Incidence simplicial matrices formalized in Coq/SSReflect

Jónathan Heras, María Poza, Maxime Denes, Laurence Rideau

Incidence simplicial matrices formalized in Coq/SSReflect*

Jónathan Heras[1], María Poza[1], Maxime Dénès[2], and Laurence Rideau[2]

[1]Department of Mathematics and Computer Science of University of La Rioja
[1]{jonathan.heras,maria.poza}@unirioja.es
[2]INRIA Sophia Antipolis - Méditerranée
[2]{Maxime.Denes,Laurence.Rideau}@inria.fr

June 24, 2011

Abstract

Simplicial complexes are at the heart of Computational Algebraic Topology, since they give a concrete, combinatorial description of otherwise rather abstract objects which makes many important topological computations possible. The whole theory has many applications such as coding theory, robotics or digital image analysis. In this paper we present a formalization in the Coq theorem prover of simplicial complexes and their incidence matrices as well as the main theorem that gives meaning to the definition of homology groups and is a first step towards their computation.

1 Introduction

Algebraic Topology is a vast and complex subject, in particular mixing Algebra and (combinatorial) Topology. Algebraic Topology consists of trying to use as much as possible "algebraic" methods to attack topological problems. For instance, one can define some special groups associated with a topological space, in a way that respects the relation of homeomorphism of spaces. This allows one to study properties about topological spaces by means of statements about groups, which are often easier to prove.

However, in spite of being an abstract mathematical subject, Algebraic Topology methods can be implemented in software systems and then applied to different contexts such as coding theory [Woo89], robotics [Mac03] or digital image analysis [GDMRSP05, GDR05] (in this last case, in particular in the study of medical images [SGF03]). Nevertheless, if we want to use these systems in real life problems, we have to be completely sure that the systems are correct. Therefore, to increase the reliability of these methods and the systems that implement them, we can use Theorem Proving tools. In this paper we are going to focus on the verification of some results about a mathematical structure which can be useful, among others things, to study properties of digital images.

Simplicial complexes are topological abstract structures which provide a good framework to apply topological methods to analyse digital images. Intuitively, a simplicial complex is a generalization of the notion of graph to higher dimensions. Indeed, all the simplicial complexes of dimension less than two are graphs.

A central problem in this context consists of computing homology groups of simplicial complexes. Homology groups characterize both the number and the type of holes and the number of connected components of a simplicial complex. This type of information is used, for instance, to determine similarities between proteins in molecular biology [DEG99].

In the context of the computation of homology groups, we can highlight the Kenzo program [DSS98], a successful Computer Algebra system, implemented in Common Lisp, which has obtained some homology groups not confirmed nor refuted by any other means.

There are two different ways of computing homology groups in Kenzo depending on the type of the object. On the one hand, the task of calculating homology groups of a *finite object* is translated to a problem

*Partially supported by Ministerio de Educación y Ciencia, project MTM2009-13842-C02-01, and by European Community FP7, STREP project ForMath.

of diagonalizing certain matrices called *incidence matrices*, see [Veb31]. On the other hand, in the case of *non-finite type* objects, Sergeraert's effective homology [RS06] theory, implemented in Kenzo, provides a framework where this question can be handled. Roughly speaking, the effective homology method links a non-finite type object, X, with a finite type object, Y, with the same homology groups; then the problem of computing the homology groups of X is reduced to the task of diagonalizing the *incidence matrices* of Y.

Sergeraert's ideas have been translated to theorem provers with the aim of not only formalizing the effective homology theory, but also applying formal methods to the study of Kenzo. Thus far, the main formalization efforts have been focused on theorems which provide the connection between non-finite type objects with finite type ones; here, we can distinguish the verification of the Basic Perturbation Lemma in the Isabelle/HOL proof assistant, see [ABR08], or the formalization in Coq of the Effective Homology of Bicomplexes, see [DR].

However, up to now, the question of formalizing the computation of homology groups of finite objects has not been undertaken. In this paper we discuss the formalization of simplicial complexes and their incidence matrices as well as the main theorem that gives meaning to the definition of homology groups. To this aim, we have used the proof assistant Coq [tdt10, BC04] as well as the SSReflect extension [GM09] and the libraries it provides.

