Prépublication de l'Institut Fourier n° 620 (2003)
http://www-fourier.ujf-grenoble.fr/prepublications.html

Algebraic Models for Homotopy Types

Julio Rubio and Francis Sergeraert

October 1, 2003

As yet we are ignorant
of an effective method of computing
the cohomology of a Postnikov complex
from π_n and k^{n+1} [7].

Abstract

The classical problem of *algebraic models* for homotopy types is *precisely stated*, to our knowledge for the first time. Two different natural statements for this problem are produced, the simplest one being *entirely solved* by the notion of $\mathcal{SS}_{EH}$-structure, due to the authors. Other tentative solutions, Postnikov towers and E_∞-chain complexes are considered and compared with the $\mathcal{SS}_{EH}$-structures. In particular, which looks like a severe error about the usual understanding of the k-invariants is explained; which implies we seem far from a solution for the ideal statement of our problem. At the positive side, the problem stated above in the title inscription is solved.

1 Introduction.

Obtaining "algebraic" models for $\mathbb{Z}$-homotopy types is a major problem. The *statement* of the problem itself is a constant source of strong and regrettable ambiguities. We explain in this article why the adjective *algebraic* is in fact inappropriate, the right one being *computable* (or effective, constructive, ...).

The problem of the title can *then* be precisely stated in two different ways, the *hard* problem (Problem 5 in Section 2) and the *soft* problem (Problem 8 in Section 3). The notion of simplicial set with effective homology ($\mathcal{SS}_{EH}$), due to the authors, is a complete solution for the soft problem, very simple from a theoretical point of view, once the possibilities of functional programming are understood. This solution has led to an interesting concrete computer work, the Kenzo program, a little demonstrated in the article to give to the reader an *experimental* evidence that the stated results are correct.

Keywords: Algebraic Model, Homotopy Type, Spectral Sequence, E_∞-Operad, Postnikov System, k-Invariant, Computable Category, Constructive Algebraic Topology.
AMS-Clas: 55P15, 55Txx, 18D50, 55S45, 55-04.

Other solutions for the soft problem are based on the operadic techniques, and they are now intensively studied. The key point is the notion of E_∞-operad; a broad outline of the main results so obtained is given and compared with the $\mathcal{SS}_{EH}$ solution. The current result is that the $\mathcal{SS}_{EH}$ solution is, for the soft problem, terribly simpler; furthermore the operadic structures are interesting, of course give many useful informations, but are *by-products* of $\mathcal{SS}_{EH}$-structures. The good point of view for future work is probably a mixture of $\mathcal{SS}_{EH}$'s and operadic techniques, the last ones to be considered as good tools to understand and improve the computability results so easily obtained through $\mathcal{SS}_{EH}$'s.

The *hard problem* is *so* reduced to the problem of equivalence between sets of k-invariants, problem which, up to further information, seems open: we explain why the so-called k-invariants are not actual invariants and therefore *do not* solve the *hard problem*.

2 The right statement of the problem.

The construction of *algebraic models* for *homotopy types* is a "classical" problem in Algebraic Topology which, to our knowledge, has never been precisely stated, that is, *mathematically* stated. Experience shows the topologists have a rather imprecise idea about the exact nature of this problem, a situation frequently leading to misunderstandings or even sometimes to severe errors; an example of this sort being the usual belief that the so-called k-invariants are… invariants, an erroneous appreciation, see Sections 3.1 and 8.

Most of the topologists should agree with the following statement of our problem.

Problem 1 — *Let $\mathcal{H}$ be the* homotopy category*. How to design an* algebraic category $\mathcal{A}$ *and a functor $F : \mathcal{H} \to \mathcal{A}$ which is an equivalence of categories?*

Instead of working in the category $\mathcal{H}$, reputed to be a difficult category, you might work in the category $\mathcal{A}$, an algebraic category, hence probably a more convenient workspace. The image $F(X)$ of some homotopy type X would be an *algebraic* object, for example a chain complex provided with a sufficiently rich structure to entirely define a homotopy type. Problem 1 leads to an auxiliary problem.

Problem 2 — *What is the definition of an* algebraic *category?*

It happens that standard logic shows such a definition *cannot exist*; this is a direct consequence of the formalization of mathematics, asked for by Hilbert, and realized through various systems, mainly the so-called Zermelo-Fraenkel and Bernays-Von Neumann systems. In a sense, formalization of mathematics consists in making *entirely* algebraic our mathematical environment, even when we work in fields that are not usually considered as algebraic, like in analysis, probability, and also in topology.

The following example is fairly striking. Most of the topologists consider a simplicial set is *not* an algebraic object. A simplicial set S is a sequence of simplex sets (S_n) combined with some sets of operators between these simplex sets, appropriate composites of these operators having to satisfy a few simple relations. Most of the topologists consider a chain complex C_* provided with a module structure with respect to some (…algebraic!) operad $\mathcal{O}$ *is* an algebraic object. Such a chain complex is a sequence of chain groups (C_n) combined with some sets of operators between these chain groups and their tensor products, appropriate composites of these operators having to satisfy a large set of sophisticated relations. Where is the basic difference? This appreciation — an $\mathcal{O}$-algebra *is* an algebraic object and a simplicial set *is not* — is arbitrary. Furthermore a simplicial structure is simpler than an $\mathcal{O}$-algebra structure, so that a beginner in the subject would probably guess the first structure type is "more" algebraic than the second one. Must we recall we are working in mathematics, not in philosophy? Our workspace require *mathematical* definitions, not fuzzy speculative claims based only on vague traditions.

Terminology 3 — *In our current mathematical environment, the border between* algebraic *objects and* non-algebraic *objects cannot be* mathematically *defined.*

Let us continue our comparison between simplicial sets and chain complexes, which will eventually lead to the right point of view. The simplest example of an interesting result produced by Algebraic Topology is the Brouwer theorem, a direct consequence of the following.

Theorem 4 — *Let $i_n : S^{n-1} \to D^n$ be the canonical inclusion of the $(n-1)$-sphere into the n-ball. There does not exist a continuous map $\rho_n : D^n \to S^{n-1}$ such that the composite $\rho_n \circ i_n$ is the identity map of S^{n-1}.*

In fact, if you apply the H_{n-1}-functor to the data, the statement is transformed into: let $i : \mathbb{Z} \to 0$ be the null morphism; there does not exist a morphism $\rho : 0 \to \mathbb{Z}$ such that the composite $\rho \circ i$ is the identity morphism of $\mathbb{Z}$.

Most of the topologists think this process produces the result because the transformed problem has an *algebraic* nature, but this is erroneous. The *algebraic* qualifier is secondary and, as previously explained cannot be mathematically justified. The right qualifier in fact is *computable*. The transformed problem is a particular case of the following: let m, n and p be three non-negative integers, and $f : \mathbb{Z}^m \to \mathbb{Z}^n$ and $F : \mathbb{Z}^m \to \mathbb{Z}^p$ be two $\mathbb{Z}$-linear morphisms; does there exist a morphism $g : \mathbb{Z}^n \to \mathbb{Z}^p$ satisfying $g \circ f = F$? It is common to think of this problem as an algebraic one, but in fact the only important point for us is that there exists an *algorithm* giving the solution: a Smith reduction of the $\mathbb{Z}$-matrices representing f and F quickly gives the solution; in the case of the Brouwer problem, the Smith reduction is already done.

