pi, n)$, it is the equivalence $\varepsilon : C_*(K(\pi, n)) \Longleftarrow\!\!\!\Longrightarrow A_*(\pi, n)$.

6 Conclusion.

The example used in the present paper around loop spaces is much more difficult than the example of the effective homology of $K(\pi, n)$, also covered by the Kenzo program. A loop space has an inevitable non-commutative nature, while on the contrary the Eilenberg-MacLane spaces $K(\pi, n)$ are commutative. The equivalence[9] $C_*(\Omega^n(X)) \Longleftarrow\!\!\!\Longrightarrow \mathrm{Cobar}^n(X)$ was obtained for the first time in 1988 in the Spanish thesis of Julio Rubio [16], using a terrible computational process. The basic perturbation lemma allowed the same author in a 1991 French thesis [17] to

[9]The Cobar construction is dual of the $A(\pi, n)$ construction of Eilenberg-MacLane, nowadays called the Bar construction. This apparent "duality" is terribly misleading: a simple direct recursive process defines $\mathrm{Bar}^{n-1}(X) \mapsto \mathrm{Bar}^n(X)$, while it is impossible to have such a correspondance to define $\mathrm{Cobar}^n(X)$; the last one can be obtained only through the *whole* information contained in $(\Omega^{n-1}(X))^{EH}$, see Diagram 11 where you replace $EC_{*,6}$ by $\mathrm{Cobar}^n(X)$ and the line X_5^{EH} by $(\Omega^{n-1}(X))^{EH}$, in particular $EC_{*,5}$ by $\mathrm{Cobar}^{n-1}(X)$.

obtain a much more convenient version of the same result. This version was immediately concretely implemented on computers, allowing us to obtain homology groups then otherwise unreachable. It so happens the same groups twenty years later remain otherwise unreachable.

It is often mentionned there does not exist any reason to be interested by the *numerical* values of homology groups of loop spaces. Sure, the main interest in these calculations in fact is in the theoretical study leading to a theoretical algorithm, efficient enough to lead in turn to concrete implementations, the standard test. Remember an *algorithm* is a *mathematical* object as respectable as a spectral sequence, a coherent sheave or a modular function. By the way, who knows significant applications ot the *numerical* values of the homotopy groups of spheres?

References

[1] M. G. Barratt and G. F. Paechter. *A note on* $\pi_r(V_{n,m})$. Proceedings of the National Academy of Sciences of USA, 1952, vol.38, pp.119-121.

[2] Ainhoa Berciano, Julio Rubio and Francis Sergeraert. *A case study of* A_∞*-structure*. Georgian Mathematical Journal, 2010, vol. 17, pp.57-77.

[3] William Browder. *Homology operations and loop spaces.* Illinois Journal of Mathematics, 1960, vol. 4, pp. 347-357.

[4] Edgar H. Brown Jr.. *Finite computability of Postnikov complexes.* Annals of Mathematics, 1957, vol. 65, pp. 1-20.

[5] Ronnie Brown. *The twisted Eilenberg-Zilber theorem.* Celebrazioni Arch. Secolo XX, Simp. Top., 1967, pp. 34-37.

[6] Henri Cartan. *Algèbres d'Eilenberg-MacLane et homotopie.* Séminaire Henri Cartan, 1954-5. Or *Henri Cartan, Œuvres*, Springer, 1979.

[7] Alonzo Church. *The calculi of lambda-conversion.* Princeton University Press, 1941.

[8] Xavier Dousson, Julio Rubio, Francis Sergeraert and Yvon Siret. *The Kenzo program.* `http://www-fourier.ujf-grenoble.fr/~sergerar/Kenzo/`

[9] Samuel Eilenberg, Saunders MacLane. *On the groups* $H(\pi, n)$*, I.* Annals of Mathematics, 1953, vol. 58, pp. 55-106.

[10] Samuel Eilenberg, Saunders MacLane. *On the groups* $H(\pi, n)$*, II.* Annals of Mathematics, 1954, vol. 60, pp. 49-139.

[11] Daniel M. Kan. *A combinatorial definition of homotopy groups.* Annals of Mathematics. 1958, vol. 67, pp. 282-312.

[12] R. James Milgram. *Iterated loop spaces.* Annals of Mathematics, 1966, vol. 84, pp. 386-403.

[13] Ieke Moerdijk. *A model for intuitionistic nonstandard arithmetic*. Annals of Pure and Applied Logic, 1995, vol.73, pp.37-51.

[14] Erik Palmgren. *Developments in Constructive Nonstandard Analysis*. Bulletin of Symbolic Logic, 1998, vol.4, pp.233-272.

[15] Jean-Pierre Serre. *Groupes d'homotopie et classes de groupes abéliens*. Annals of Mathematics, 1953, vol. 58, pp. 258-294.

[16] Julio Rubio. *Homologia efectiva y sucesión espectral de Eilenberg-Moore*. Thesis, Universidad de Zaragoza, 1988.
`http://dialnet.unirioja.es/servlet/tesis?codigo=1330`

[17] Julio Rubio. *Homologie effective des espaces de lacets itérés*. Thèse, Université Joseph Fourier, Grenoble 1991.
`http://dialnet.unirioja.es/servlet/tesis?codigo=1331`

[18] Julio Rubio, Francis Sergeraert. *Constructive Algebraic Topology*. In preparation, see `www-fourier.ujf-grenoble.fr/s̃ergerar`.

[19] Julio Rubio, Francis Sergeraert. *Constructive Algebraic Topology*. Bulletin des Sciences Mathématiques, 2002, vol. 126, pp. 389-412.

[20] Julio Rubio, Francis Sergeraert. *Algebraic Models for Homotopy Types*. Homology, Homotopy and Applications, 2005, vol.7, pp.139160.

[21] Julio Rubio, Francis Sergeraert. *Genova Lecture Notes*.
`http://www-fourier.ujf-grenoble.fr/~sergerar/...`
`.../~sergerar/Papers/Genova-Lecture-Notes.pdf`

[22] Francis Sergeraert. ✠$_k$*, objet du 3*^{e} *type*. Gazette des Mathématiciens, 2000, vol. 86, pp. 29-45.

[23] Wikipedia, article "*Closure (computer science)*".