The rest of the paper is organized as follows. Section 2 contains some preliminaries on Algebraic Topology. A sketch of the proof of the main theorem is presented in Section 3. A brief introduction to SSReflect is provided in Section 4. The main steps of the formalization are given in Section 5. The paper ends with a section of Conclusions and Further Work, and the bibliography.

2 Mathematical preliminaries

In this section, we briefly provide the minimal standard background needed in the rest of the paper. We mainly focus on definitions. Many good textbooks are available for these definitions and results about them, the main one being maybe [Mac63].

The notion of simplicial complex gives rise to the most elementary method to settle a connection between common Topology and Algebraic Topology. The notion of topological space is too *abstract* to perform computations. Simplicial complexes provide a purely combinatorial description of topological spaces which admit a triangulation. The computability of properties, such as homology groups, from a simplicial complex associated with a topological space is well-known and the algorithm uses simple linear algebra [Veb31]. Then, an algebraic topologist can decide every sensible space (that is to say, a topological space which admit a triangulation) is a simplicial complex, making computations easier.

Let us start with some basic terminology. Let V be an ordered set, called the *vertex set*. An *(abstract) simplex* over V is any finite subset of V. An *(abstract) n-simplex* over V is a simplex over V whose cardinality is equal to $n+1$. Given a simplex α over V, we call subsets of α *faces* of α.

Definition 1 An *(ordered abstract) simplicial complex* over V is a set of simplices $\mathscr{K}$ over V such that it is closed by taking faces (subsets); that is to say, if $\alpha \in \mathscr{K}$ all the faces of α are in $\mathscr{K}$, too.

Let $\mathscr{K}$ be a simplicial complex. Then the set $S_n(\mathscr{K})$ of n-simplices of $\mathscr{K}$ is the set made of the simplices of cardinality $n+1$.

Example 1 Let us consider $V = (0,1,2,3,4,5,6)$.

The small simplicial complex drawn in Figure 1 is mathematically defined as the object:

$$\mathscr{K} = \left\{ \begin{array}{l} \emptyset,(0),(1),(2),(3),(4),(5),(6),(0,1),(0,2),(0,3),(1,2), \\ (1,3),(2,3),(3,4),(4,5),(4,6),(5,6),(0,1,2),(4,5,6) \end{array} \right\}$$

It is worth noting that simplicial complexes can be infinite. For instance if $V = \mathbb{N}$ and the simplicial complex $\mathscr{K}$ is $\{(n)\}_{n\in\mathbb{N}} \cup \{(0,n)\}_{n\geq 1}$, the simplicial complex obtained can be seen as an infinite bunch of segments.

Definition 2 A *facet of a simplicial complex $\mathscr{K}$ over V is a maximal simplex with respect to the subset order $\subseteq$ among the simplexes of $\mathscr{K}$.*

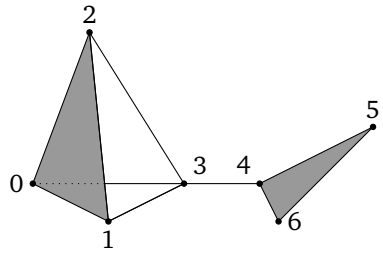

Figure 1: Butterfly Simplicial Complex

Example 2 The facets of the simplicial complex depicted in Figure 1 are:

$$\{(0,3),(1,3),(2,3),(3,4),(0,1,2),(4,5,6)\}$$

To construct the simplicial complex associated with a sequence of facets, $\mathscr{F}$, we generate all the faces of the simplexes of $\mathscr{F}$. Subsequently, if we perform the set union of all the faces we obtain the simplicial complex associated with $\mathscr{F}$.

Definition 3 Let $\mathscr{K}$ be a simplicial complex over V. Let n and i be two integers such that $n \geq 1$ and $0 \leq i \leq n$. Then the *face operator* ∂_i^n is the linear map $\partial_i^n : S_n(\mathscr{K}) \to S_{n-1}(\mathscr{K})$ defined by:

$$\partial_i^n((v_0,\ldots,v_n)) = (v_0,\ldots,v_{i-1},v_{i+1},\ldots,v_n)$$

the i-th vertex of the simplex is removed, so that an $(n-1)$-simplex is obtained.