The previous considerations about simplicial sets give another idea. Because a simplicial set is in fact as "algebraic" as a homology group or a chain complex, why not to work directly with simplicial models for S^{n-1} and D^n? It is easy to give simplicial models with two (resp. three) non-degenerate simplices for S^{n-1} (resp. D^n), models that are undoubtedly "algebraic". But these models have an essential failing: they do not satisfy the Kan extension condition, so that they are not appropriate for working in the homotopy category $\mathcal{H}$. In general the Kan simplicial models are highly infinite and cannot be directly used for computations : any tentative solution using in an essential way the Kan simplicial sets raises hard computability problems. We will see later that our solution for "algebraic" models for homotopy types is a simple but subtle combination of simplicial sets most often *not of finite type* with chain complexes of finite type.

There is a common fundamental confusion between the *algebraic* and *computable* qualifiers, still present in the ordinary understanding of the very nature of Algebraic Topology. From this point of view, it can be useful to recall the frequent opinion of the pupils in secondary schools: "I prefer Algebra rather than Geometry, because in Algebra we can use *automatic* methods giving the results that are looked for; on the contrary, in Geometry, we often have to *discover* the appropriate method for some particular problem"; another example of the same confusion between *algebraic* and *computable*.

Let us look again at the statement of Problem 1. We see the requirement for the category $\mathcal{A}$ to be algebraic cannot be defined; in fact we are looking for a target category where *automatic* computations (pleonasm) can be undertaken. We so obtain a new statement for our problem.

Problem 5 (Hard Problem) — *Let $\mathcal{H}$ be the* homotopy category*. How to design a* computable category *$\mathcal{C}$ and a functor $F : \mathcal{H} \to \mathcal{C}$ which is an equivalence of categories?*

With the satellite problem:

Problem 6 — *What is the definition of a* computable *category?*

We do not want to consider the details of the last subject, an interesting subject, out of scope of the present paper; fundamentally different answers are possible, mainly from the following point of view: do you intend to apply the "computable" qualifier to the *elements* of an object in the category or to the *objects* themselves, or both? To our knowledge, the relevant corresponding theory is not yet settled[1]. The few examples given in the paper could be a good guideline toward the most natural solutions of this question.

In other words, we must click on the `rename` button and replace the incorrect identifier *Algebraic Topology* by the unique correct one: *Computable Topology*.

[1]For example the reference [13], interesting, cannot be useful for our main problem; look for the entry *equality* in the index, and you will quickly understand that no tool is provided there for the equality problem between objects of a category.

3 Three tentative solutions.

The current state of Algebraic Topology gives mainly three possibilities:

1. The Postnikov category;
2. The operadic solutions;
3. The authors' solution: the category $\mathcal{SS}_{EH}$.

In short, the first possibility is currently inadequate in the standard framework, because of an essential lack of computability, see the title inscription, and also because of the underlying classification problem which does not yet seem solved. It can be reasonably conjectured that the second idea, using operadic techniques, should in finite time lead to a complete solution, but we are still far from it. The $\mathcal{SS}_{EH}$ category solves a *subproblem*, the soft problem, stated a little later and furthermore makes the Postnikov category computable; a consequence is the fact that the Postnikov category, when modelled as a satellite category of the $\mathcal{SS}_{EH}$ category, solves the same subproblem. The gap about the classification problem is present for the three solutions.

Once the theoretical and concrete possibilities of functional programming are understood, the $\mathcal{SS}_{EH}$ category is not complicated, so that it has been possible to write down an interesting computer program implementing the $\mathcal{SS}_{EH}$ category and to use it, see [6] and Sections 5 and 7 of the present paper.

3.1 The Postnikov category.

Restriction 7 — *Unless otherwise stated, all our topological spaces are connected and simply connected.*

An object of the Postnikov category is a pair of sequences $((\pi_n)_{n\geq 2}, (k_n)_{n\geq 3})$, made of homotopy groups and "k-invariants" defining a *Postnikov tower* $(X_n)_{n\geq 2}$. The first stage of the tower X_2 is $K(\pi_2, 2)$, the first k-invariant $k_3 \in H^4(X_2, \pi_3)$ defines a fibration $X_3 \twoheadrightarrow X_2$ the fiber of which being $K(\pi_3, 3)$, and so on. It is not hard to define the valid morphisms between two towers, and we have so defined the Postnikov category $\mathcal{P}$. We know it is not a common opinion, but this category is as "algebraic" as the usual so-called algebraic categories; note in particular the ingredients defining a Postnikov tower are commutative groups and elements of some commutative groups; are not they *algebraic*?

The so-called k-invariants are not invariants, for the following reason: *different* k-invariants frequently give the *same* homotopy type. Identifying the corresponding equivalence classes is a problem which, to our knowledge, is still without any solution. Let us look at this simple example: what about the Postnikov towers with only $\pi_2 = \mathbb{Z}^p$, $\pi_5 = \mathbb{Z}$ and the other π_n's are null. The only relevant k-invariant is $k_5 \in H^6(K(\pi_2, 2), \pi_5) = \mathrm{Cub}(\mathbb{Z}^p, \mathbb{Z})$, the $\mathbb{Z}$-module of the cubical forms over $\mathbb{Z}^p$; making these cubical forms actual invariants amounts to being able to construct and describe in a computational way the quotient set $\mathrm{Cub}(\mathbb{Z}^p, \mathbb{Z})$ / (linear equivalence). We have questioned several arithmeticians and they did not know whether appropriate references would allow a k-invariant user to solve this problem: the classification problem does not seem to be *solved* by the "k-invariants" and our example is one of the simplest ones[2].

Let us quote certainly one of the best specialists in homotopy theory. Hans Baues explains in [3, p. 33] :

> Here k_n is actually an *invariant* of the homotopy type of X in the sense that a map $f : X \to Y$ satisfies
>
> $$(P_{n-1}f)^* k_n Y = (\pi_n f)_* k_n X$$
>
> in $H^{n+1}(P_{n-1}X, \pi_n Y)$.

This explanation is not correct; the cohomology class k_n would be an actual *invariant* of the homotopy type if a homotopy equivalence $f : X \simeq Y$ implies $k_n X \boxed{=} k_n Y$; in fact the framed equal sign does not make sense: the underlying cohomology groups are not the *same*, they are only, in the relevant cases, *isomorphic* and two invariants should be considered as "equal" as soon as they are in turn "isomorphic" in an obvious sense. Baues' relation only shows the "k-invariant" depends *functorially* from the data, but it is not an *invariant*; the definition would be acceptable if the isomorphism problem between the various possible k-invariants in the same homotopy class had a (computable) solution, but the simple example given before shows such a solution does not seem currently known. We will examine again this question in a more explicit way in Section 8, where another classical reference about k-invariants [10] is also studied.

This is probably the reason why Hans Baues uses entirely different techniques to obtain certainly the most interesting concrete results so far reached in the *general* classification problem; see [3, Section 11] and Baues' references in the same paper.

Let us consider the following subproblem of the hard one:

Problem 8 (Soft Problem) — *How to design a computable category $\mathcal{C}$ and a functor $F : \mathcal{C} \to \mathcal{H}$ such that any recursive homotopy type is in the image of F?*

In fact the *hard problem* as it is stated in Problem 5 cannot have a solution: the standard homotopy category $\mathcal{H}$ is much too rich to make it equivalent to a computable category. This is a situation analogous to which is well known for example for the real numbers. A computable real number is usually called a *recursive* real number and the set of the recursive real numbers is countable, much smaller than the set of "ordinary" real numbers, see [20]. In the same way:

[2]We would like to thank Daniel Lazard for his study (private communication) which opens several interesting research directions around this subject.

Definition 9 — A *recursive homotopy type* is defined by a recursive Postnikow tower $((\pi_n)_{n\geq 2}, (k_n)_{n\geq 3})$: the data of this tower are defined by an *algorithm* $n \mapsto (\pi_n, k_n)$. In other words the recursive homotopy category is the image of the canonical functor $\mathcal{P}_r \to \mathcal{H}$ if $\mathcal{P}_r$ is the category of the recursive Postnikov towers.

In fact, in the standard context, this definition does not make sense : the required algorithm must be able to *compute* the $H^{n+1}(X_{n-1}, \pi_n)$ to allow it to "choose" the next k_n, and classical Algebraic Topology does not solve this question. To our knowledge, there are currently only two solutions for this problem, independantly and simultaneously found by Rolf Schön [14] and the present authors [15, 12, 17]. We will see later that our $\mathcal{SS}_{EH}$ category allows us in particular to compute the cohomology groups $H^{n+1}(X_{n-1}, \pi_n)$ when the previous data are available, so recursively defining where the k_n is to be chosen. In this way our category $\mathcal{SS}_{EH}$ makes coherent Definition 9 and, *then only*, the Postnikov category becomes an obvious solution for the *soft problem.* In fact we will also see the $\mathcal{SS}_{EH}$ category directly gives a solution for the *soft problem.*

It should be clear now that in the statement of the *hard problem*, the category $\mathcal{H}$ must obviously be replaced by the category of the *recursive* homotopy types. Up to a finite dimension, this amounts only to requiring that the homotopy groups π_n are of finite type, but if the situation is considered without any dimension limit, the requirement is much stronger.

Restriction 10 — *From now on, all our categories are implicitely limited to* recursive *objects and* recursive *morphisms.*

The *soft problem* then is the same as the *hard problem* except that the classification question is given up.

3.2 The operadic solutions.

Many interesting works have been and are currently undertaken to reach operadic solutions for the *hard* and for the *soft problem.* Probably the most advanced one is due to M. Mandell [9], usually considered as a "terminal" solution. Of course we do not intend to reduce the interest of this work, essential, but Mandell's article *is not* a solution for the *hard problem* : the computability question is not considered, and the proposed solution invokes numerous layers of sophisticated techniques, so that the computational gap is not secondary. In fact the interesting next problem raised by Mandell's paper is the following : is it possible to "naturally" extend the paper to obtain the corresponding *effective* statements, or on the contrary is it necessary to add something which is essentially new and/or different? Note in particular that the classification problem implicitly "solved" by Mandell's paper would solve crucial computability problems in arithmetic.

We suspect the situation is analogous to which has been observed about the spectral sequences. Mandell's paper can in a sense be compared to Koszul and Serre's classical papers creating the technology of spectral sequences; an essential technology but in its traditional form which *does not solve* the computability problem in Homological Algebra in general, in Algebraic Topology in particular. The most efficient solution which is currently available for this computability problem [12, 6] needs further tools, mainly the so-called Basic Perturbation Lemma and an intensive use of functional programming techniques which have their own interest. We will explain later how our solution around the $\mathcal{SS}_{EH}$ category could be the right complementary ingredient to be combined with Mandell's work to eventually solve the *hard problem*.

4 The category $\mathcal{SS}_{EH}$.

Restriction 11 — *All the chain complexes considered from now on are implicitly assumed to be* free *$\mathbb{Z}$-complexes, not necessarily of finite type.*

The notion of *reduction*[3] is well known.

Definition 12 — A *reduction* $\rho : C_* \Rrightarrow D_*$ between two chain complexes C_* and D_* is a triple $\rho = (f, g, h)$ where:

1. The first component f is a chain complex morphism $f : C_* \to D_*$;
2. The second component g is a chain complex morphism $g : D_* \to C_*$;
3. The third component h is a homotopy operator (degree $= +1$) $h : C_* \to C_*$;
4. These components satisfy the relations :
   (a) $f \circ h = 0$;
   (b) $h \circ g = 0$;
   (c) $h \circ h = 0$;
   (d) $\mathrm{id}_{D_*} = f \circ g$
   (e) $\mathrm{id}_{C_*} = g \circ f + d_{C_*} \circ h + h \circ d_{C_*}$.

These relations express in an *effective* way how the "big" chain complex C_* is the direct sum of the "small" one D_* and an *acyclic* one, namely the kernel of f.

Definition 13 — A *strong chain equivalence* (or simply an *equivalence*):

$$\varepsilon : C_* \Lleftarrow\!\!\Rrightarrow D_*$$

is a pair of reductions $\varepsilon = (\rho_\ell, \rho_r)$ where:

$$C_* \overset{\rho_\ell}{\Lleftarrow} \widehat{C}_* \overset{\rho_r}{\Rrightarrow} D_*$$

with $\widehat{C}_*$ some intermediate chain complex.

[3] Often called *contraction*, but this is a non-negligible terminological error : a *contraction* is a *topological* object and a reduction is only an *algebraic* object; it is important to understand a reduction *does not* solve the underlying *topological* problem. Exercise: why this remark does not contradict Terminology 3?

Definition 14 — A *simplicial set with effective homology* is a 4-tuple

$$X_{EH} = (X, C_*X, EC_*^X, \varepsilon^X)$$

where:

1. The first component X is a *locally effective* simplicial set;
2. The second component C_*X is the *locally effective* chain complex canonically associated to X;
3. The third component EC_*^X is an *effective* chain complex;
4. The last component ε^X is a *strong chain equivalence* $\varepsilon^X : C_*X \lll\!\!\Longrightarrow\!\!\ggg EC_*^X$.

An *effective* chain complex is an ordinary object, no surprise; it is an algorithm $n \mapsto (C_n, d_n)$ where, for every integer n, the corresponding chain group C_n is a free $\mathbb{Z}$-module of finite type, and d_n is a $\mathbb{Z}$-matrix describing the boundary operator $d_n : C_n \to C_{n-1}$. Elementary algorithms then allow to compute the homology groups of such a complex. The third component EC_*^X of a simplicial set with effective homology is of this sort.

A *locally effective* chain complex is *quite different.* It is an algorithm:

$$n \mapsto (\chi_n, d_n)$$

to be interpreted as follows.

1. The first component χ_n of a result is also an algorithm $\chi_n : \mathcal{U} \to \{\top, \bot\}$ where $\mathcal{U}$ (universe) is the set of *all* the machine objects, so that for every machine object ω, the algorithm χ_n returns $\chi_n(\omega) \in \{\top, \bot\}$, that is, true or false, true if and only if ω is a *generator* of the n-th chain group of the underlying chain complex.
2. The second component d_n of a result is again an algorithm: if $\chi_n(\omega) = \top$, then $d_n(\omega)$ is defined and is the boundary of the generator ω, therefore a finite $\mathbb{Z}$-combination of generators of degree $n - 1$.