Now, we are going to introduce a central notion in Algebraic Topology. We assume as known the notions of ring, module over a ring and module morphism (see [Jac89] for details).

Definition 4 Given a ring R, a *graded module* M is a family of left R-modules $(M_n)_{n\in\mathbb{Z}}$.

Definition 5 Given a pair of graded modules M and M', a *graded module morphism* f of degree k between them is a family of module morphisms $(f_n)_{n\in\mathbb{Z}}$ such that $f_n : M_n \to M'_{n+k}$ for all $n \in \mathbb{Z}$.

Definition 6 Given a graded module M, a *differential* $(d_n)_{n\in\mathbb{Z}}$ is a family of module endomorphisms of M of degree -1 such that $d_{n-1} \circ d_n = 0$ for all $n \in \mathbb{Z}$.

The previous definitions define a graded structure and a way of going from a level of the structure to the inferior one. From the previous definitions, the notion of chain complex is defined as follows.

Definition 7 A *chain complex* C_* is a family of pairs $(C_n, d_n)_{n\in\mathbb{Z}}$ where $(C_n)_{n\in\mathbb{Z}}$ is a graded module and $(d_n)_{n\in\mathbb{Z}}$ is a differential on $(C_n)_{n\in\mathbb{Z}}$.

The module C_n is called the module of *n-chains*. The image $B_n = im\ d_{n+1} \subseteq C_n$ is the (sub)module of n-boundaries. The kernel $Z_n = ker\ d_n \subseteq C_n$ is the (sub)module of *n-cycles*.

Given a chain complex $C_* = (C_n, d_n)_{n\in\mathbb{Z}}$, the identities $d_{n-1} \circ d_n = 0$ are equivalent to the inclusion relations $B_n \subseteq Z_n$: every boundary is a cycle but the converse is not generally true. Thus, the next definition makes sense.

Definition 8 Let $C_* = (C_n, d_n)_{n\in\mathbb{Z}}$ be a chain complex of R-modules. For each degree $n \in \mathbb{Z}$, the *n-homology module* of C_* is defined as the quotient module

$$H_n(C_*) = \frac{Z_n}{B_n}$$

Once we have defined the notions of simplicial complexes and chain complexes, we can define the link between them considering $\mathbb{Z}$ as the ring R; the most common case in Algebraic Topology.

Definition 9 Let $\mathscr{K}$ be a simplicial complex over V. Then *the chain complex $C_*(\mathscr{K})$ canonically associated with $\mathscr{K}$* is defined as follows. The chain group $C_n(\mathscr{K})$ is the free $\mathbb{Z}$ module generated by the n-simplices of $\mathscr{K}$. In addition, let $(v_0,\ldots,v_n)$ be an n-simplex of $\mathscr{K}$, the differential of this simplex is defined as:

$$d_n := \sum_{i=0}^{n}(-1)^i \partial_i^n$$

In order to clarify the notion of chain complex canonically associated with a simplicial complex, let us present an example. The chain complexes associated with simplicial complexes are good candidates for this purpose.

Example 3 Let $\mathscr{K}$ be the simplicial complex defined in Figure 1. The chain complex $C_*(\mathscr{K})$ canonically associated with $\mathscr{K}$ is:

$$\cdots \to 0 \to C_2(\mathscr{K}) \xrightarrow{d_2} C_1(\mathscr{K}) \xrightarrow{d_1} C_0(\mathscr{K}) \to 0 \to \cdots$$

where there are 3 associated chain groups:

- $C_0(\mathscr{K})$, the free $\mathbb{Z}$-module on the set of 0-simplexes (vertices) $\{(0),(1),(2),(3),(4),(5),(6)\}$.
- $C_1(\mathscr{K})$, the free $\mathbb{Z}$-module on the set of 1-simplexes (edges) $\{(0,1),(0,2),(0,3),(1,2),(1,3),(2,3),(3,4),(4,5),(4,6),(5,6)\}$.
- $C_2(\mathscr{K})$, the free $\mathbb{Z}$-module on the set of 2-simplexes (triangles) $\{(0,1,2),(4,5,6)\}$.