The set $\mathcal{U}$, for any reasonable machine model, is infinite countable, so that a locally effective chain complex in general *is not of finite type.* The adverb *locally* has the following meaning: if someone produces some (every!) generator ω of degree n, then the d_n-component is able to compute the boundary $d_n(\omega)$. The terminology *generator-wise* effective chain complex would be more precise but a little unwieldy.

A non-interesting but typical example of locally effective chain complex would be produced by $\chi_n(\omega) = \top$ if and only if $\omega \in \mathbb{N}$, independently of n, and $d_n(\omega) = 0$ for every $n \in \mathbb{Z}$ and $\omega \in \mathbb{N}$. In other words the underlying chain complex would be the periodic one $C_n = \mathbb{Z}^{(\mathbb{N})}$ with a null boundary.

Standard logic shows in general the homology groups of a locally effective chain complex *are not* computable; this is an avatar of the Gödel-Turing-Church-Post theorems about incompleteness. The second component C_*X of a simplicial set with effective homology is of this sort.

A locally effective simplicial set is defined in the same way; the simplices are defined through characteristic algorithms χ_n, and instead of computing boundaries, a set of appropriate operators compute faces and degeneracies.

The second component C_*X of a simplicial set with effective homology is redundant: a simple algorithm can construct it from the locally effective simplicial set X; and strictly speaking, we could forget it in the presentation. But the key points in an object with effective homology are:

1. The main components are two $\mathbb{Z}$-free chain complexes C_*X and EC_*^X, the first one being a direct consequence of the underlying object X, the second one describing the homology of this object, reachable through an elementary algorithm;
2. The component C_*X is *locally effective* allowing it not to be of finite type, with the drawback that in general its homology is not computable;
3. The component EC_*^X is *effective*, therefore of finite type with a computable homology;
4. The equivalence ε^X is the key connection between the locally effective object C_*X and the effective one EC_*^X.

and it is hoped the very nature of this organization is better explained in the notation $(X, C_*X, EC_*^X, \varepsilon^X)$.

Theorem 15 — *The category $\mathcal{SS}_{EH}$ is a solution for the* soft problem*.*

It is not possible in the framework of this paper to give a *proof* of Theorem 15, we will show only a *demonstration.* We apologize for the poor joke: "demonstration" has two different meanings in our context, it can be a *mathematical* proof, ant it can be also a *machine* (computer) demonstration. It is expected in this case a machine demonstration should give to the reader a strong conviction the machine program Kenzo *contains* a proof of Theorem 15. This is the aim of Sections 5 and 7.

5 A small machine demonstration.

This section uses a small machine demonstration to explain how, thanks to the powerful computer language Common Lisp, the Kenzo program[6] makes the objects and morphisms of the $\mathcal{SS}_{EH}$ category concretely available to the topologist.

Let us consider the following space :

$$X = \Omega(\Omega(P^\infty(\mathbb{R})/P^3(\mathbb{R})) \cup_4 D^4) \cup_2 D^3$$

The infinite real projective space truncated to the dimension 4, $P^\infty(\mathbb{R})/P^3(\mathbb{R})$, is firstly considered; its loop space is constructed and the homotopy of this loop space begins with $\pi_3 = \mathbb{Z}$; so that attaching a 4-cell by a map $\partial D^4 \to S^3$ of degree 4 makes sense and this is done. The loop space functor is again applied to the last space and finally a 3-cell is attached by a map of degree 2. This artificial space X is chosen because it is not too complicated, yet it accumulates the main known obstacles to the theoretical and concrete computation of homology groups in small dimensions.

The space X is an object of the category $\mathcal{SS}_{EH}$, so that the Kenzo program can construct it as such an object. As follows:

```
> (progn
   (setf P4 (r-proj-space 4))
   (setf OP4 (loop-space P4))
   (setf attach-4-4
     (list (loop3 0 4 4) (loop3) (loop3) (loop3) (loop3)))
   (setf DOP4 (disk-pasting OP4 4 'D4 attach-4-4))
   (setf ODOP4 (loop-space DOP4))
   (setf attach-3-2
     (list (loop3 0 (loop3 0 4 1) 2) (loop3) (loop3) (loop3)))
   (setf X (disk-pasting ODOP4 3 'D3 attach-3-2))) ✠
```

We cannot explain the technical details of the construction, but most of the statements are self-explanatory. Each object is located by a symbol and the assignment is set through a `setf` Lisp statement. For example the initial truncated projective space is assigned to the symbol `P4`. An object such as `attach-4-4` describes an attaching map as a *simplicial* map $\partial\Delta^4 \to$ `OP4` and this description is then used by the Lisp function `disk-pasting` which constructs the desired space by attaching a cell according to the descriptor `attach-4-4`. The same for the end of the construction.

When this statement is executed, Lisp returns:

```
   ...
   (setf X (disk-pasting ODOP4 3 'D3 attach-3-2))) ✠
[K17 Simplicial-Set]
```

A maltese cross ✠ means the Lisp statement is complete, the `read` stage of the `read-eval-print` Lisp cycle is finished, the `eval` stage starts for an execution of the just read Lisp statement, it is the stage where the machine actually works, *evaluating* the statement; most often, an object is *returned* (printed), it is the result of the evaluation process, in this case the simplicial set #`K17`, located through the `X` symbol. This object `X` is a (machine) version *with effective homology* of the topological space X.

So that we can ask for the *effective homology* of X; it is reached by the function `efhm` (effective homology) and assigned to the symbol `SCE` (strong chain equivalence):

```
> (setf SCE (efhm X)) ✠
[K268 Equivalence K17 <= K256 => K258]
```

The Kenzo program *returns* a (strong) equivalence (Definition 13) between the chain complexes #`K17` and #`K258`. Usually it is understood a simplicial set *produces* an associated chain complex, but we may conversely consider that a simplicial set is nothing but a chain complex where a simplicial structure is *added*, compatible with the differential; it is the right point of view and Kenzo follows this idea. Please compare with the discussion after Problem 2: it should be more and more obvious that a simplicial set is itself a chain complex with a further *algebraic* (!) structure. In other words if you are only looking for an *algebraic model for a homotopy type*, the notion of simplicial set is a simple undeniable definitive solution; this is the reason why you *must* add a computability requirement to finally obtain an interesting problem.

In our equivalence describing the effective homology of X, the right chain complex #`K258` is *effective*, the left one #`K17` is not:

```
> (basis (K 258) 4) ✠
(<<AlLp[4 <<AlLp[5 6]>>]>> <<AlLp[2 <<AlLp[3 4]>>][2 <<AlLp[3 4]>>]>>)
> (length *) ✠
2
> (basis (K 17) 4) ✠
Error: attempt to call ‘:LOCALLY-EFFECTIVE’ which is an undefined function.
```

The basis in dimension 4 of the chain complex #`K258` is computed, it is a list of length 2 (‘`*`’ = the last returned object). The elements of the basis themselves are “algebraic loops” (`AlLp`), elements of some appropriate cobar constructions.

On the contrary you see it is not possible to obtain the basis in dimension 4 of the chain complex #`K17` $= C_*X$; the necessary functional object is in fact the keyword `:locally-effective` which generates an error.

A homology group of X can be computed:

```
> (homology X 5) ✠
Homology in dimension 5 :
Component Z/4Z
Component Z/2Z
Component Z
---done---
```

which means $H_5X = \mathbb{Z}_4 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}$. It is in fact the homology of #`K258`:

```
> (homology (K 258) 5) ✠
Homology in dimension 5 :
Component Z/4Z
Component Z/2Z
Component Z
---done---
```

The strong chain equivalence #K268 contains three chain complexes *and* two reductions, therefore four chain complex morphisms and two homotopy operators. In particular there is in the right reduction $\rho_r : K_{256} \Rrightarrow K_{258}$ a right $g : K_{258} \to K_{256}$ reachable by means of a rg function in the program; in the same way the left reduction $\rho_\ell : K_{17} \Lleftarrow K_{256}$ contains a left $f : K_{256} \to K_{17}$ reachable thanks to a lf function. The Kenzo program can use these maps for arbitrary generators or combinations. For example the next Lisp statements play to verify the composite of the left f and the right g is compatible with the differentials.

We assign to the symbol gen the first generator of #K258 in dimension 4, we apply the right g (rg) to this generator, then the left f (lf), finally the differential of #K17:

```
> (setf gen (first (basis (K 258) 4))) ✠
<<AlLp[4 <<AlLp[5 6]>>]>>
> (rg SCE 4 gen) ✠
------------------------------------------------------------------------{CMBN 4}
 <-1 * <BcnB <TnPr <<AlLp[4 <<Loop[2-1 4][4-3 4]>>]>> <TnPr ... ...
 <1 * <BcnB <TnPr ... ...
-----------------------------------------------------------------------------
> (lf SCE *) ✠
------------------------------------------------------------------------{CMBN 4}
 <-2 * <<Loop[1-0 <<Loop[4]>>][3-2 <<Loop[4]>>]>>>
 <2 * <<Loop[2-0 ...
 [... Lines deleted...]
-----------------------------------------------------------------------------
> (? (K 17) *) ✠
------------------------------------------------------------------------{CMBN 3}
<-2 * <<Loop[<<Loop[3 4][5]>>]>>>
-----------------------------------------------------------------------------
```

The result is an actual "loop of loops", $(-2)\times$ some simplex in X. Large parts of the intermediate results are not showed. A result between two dash lines '---' labeled for example {CMBN 3} is a combination of degree 3 of integer coefficients and generators, one term per line.

The other path consists in applying to the same generator firstly the differential of #K258 and then the same maps:

```
> (? (k 258) 4 gen) ✠
--------------------------------------------------------------------{CMBN 3}
<2 * <<AlLp[3 <<AlLp[4 5]>>]>>>
------------------------------------------------------------------------------
> (rg sce *) ✠
--------------------------------------------------------------------{CMBN 3}
<2 * <BcnB <TnPr <<AlLp[3 <<Loop[3 4][5]>>]>> <TnPr <<Loop>> <<Loop>>>>>>
<-2 * <BcnD <<AlLp[3 <BcnB <TnPr <<AlLp[4 5]>> <TnPr 0 <<Loop>>>>>]>>>>
<2 * <BcnD <<AlLp[3 <BcnD <<AlLp[4 5]>>>]>>>>
------------------------------------------------------------------------------
> (lf sce *) ✠
--------------------------------------------------------------------{CMBN 3}
<-2 * <<Loop[<<Loop[3 4][5]>>]>>>
------------------------------------------------------------------------------
```

The results are the same.

[Section to be continued, see Section 7]

6 The fundamental theorem of Effective Homology.

It is well known (?) the classical spectral sequences (Serre, Eilenberg-Moore, Adams, ...) *are not* algorithms. See for example [11, Section 1.1], in particular the comments following the unique theorem of the quoted section: most often, the available input for a spectral sequence *does not* determine the higher differentials. Something more is necessary for this essential problem, and it happens the category $\mathcal{SS}_{EH}$ is from this point of view a perfect solution; moreover it is a simple solution, once the possibilities of functional programming are understood.

Meta-Theorem 16 — *Let*

$$\chi : (X_i)_{1 \leq i \leq n} \mapsto Y$$

be a "reasonable" construction of the Algebraic Topology world producing Y from the X_i's. Then an algorithm χ_{EH} can be written down which is a version with effective homology of the construction χ:

$$\chi_{EH} : ((X_i)_{EH})_{1 \leq i \leq n} \mapsto Y_{EH}$$

Most often the X_i's and Y are topological spaces. A construction is "reasonable" if it leads to some classical spectral sequence giving to the topologists the illusion that if the homology (for example) of the X_i's is known, then the homology of Y can be "deduced".

A typical and important situation of this sort is the case where X is a simply connected space and $\chi = \Omega$ is the loop space functor: $Y = \Omega X$. The Eilenberg-Moore spectral sequence gives interesting relations between H_*X and $H_*\Omega X$, but this spectral sequence *is not* an algorithm computing $H_*\Omega X$ from H_*X, for a simple reason: it is possible $H_*X = H_*X'$ and $H_*\Omega X \neq H_*\Omega X'$. More precisely, the cobar construction [1] gives the homology of the *first* loop space when some coproduct is available on H_*X, but the cobar construction *does not* give a coproduct on $H_*\Omega X$, so that the process cannot be iterated; this is *Adams' problem*: how to iterate the cobar construction? More than twenty years after Adams, Baues succeeded in a beautiful work [2] in iterating *one time* the cobar construction, giving the homology of the second loop space $\Omega^2 X$ in reasonable situations, but Baues' method cannot be extended for the homology of $\Omega^3 X$ either.

The category $\mathcal{SS}_{EH}$ gives at once a complete and simple solution for Adams' problem; it is a consequence of the following particular case of Meta-Theorem 16.

Theorem 17 — *An algorithm Ω_{EH} can be written down:*

$$\Omega_{EH} : X_{EH} \mapsto (\Omega X)_{EH}$$

producing a version with effective homology *of the loop space ΩX when a version* with effective homology *of the initial simply connected space X is given.*

The algorithm Ω_{EH} not only *can be* written down, but *it is* written down; the algorithm Ω_{EH} is certainly the most important component of the Kenzo program [6], a program which is by itself the most detailed proof which can be required for Theorem 17, of course not very convenient for an ordinary reader[4].

The data type of the output Y_{EH} is exactly the same as the data type of the input X_{EH}, so that the algorithm Ω_{EH} can be trivially iterated.