The elements of either of those groups C_p are linear integer combinations of the corresponding basis (set of σ_i's), i.e. elements of the form $\sum \lambda_i\sigma_i$, $\lambda_i \in \mathbb{Z}$.

The differential homomorphism is in this case:

$$d_n((v_0,\ldots,v_n)) := \sum_{i=0}^{n}(-1)^i(v_0,\ldots,v_{i-1},v_{i+1},\ldots,v_n) \tag{1}$$

For instance, $d_2((0,1,2)) = (1,2)-(0,2)+(0,1)$.

From the previous definition, we can introduce a very useful concept for the computation of homology groups of simplicial complexes.

Definition 10 Let $\mathscr{K}$ be a simplicial complex over V and let n be an integer such that $n \geq 1$. The *n-th incidence matrix of $\mathscr{K}$* over the ring $\mathbb{Z}$, denoted by $M_n(\mathscr{K},\mathbb{Z})$, represents the $(n-1)$-simplices of $\mathscr{K}$ as rows and the n-simplices of $\mathscr{K}$ as columns. Assuming an ordering on the simplices of the same dimension (in the rest of the paper we assume that the simplices of the same dimension will be ordered), $M_n(\mathscr{K},\mathbb{Z})$ is $[a_i^j]$ where i ranges from 1 to the cardinality of $S_{n-1}(\mathscr{K})$, j ranges from 1 to the cardinality of $S_n(\mathscr{K})$ and the value of a_i^j is the coefficient of the i-th $(n-1)$-simplex in the differential of the j-th n-simplex; then a_i^j is a value in $\{0,\pm 1\}$.

Example 4 If we impose a lexicographical order on the simplices of the same dimension of the simplicial complex depicted in Figure 1 (if $v=(a_0,\ldots,a_n)$ and $w=(b_0,\ldots,b_n)$ are n-simplices of the simplicial complex, then $v<w$ if $a_0<b_0$, or $a_0=b_0$ and $a_1<b_1$, or $a_0=b_0$ and $a_1=b_1$ and $a_2<b_2$,..., or $a_0=b_0,\ldots a_{n-1}=b_{n-1}$ and $a_n<b_n$), then its first incidence matrix is:

$$\begin{array}{c} \\ (0) \\ (1) \\ (2) \\ (3) \\ (4) \\ (5) \\ (6) \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} (0,1) & (0,2) & (0,3) & (1,2) & (1,3) & (2,3) & (3,4) & (4,5) & (4,6) & (5,6) \end{array} \\
\begin{pmatrix}
-1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}
\end{array}$$

The relevance of the incidence matrices of simplicial complexes lies in the fact that they can be used to compute the homology groups of the simplicial complex by means of a diagonalization process, as explained for instance in [Veb31].

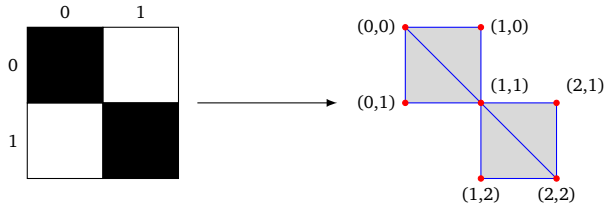

Figure 2: A digital image and its simplicial complex representation

3 The theorem formalized and its context

The definitions presented in the previous section are classical definitions from Algebraic Topology. However, since our final goal consists of working with mathematical objects coming from digital images, let us show how this machinery from algebraic topology may be used in this context.

It is worth noting that there are several methods to construct a simplicial complex from a digital image [ADFQ03]. We are going to explain one of these methods. Roughly speaking, the chosen method consists of obtaining a sequence of facets from a digital image. Then, as we have explained in the previous section, we can obtain the simplicial complex associated with the facets. So, we only need to explain how to get the facets from a digital image.