Theorem 18 (Solution of Adams' problem[5]) — *An algorithm* **ICB** *(iterated cobar) can be written down:*

$$\mathbf{ICB} : (X_{EH}, n) \mapsto (\Omega^n X)_{EH}$$

which produces a version with effective homology *of the n-th loop space $\Omega^n X$ when a version* with effective homology *of the initial space X, assumed to be n-connected, is given.*

When $X_{EH} = (X, C_*X, EC_*^X, \varepsilon^X)$ is given, the algorithm **ICB** produces a 4-tuple $(\Omega^n X)_{EH} = (\Omega^n X, C_*\Omega^n X, EC_*^{\Omega^n X}, \varepsilon^{\Omega^n X})$, where the "$n$-th cobar" of EC_*^X

[4]Several articles containing such a proof written in common mathematical language have been proposed to various mathematical journals, but they were always rejected by the editorial boards, see in particular [18]. It is likely that the totally new nature of the result, which can be stated and proved only in a computational framework, does not fit the standard style expected by the referees. A matter of evolution; yet the scientific journals should precisely be mainly interested by the papers reflecting new unavoidable scientific trends, papers which of course are a little more difficult to be appropriately refereed.
See [16] for a survey which gives the plan and the main ideas of the proof of Theorem 17.

[5]This theorem *is* also a solution for Carlsson and Milgram's problem [5, p. 545, Section 6], a problem the authors cannot properly state, again because of the lack of a computational framework.

is the third component $EC_*^{\Omega^n X}$. This n-th cobar cannot be constructed from EC_*^X only; the first cobar needs some coproduct and the n-th cobar needs much more supplementary informations hidden in X and ε^X; these objects X and ε^X are locally effective and model mathematical objects which are infinite; yet X and ε^X are *finite* machine objects (pleonasm), namely finite bit strings actually created, processed and used by the Kenzo program; this process works thanks to *functional programming*.

The further components $\Omega^n X$ and $\varepsilon^{\Omega^n X}$ in the result would allow to undertake other calculations starting from $\Omega^n X$.

7 A small machine demonstration [sequel].

Let us consider again the space X of Section 5. The Kenzo program had constructed a version *with effective homology* of this space, allowing in particular to compute its homology groups. Much more important, because of Theorem 17, the machine object Ω_{EH} of the same program can be applied to produce a version *with effective homology* of the loop space ΩX:

```
> (setf OX (loop-space X)) ✠
[K273 Simplicial-Group]
```

Kenzo[6] returns a new locally effective simplicial *group*, the Kan model of ΩX and its effective homology:

```
> (setf SCE2 (efhm OX)) ✠
[K405 Equivalence K273 <= K395 => K391]
```

with which exactly the same experiences which were tried with the effective homology of X in Section 5 could be repeated. In particular the right chain complex #`K391` is effective and allows a user to compute a homology group:

```
> (homology OX 5) ✠
Component Z16
Component Z8
...
...
Component Z2
Component Z2
```

[6]The Kenzo function `loop-space` follows the modern rules of *Object Oriented Programming* (OOP): if the argument is a simplicial set, then the Kan model of the loop space is constructed, and if furthermore the argument contains the effective homology of the initial simplicial set, then the `loop-space` function constructs also the effective homology of the loop space.

where we have deleted 21 lines, for the result is in fact:

$$H_5(\Omega(\Omega(\Omega(P^\infty(\mathbb{R})/P^3(\mathbb{R})) \cup_4 D^4) \cup_2 D^3)) = \mathbb{Z}_{16} \oplus \mathbb{Z}_8 \oplus \mathbb{Z}_2^{23}.$$

This is nothing but the corresponding homology group of #`K391`.

To our knowledge, the Kenzo program is the only object, human or not, currently able to reach this result. See also [14, 19] for two other interesting theoretical solutions which unfortunately have not yet led to concrete machine programs.

Adams' and Carlsson-Milgram's problems are solved.

8 The category $SSEH$ and the Postnikov category.

A rough "definition" of the Postnikov category was given in Section 3.1, but we must be now more precise to obtain a correct relation between the category $\mathcal{SS}_{EH}$ and the Postnikov category.

Definition 19 — An *Abelian group of finite type* π is a direct sum $\pi = \mathbb{Z}/d_1\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/d_n\mathbb{Z}$ where every d_i is a non-negative integer and d_{i-1} divides d_i. We denote by Π the *set* of *these* groups.

The set Π is designed for having exactly *one* group isomorphic to every Abelian group of finite type. For example the group H_5X in Section 5 is *isomorphic* to the element of Π defined by the integer triple $(2, 4, 0)$, that is, the group $\mathbb{Z}_2 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}$, but there are 128 different isomorphisms.

Definition 20 — A Postnikov tower is a pair of sequences $((\pi_n)_{n\geq 2}, (k_n)_{n\geq 3})$ where $\pi_n \in \Pi$ and $k_n \in H^{n+1}(X_{n-1}, \pi_n)$, if X_n is the n-th stage of the Postnikov tower constructed by the standard process.

Because π_n *is* some *precise* group, the standard Eilenberg-MacLane process gives a *precise* $K(\pi_n, n)$, a Kan simplicial set, producing in turn a *precise* n-th Postnikov stage X_n and with the next π_{n+1} a *precise* cohomology group $H^{n+2}(X_n, \pi_{n+1})$ *where* the k_{n+1} *must* be "chosen". A Postnikov tower so produces in a deterministic way a realization X. A morphism $f : (\pi_n, k_n) \to (\pi'_n, k'_n)$ between two Postnikov towers is a collection $(f_n : \pi_n \to \pi'_n)$ of group homomorphisms compatible with the k_n's and k'_n's, that is, satisfying Baues' relation, and we have so defined the Postnikov category $\mathcal{P}$. The isomorphism problem consists in deciding whether two Postnikov towers (π_n, k_n) and (π'_n, k'_n) produce realizations with the same homotopy type, that is, because of the context, that are isomorphic. Of course the condition $\pi_n = \pi'_n$ is required for every n, but simple examples show that the condition $k_n = k'_n$ on the contrary is not necessarily required. This is the reason why the k_n's *are not* invariants of the homotopy type.

The computable category $\mathcal{SS}_{EH}$ makes the realization process computable.

Theorem 21 — *An algorithm* **PR** *(Postnikov realization) can be written down:*

$$\mathbf{PR} : \mathcal{P} \to \mathcal{SS}_{EH}$$

implementing the realization process.

In fact the situation is significantly more complex. Before being… *true*, the statement of this theorem must *make sense*, so that a machine implementation of the category $\mathcal{P}$ must be available, at least from a theoretical point of view. This is obtained thanks to the category $\mathcal{SS}_{EH}$ itself: a component k_n must be a machine object, so that the *data type* $H^{n+1}(X_{n-1}, \pi_n)$ where k_n is to be picked up must be *previously* defined, which is possible only if a calculation of this cohomology group can be undertaken. And again it is the category $\mathcal{SS}_{EH}$ which gives this possibility. It is an amusing situation where a category, the category $\mathcal{SS}_{EH}$, is simultanously used to give sense to the *statement* of a theorem, and *synchronously* finally to prove it.

Combining Theorem 21 with the appropriate particular cases of Meta-Theorem 16, we see the problem implicitly stated in the title inscription is now solved. In particular, the Kenzo program allows a Postnikov user to undertake many computations of this sort.

It is not possible, with the currently available tools, to make the categories $\mathcal{P}$ and $\mathcal{SS}_{EH}$ *effectively* equivalent, of course up to the homotopy relation.

Theorem 22 — *An algorithm* **SP** *can be written down:*

$$\mathbf{SP}: \quad \mathcal{SS}_{EH} \to \mathcal{P} \times \mathcal{I} \quad : \quad X \mapsto (\pi_n, k_n, I_n)_{n \geq 2}$$

where the component I_n is some *isomorphism $I_n : \pi_n(X) \cong \pi_n \in \Pi =$ the set of "canonical" models of Abelian groups of finite type; when the I_n's are chosen,* then only *the k-invariants k_n are unambiguously defined.*

The algorithm **SP** is essentially non-unique, for the *choice* of the component I_n is arbitrary, and the various choices of these isomorphisms will produce all the possible collections of k-"invariants" (!). Unfortunately the group $GL(p, \mathbb{Z})$ for example is infinite for $p > 1$, so that it is a non-trivial arithmetical problem to determine whether two collections of k-invariants correspond to the same homotopy type or not. For example the problem has an obvious solution up to arbitrary dimensions if every π_n is finite, but as soon as a π_n is not finite, we are in front of interesting but difficult problems of arithmetic, to our knowledge not yet solved in general[7].

[7]Compare with [14, pp. 54-59]; the possible equivalence of k_n and k'_n with respect a possible automorphism of the *last* π_n is there proved decidable, which is relatively easy. But this does not seem to be sufficient, because the possible automorphisms of *all* the previous π_m, $m \leq n$, must be considered. The example of a Postnikov tower where only $\pi_2 = \mathbb{Z}^p$ and $\pi_5 = \mathbb{Z}$ given in Section 3.1 shows the main problem for the equivalence of k-invariants is in the automorphisms of π_2, because the group of automorphisms is $GL(p, \mathbb{Z})$, leading to essential hard arithmetical

In conclusion, thanks to the computable category $\mathcal{SS}_{EH}$, the category $\mathcal{P}$ *becomes* also a computable category. There are "good" but non-canonical correspondances between these categories. Both categories solve the *soft problem* and from this point of view give equivalent results. Both categories would solve the *hard problem* if the equivalence problem between systems of k-"invariants" was *effectively* solved.

It is easier now to understand the common confusion about the nature of the k-invariants. We follow exactly here [10, §25], up to obvious slight differences of notations. We can start with a minimal Kan model of X, "unique" up to numerous *different* isomorphisms in general; the Postnikov stages X_{n-1} and X_n are then *canonical* quotients of X. There is also a *canonical* fibration between X_n and X_{n-1}, the fiber space of which being the *canonical* space $K(\pi_n\underline{(X)}, n)$, defining *unambiguously* a

$$k_n \in H^{n+1}(X_{n-1}, \pi_n\underline{(X)}) = H^{n+1}(\underline{X}/\sim_{n-1}, \pi_n\underline{(X)}).$$

It is then clear that a claimed invariant living in $H^{n+1}(X_{n-1}, \pi_n)$ with $\pi_n \in \Pi$ depends on an isomorphism $\pi_n \cong \pi_n(X)$, which is essentially the g correctly considered at [10, Theorem 25.7]. But if you think that X_{n-1} comes *only* from the previous data $\pi_2, \pi_3, k_3, \ldots, \pi_{n-1}, k_{n-1}$ and not from X itself, you can *freely* apply a self-equivalence of X_{n-1} to change (!) the invariant, the *fibration* $X_n \to X_{n-1}$ is changed, but on the contrary the homotopy type of X_n remains unchanged: *different* invariants correspond to *equal homotopy types*. The group of all the self-equivalences of X_{n-1} must be considered, of course in general a serious question. The only way to cancel this ambiguity consists in choosing a well defined partial equivalence between X and X_{n-1}, which amounts to choosing some isomorphisms $\pi_i \cong \pi_i(X)$ for $2 \leq i < n$.

Maybe it is useful to recall an *invariant* with respect to any notion must be chosen in a "fixed" world independent of the object the invariant of which is being defined. Otherwise the definitively simplest complete invariant for the homotopy type of X is X itself. Not very interesting. The *non-ambiguous* definition of k_n above lives in a set the definition of which contains two occurences of X and this is forbidden when an *invariant* of X is defined. And which must be called an error in [10, Theorem 25.7] comes from the definition $k_n \in H^{n+1}(X_{n-1}, \pi_n)$ (p. 113, line 19), and the notation $\pi_n = \pi_n(X, \emptyset)$ (line 14), again two illegal occurences of X in this situation. It is the reason why, in the statement of Theorem 22, π_n is not *equal* to $\pi_n(X)$, they are only isomorphic through some isomorphism which plays an essential role.

The classical example of the minimal polynomial of a matrix is helpful; if the ground field K is given, then the minimal polynomial can be chosen once and for

problems. In a later preprint, not published, Rolf Schön considers again the problem, solves it in the case where all the π_n's are finite, and *announces* a general solution which "takes considerable work". The authors have not succeeded in getting in touch with Rolf Schön for several years, and any indication about his current location would be welcome. Note that these comments in particular cancel the assertion in [17, Section 5.4] about the classification problem, which assumed the correctness of Schön's paper: the solutions called JS, SRH and SRG in [17] solve only the *soft problem*; an essential gap remains present for the *hard problem*.

all in $K[\lambda]$, a set of polynomials independent of the particular considered matrix. So that if two matrices are conjugate, more generally if two endomorphisms of two finite-dimensional K-vector spaces are *conjugate*, their minimal polynomials are *equal*, not mysteriously "*isomorphic*"; this is the reason why the miminal polynomial is a correct conjugation *invariant*.

9 The category $\mathcal{SS}_{EH}$ and the E_∞-algebras.

We had briefly mentioned in Section 3.2 the possibility of other solutions for the *hard problem* based on E_∞-operads.

A particularly interesting E_∞-operad is the *surjection operad* $\mathcal{S}$ defined and studied in [4], a work undertaken to make completely explicit some results of Mandell's paper [9] already quoted in Section 3.2. The so-called surjection operad and its action on a simplicial set can be understood as a "complete" generalization of the Steenrod operations, and we therefore propose to call it the *Steenrod operad*, which furthermore allows to naturally keep the same notation $\mathcal{S}$.

Theorem 23 — *A "natural" algorithm* **SSC** *(simplicial sets to Steenrod chain complexes) can be written down:*

$$\mathbf{SSC} : \mathcal{SS}_{EH} \to \mathcal{CC}_{\mathcal{S}}$$

where $\mathcal{CC}_{\mathcal{S}}$ is the category of the free $\mathbb{Z}$-chain complexes of finite type provided with a $CB\mathcal{S}$-operadic structure.

An appropriate bar construction can be applied to the operad $\mathcal{S}$ to produce a cooperad $B\mathcal{S}$; then an analogous cobar construction can in turn be applied to this cooperad to produce a new operad denoted by $CB\mathcal{S}$, another model for an E_∞-operad which has the following advantage[8]: let $f : C_* \to D_*$ a chain equivalence between two free $\mathbb{Z}$-chain complexes; then every $CB\mathcal{S}$-structure on C_* induces such a structure on D_*.

Definition 24 — A Steenrod chain complex is a free $\mathbb{Z}$-chain complex provided with a $\mathcal{S}$-structure or with a $CB\mathcal{S}$-structure.