We are going to work with monochromatic two dimensional images. An image can be represented by a finite 2-dimensional array of 1's and 0's in which the black pixels are represented by 1's and white pixels are represented by 0's.

Let $\mathscr{I}$ be an image codified as a 2-dimensional array of 1's and 0's. Let $V = (\mathbb{N}, \mathbb{N})$ be the vertex set, each vertex is a pair of natural numbers. Let $p = (a, b)$ be the coordinates of a black pixel in $\mathscr{I}$. For each p we can obtain two 2-simplexes which are two facets of the simplicial complex associated with $\mathscr{I}$. Namely, for each $p = (a, b)$ we obtain the following facets: $((a, b), (a+1, b), (a+1, b+1))$ and $((a, b), (a, b+1), (a+1, b+1))$. If we repeat the process for the coordinates of all the black pixels in $\mathscr{I}$, we obtain the facets of a simplicial complex associated with $\mathscr{I}$, let us called it $\mathscr{K}_{\mathscr{I}}$.

Example 5 Consider the image depicted in Figure 2. This image, $\mathscr{I}$, can be codified by means of the 2-dimensional array: ((1,0),(0,1)). Then, with the previously explained process we obtain the facets of $\mathscr{K}_{\mathscr{I}}$. The coordinates of the black pixels are $(0, 0)$ and $(1, 1)$, so the facets that we obtain are:

$$(((0,0),(1,0),(1,1)),((0,0),(0,1),(1,1)),((1,1),(2,1),(2,2)),((1,1),(1,2),(2,2)))$$

We have presented a method to obtain a simplicial complex associated with a 2D-image, this process can be generalized to higher-dimensional images [OS03].

It is worth noting that even the bigger digital images have always a finite number of components, hence a finite number of vertices and then our vertex set V consists of a finite number of vertices. Therefore, the simplicial complexes coming from digital images are always of finite type. This point will be important in our formalization.

Moreover, instead of working with the ring $\mathbb{Z}$, we consider the ring $\mathbb{Z}/2\mathbb{Z}$ since the computation of homology groups is easier working with $\mathbb{Z}/2\mathbb{Z}$. This approach is usually followed when algebraic topology methods are applied to the study of digital images, see [GDMRSP05, GDR05].

Then, we are going to work with a different definition of the face operator and associated incidence matrices. Indeed, since coefficients (in $\mathbb{Z}/2\mathbb{Z}$) of opposite sign are identified, we do not have to deal with orientations of faces.

Thus, in the following $\mathscr{K}$ will denote a simplicial complex over a finite set V and n an integer such that $n \geq 1$. The incidence matrix is now defined by:

Definition 11 The *n-th incidence matrix of* $\mathscr{K}$ over the ring $\mathbb{Z}/2\mathbb{Z}$, denoted by $M_n(\mathscr{K})$, is a matrix of size $m \times p$, where m is the cardinality of $S_{n-1}(\mathscr{K})$ and p is cardinality of $S_n(\mathscr{K})$. Its coefficients $[a_i^j]$ are 1 if the i-th $(n-1)$-simplex is a face of the j-th n-simplex and 0 otherwise.

Note that the *n-th incidence matrix of* $\mathscr{K}$ over the ring $\mathbb{Z}/2\mathbb{Z}$ is the absolute value of the *n-th incidence matrix of* $\mathscr{K}$ over the ring $\mathbb{Z}$.

Using this definition of incidence matrices, it is not necessary to use chain complexes to compute homology groups of simplicial complexes, but just applying a diagonalization process, as described in [Veb31].

Example 6 If we impose a lexicographical order on the simplices of the same dimension of the simplicial complex depicted in Figure 1, then its first incidence matrix over the ring $\mathbb{Z}/2\mathbb{Z}$ is:

$$\begin{array}{c} \\ (0) \\ (1) \\ (2) \\ (3) \\ (4) \\ (5) \\ (6) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (0,1) & (0,2) & (0,3) & (1,2) & (1,3) & (2,3) & (3,4) & (4,5) & (4,6) & (5,6) \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

As we have said previously, incidence matrices of simplicial complexes come from the differentials of the chain complexes canonically associated with the simplicial complexes. Theses differentials satisfy a nilpotency condition ($d_{n-1} \circ d_n = 0$).

Then, we can state and proof the following theorem that is analogous to this nilpotency condition on the incidence matrices we have defined above. It should be noted that the statement below is the immediate transcription of the one we formalized and proved in Coq/SSReflect.

Theorem 1 The product of the n-th incidence matrix of $\mathscr{K}$ over the ring $\mathbb{Z}/2\mathbb{Z}$, $M_n(\mathscr{K})$, and the $(n+1)$-incidence matrix of $\mathscr{K}$ over the ring $\mathbb{Z}/2\mathbb{Z}$, $M_{n+1}(\mathscr{K})$ is equal to the null matrix.

Sketch of the proof. Let S_{n-1}, S_n, S_{n+1} be the set of $(n-1)$-simplices of $\mathscr{K}$, the set of n-simplices of $\mathscr{K}$ and the set of $(n+1)$-simplices of $\mathscr{K}$ respectively. Then,

$$M_n(\mathscr{K}) = \begin{array}{c} \\ S_{n-1}[1] \\ \vdots \\ S_{n-1}[r2] \end{array} \begin{array}{c} \begin{array}{ccc} S_n[1] & \cdots & S_n[r1] \end{array} \\ \begin{pmatrix} a_{1,1} & \cdots & a_{1,r1} \\ \vdots & \ddots & \vdots \\ a_{r2,1} & \cdots & a_{r2,r1} \end{pmatrix} \end{array}, M_{n+1}(\mathscr{K}) = \begin{array}{c} \\ S_n[1] \\ \vdots \\ S_n[r1] \end{array} \begin{array}{c} \begin{array}{ccc} S_{n+1}[1] & \cdots & S_{n+1}[r3] \end{array} \\ \begin{pmatrix} b_{1,1} & \cdots & b_{1,r1} \\ \vdots & \ddots & \vdots \\ b_{r1,1} & \cdots & b_{r1,r3} \end{pmatrix} \end{array}$$

where $r1 = \sharp|S_n|$, $r2 = \sharp|S_{n-1}|$ and $r3 = \sharp|S_{n+1}|$. Thus,

$$M_n(\mathscr{K}) \times M_{n+1}(\mathscr{K}) = \begin{pmatrix} c_{1,1} & \cdots & c_{1,r3} \\ \vdots & \ddots & \vdots \\ c_{r2,1} & \cdots & c_{r2,r3} \end{pmatrix} \text{ where } c_{i,j} = \sum_{1 \leq k \leq r1} a_{i,k} \times b_{k,j}$$

To prove that $M_n \times M_{n+1}$ is equal to the null matrix, it is enough to prove that $\forall i, j$ such that $1 \leq i \leq \sharp|S_{n-1}|$ and $1 \leq j \leq \sharp|S_{n+1}|$, then $c_{i,j} = 0$. Each of these coefficients is written:

$$c_{i,j} = \sum_{1 \leq k \leq r1} a_{i,k} \times b_{k,j}$$

Since k enumerates the indices of elements of S_n, we may write:

$$c_{i,j} = \sum_{X \in S_n} F(S_{n-1}[i], X) \times F(X, S_{n+1}[j]) \text{ with } F(Y,Z) = \begin{cases} 1 & \text{if } Y \in dZ \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

dZ is the analogous in our context of the differential operator defined by (1) and is equal to:

$$dZ = \{Z \setminus \{x\} \mid x \in Z\}$$

This summation can be split depending on whether $X \in \partial S_{n+1}[j]$ or $X \notin \partial S_{n+1}[j]$.

$$\begin{aligned} c_{i,j} &= \sum_{X \in S_n | X \in \partial S_{n+1}[j]} F(S_{n-1}[i], X) \times 1 && (3) \\ &\quad + \sum_{X \in S_n | X \notin \partial S_{n+1}[j]} F(S_{n-1}[i], X) \times 0 \\ &= \sum_{X \in S_n | X \in \partial S_{n+1}[j]} F(S_{n-1}[i], X) && (4) \end{aligned}$$