Let X be an object of $\mathcal{SS}_{EH}$, that is, a simplicial set with effective homology. The article [4] explains how the initial definition by Steenrod of his famous cohomological operations can be naturally used to install a canonical $\mathcal{S}$-structure on the chain complex C_*X; the strong chain equivalence $\varepsilon^X : C_*X \lll\!\!\Longleftrightarrow\!\!\ggg EC_*^X$ then allows to install a $CB\mathcal{S}$-structure on EC_*^X, and this is enough to define the functor **SSC**.

Taking account of Mandell's article [9], the following problems are natural.

[8]We would like to thank Tornike Kadeishvili for his clear and useful explanations about this process.

Problem 25 — *Does there exist an* algorithm **R** *(realizability):*

$$\mathbf{R} : \mathcal{CC}_{\mathcal{S}} \to \mathbf{Bool} = \{\top, \bot\}$$

allowing to decide whether some object $C_* \in \mathcal{CC}_{\mathcal{S}}$ correspond or not to some topological object, of course up to some given dimension?

Because of the Characterization Theorem [9, p. 2], a solution for this problem is probably a "simple" exercise, simple in theory but the operad $CB\mathcal{S}$ is rather sophisticated, so that a *concrete* solution is a nice challenge.

Problem 26 *Does there exist an* algorithm **SHT** *(same homotopy type):*

$$\mathbf{SHT} : \mathcal{CC}'_{\mathcal{S}} \times \mathcal{CC}'_{\mathcal{S}} \to \mathbf{Bool}$$

allowing to decide whether two $\mathcal{CC}_{\mathcal{S}}$-objects obtained through the **SSC***-algorithms, therefore certainly corresponding to actual recursive simplicial sets, have the same homotopy type or not, of course up to some given dimension?*

The authors are not sufficiently experienced in operadic techniques to estimate the difficulty of this question. The Main Theorem of [9, p. 1] seems to imply that the same considerations as for Problem 25 could be applied; but as already observed, an effective solution of Problem 26 would solve indirectly hard and interesting computability problems in arithmetic, problems which seem to raise essential obstacles in front of the professionals. It is difficult to think the E_∞-operad could be a mandatory tool to solve these arithmetical problems, so that for a concrete solution it is more tempting to solve directly the arithmetical problems and to use *only* the category $\mathcal{SS}_{EH}$ and its satellite category $\mathcal{P}$, thanks to Theorems 21 and 22, to obtain a solution of the hard problem.

If the operadic methods become unavoidable, it seems terribly difficult to design *directly* the category $\mathcal{CC}_{\mathcal{S}}$ as a *computable* category. We think it would be more sensible to work *simultaneously* with the categories $\mathcal{SS}_{EH}$ and $\mathcal{CC}_{\mathcal{S}}$: it is frequent in mathematics in general, and in computer science in particular, that it is not a good idea to give up too early informations which *look* redundant. This is well known for example by the theoreticians in homotopy theory: it is much better to work with an *explicit* homotopy equivalence than only with the *existence* of such an object, and it is still better to keep also the various maps which describe how this homotopy equivalence actually is one, and so on. This is nothing but the philosophy always underlying when we work with E_∞-operads.

In our situation, Theorem 23 implies a simplicial set with effective homology X_{EH} *contains* in an effective way a Steenrod chain complex; and we do not need any realizibility criterion, an object of $\mathcal{SS}_{EH}$ *certainly* corrresponds to a *true* topological space. Therefore the right objects to work with in Algebraic Topology could be the pairs $(X_{EH}, \Sigma^X_{\mathcal{S}})$ where the second component $\Sigma^X_{\mathcal{S}}$ is the $CB\mathcal{S}$-structure induced on EC^X_* by the canonical Steenrod structure on C_*X. Then, when a new object is constructed from such objects, the ingredients present in the second components could facilitate the computation of some parts of the constructed object, but others would certainly obtained much more easily thanks to the first components.

In a sense the success of the category $\mathcal{SS}_{EH}$ is already of this sort: instead of working only with a chain complex EC_*^X describing the homology of X, certainly in general non-sufficient for the planned computations, it is much better to work with X itself under its locally effective form, the only form which can be processed on a machine when X is not of finite type. The amazing fact is that this is sufficient to solve many computability problems, though this version of X *does not effectively* defines the mathematical object X, because of Gödel and his friends, see [17, Section 5.3]. The same people, helped by Matiyasevich, have also made impossible a universal solver of systems of polynomial $\mathbb{Z}$-equations, and after all, the hard problem is equivalent to a problem about such equations, so that we cannot even be sure, up to further information, a solution of the hard problem *exists*.

References

[1] J. F. Adams, Peter J. Hilton. *On the chain algebra of a loop space.* Commentarii Mathematici Helvetici, 1956, vol. 30, pp 305-330.

[2] Hans J. Baues. *Geometry of loop spaces and the cobar construction.* Memoirs of the American Mathematical Society, 1980, vol. 230.

[3] Hans J. Baues. *Homotopy types.* in [8], pp 1-72.

[4] Clemens Berger and Benoit Fresse. *Combinatorial operad actions on cochains.* Preprint.

[5] Gunnar Carlsson and R. James Milgram. *Stable homotopy and iterated loop spaces.* in [8], pp 505-583.

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

[7] *Encyclopedic Dictionary of Mathematics*, sub-article *Postnikov complexes*, in different articles according to the edition (look for *Postnikov complex* in the final index). Mathematical Society of Japan and American Mathematical Society.

[8] *Handbook of Algebraic Topology* (Edited by I.M. James). North-Holland (1995).

[9] M. Mandell. *E_∞-algebras and p-adic homotopy theory.* Topology, 2001, vol. 40, pp 43-94.

[10] J. Peter May. *Simplicial objects in algebraic topology.* Van Nostrand, 1967.

[11] John McCleary. *User's guide to spectral sequences.* Publish or Perish, Wilmington DE, 1985.

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

[13] David E. Rydeheard, Rod M. Burstall. *Computational Category Theory.* Prentice Hall, 1988.

[14] Rolf Schön. *Effective algebraic topology.* Memoirs of the American Mathematical Society, 1991, vol. 451.

[15] Francis Sergeraert. *The computability problem in algebraic topology.* Advances in Mathematics, 1994, vol. 104, pp 1-29.

[16] Francis Sergeraert. *Effective homology, a survey.* `www-fourier.ujf-grenoble.fr/~sergerar/Papers/survey.dvi or ps`.

[17] Francis Sergeraert. *$\maltese_K$, objet du 3^e type.* Gazette des Mathématiciens, 2000, vol. 86, pp 29-45.

[18] Francis Sergeraert. `www-fourier.ujf-grenoble.fr/~sergerar/Papers/`.

[19] Justin R. Smith. *Iterating the cobar construction.* Memoirs of the American Mathematical Society, 1994, vol. 524.

[20] A.S. Troelstra, D. van Dalen. *Constructivism in mathematics, an introduction.* North-Holland, 1988.

JR : Depto Mat. y Comp.
Univ. de la Rioja
26004 Logroño, La Rioja
Spain

`Julio.Rubio@dmc.unirioja.es`

FS : Institut Fourier
Laboratoire de Mathématiques
UMR5582 (Ujf-Cnrs)
BP74
38402 St Martin d'Hères Cedex
France

`Francis.Sergeraert@ujf-grenoble.fr`