The last summation is expressed over the image of the face operator $x \mapsto S_{n+1}[j] \setminus \{x\}$ which, restricted to $S_{n+1}[j]$, is injective. Thus, we can reindex:

$$c_{i,j} = \sum_{x \in S_{n+1}[j]} F(S_{n-1}[i], S_{n+1}[j] \setminus \{x\}) \tag{5}$$

Subsequently, this summation can also be split depending on whether $x \in S_{n-1}[i]$ or $x \notin S_{n-1}[i]$.

$$\begin{aligned} c_{i,j} \quad = \quad & \sum_{x \in S_{n+1}[j] | x \in S_{n-1}[i]} F(S_{n-1}[i], S_{n+1}[j] \setminus \{x\}) + \\ & \sum_{x \in S_{n+1}[j] | x \notin S_{n-1}[i]} F(S_{n-1}[i], S_{n+1}[j] \setminus \{x\}) \end{aligned} \tag{6}$$

Let us note that if $x \in S_{n-1}[i]$ then $S_{n-1}[i] \not\subset S_{n+1}[j] \setminus \{x\}$, hence the first sum above is 0.

$$c_{i,j} = \sum_{x \in S_{n+1}[j] | x \notin S_{n-1}[i]} F(S_{n-1}[i], S_{n+1}[j] \setminus \{x\}) \tag{7}$$

Here, we can split our proof considering two cases: $S_{n-1}[i] \not\subset S_{n+1}[j]$ and $S_{n-1}[i] \subset S_{n+1}[j]$.

In the first case, we have: $\forall x \in S_{n-1}[i], F(S_{n-1}[i], S_{n+1}[j] \setminus \{x\}) = 0$, hence the result holds.

In the second case, $S_{n-1}[i] \subset S_{n+1}[j]$ implies that if $x \notin S_{n-1}[i]$ then $S_{n-1}[i] \in \partial S_{n+1}[j] \setminus \{x\}$, so the terms are all 1.

$$\begin{aligned} c_{i,j} \quad &= \quad \sum_{x \in S_{n+1}[j] | x \notin S_{n-1}[i]} 1 \\ &= \quad \sharp |S_{n+1}[j] \setminus S_{n-1}[i]| \\ &= \quad n + 2 - n = 2 = 0 \bmod 2 \end{aligned} \tag{8}$$

4 SSReflect basics

To formalize Theorem 1, we have used SSReflect [GM09], an extension for the Coq proof assistant [BC04, tdt10]. Its development was started by G. Gonthier during the formal proof of the Four Color Theorem [Gon08] and is now involved in the formalisation of the Feit-Thompson theorem [Mat].

SSReflect (for Small Scale Reflection) introduces a new language for tactics that eases the development of proof scripts. Another main feature is the generic reflection mechanism. More details on the SSReflect tactics language and reflection techniques are presented in its manual [GM09].

Moreover, SSReflect provides a set of libraries embedding definitions and properties for a variety of mathematical structures. In our formalization, it is worth mentioning the following libraries:

- matrix.v: this library formalizes matrix theory, determinant theory and matrix decompositions. In our problem, this library is used to define incidence matrices, and to state and prove Theorem 1.
- finset.v and fintype.v: theory of finite sets and finite types. We use these libraries to define the basic concepts about simplicial complexes.
- bigop.v: generic indexed "big" operations, like $\sum_{i=0}^{n} f(i)$ or $\bigcup_{i \in I} f(i)$ and their properties, which are useful to deal with the matricial product in Theorem 1.
- zmodp.v: additive group and ring $\mathbb{Z}_p$, together with field properties when p is a prime. As we work with elements of the field $\mathbb{F}_2$, we need this library.

For more precise details on these libraries we refer to [BGBP08, GMR+07]

5 Formal development

The SSReflect libraries include all the necessary ingredients to represent the mathematical structures of our formalization.

First of all, we define the notions related to simplicial complexes. The vertices are represented by a finite type `V`. A simplex is defined as a finite set of vertices. Then, the definition of a simplicial complex as a set of simplices closed under inclusion is straightforward:

```
Variable V : finType.
Definition simplex := {set V}.
Definition simplicial_complex (c : {set simplex}) :=
  forall x, x \in c -> forall y : simplex, y \subset x -> y \in c.
```

Since we do not take into account the signs of the coefficients appearing in the incidence matrices, we define a face operator as a set difference (we remove a vertex from a simplex) and the boundary as the image of a simplex by the face operator.

```
Definition face_op (S : simplex) (x : V) := S :\ x.
Definition boundary (S : simplex) := (face_op S) @: S.
```

We prove the correctness of our definition of boundary by showing it is equivalent to a subset relation with constraints on cardinality:

```
Lemma boundaryP: forall (S : simplex) (B : simplex),
  reflect (B \subset S /\ #|S| = #|B|.+1) (B \in boundary S).
```

The statement `reflect P b` expresses an equivalence between a proposition `P` and a boolean expression `b`. This allows to take advantage of the decidability of some propositions by going back and forth from their logical expressions (useful for reasoning) to their boolean counterparts (well suited for computations).

A key argument for our proof is the injectivity of the face operator above, which we establish as a lemma:

```
Lemma face_op_inj2: forall (S : simplex),
  {in S &, injective (face_op S)}.
```

The notation `{in S &, P}` performs localization of predicates: if `P` is of the form `forall x y, Qxy` then `{in S &, P}` means `forall x y, x \in S -> y \in S -> Qxy`. In our case, `injective f` stands for `forall x y, f x = f y -> x = y`.

Now, before giving the definition of the n-th incidence matrix of a simplicial complex, we can define the more generic notion of incidence matrix of two finite sets of simplices.

Representing a matrix requires an indexing of the simplices in `Left` (for the rows) and `Top` (for the columns). Since `Left` and `Top` are finite sets, they are equipped with a canonical enumeration: `(enum_val Left i)` returns the i-th element of the set `Left`. A coefficient a_{ij} of the incidence matrix will be 1 if the i-th simplex of `Left` is a face (subset) of the j-th simplex of `Top` and 0 otherwise.

Thus we can define the incidence matrix of two finite sets of simplices as follows:

```
Variables Left Top : {set simplex}.
Definition incidenceMatrix :=
  \matrix_(i < #|Left|, j < #|Top|)
     if enum_val i \in boundary (enum_val j) then 1 else 0:'F_2.
```

In the definition above, it can be noted that the first argument of `enum_val` is implicit and determined by the context. Indeed, the notation `i < #|Left|` means that the type of `i` is `'I_(#|Left|)`, that is `i` is an ordinal ranging from 0 to `#|Left|`-1, where `#|X|` denotes the cardinal of the set `X`. With this type information, the system expands `enum_val i` to `enum_val Left i`, thus resolving the ambiguity (and similarly for `j`).

The type annotation `0:'F_2` indicates that the 0 and 1 appearing as coefficients of the matrix are the two elements of $\mathbb{F}_2$, that is $\mathbb{Z}/2\mathbb{Z}$ as a field.

We now define the `n`-th incidence matrix of a simplicial complex `c`, by instantiating Left to the set of $n-1$-simplices (of c) and Top to the set of n-simplices. Note that n should be nonzero.

```
Section nth_incidence_matrix.
Variable c: {set simplex}.
Variable n:nat.
```

```
Definition n_1_simplices := [set x \in c | #|x| == n].
Definition n_simplices := [set x \in c | #|x| == n+1].
Definition incidence_matrix_n :=
  incidenceMatrix n_1_simplices n_simplices.
End nth_incidence_matrix.
```

Then we have all the ingredients to state Theorem 1:

```
Theorem incidence_matrices_sc_product:
  forall (V:finType) (n:nat) (sc: {set (simplex V)}),
    simplicial_complex sc ->
      (incidence_mx_n sc n) *m (incidence_mx_n sc (n.+1)) = 0.
```

In the statement above, `*m` denotes the matricial product. The type information of each matrix includes its size. When the product operator is applied, the typechecking ensures that the two arguments have compatible sizes. Then the system knows the expected size of the result matrix and reads 0 as the null matrix of this size.

The formal proof of Theorem 1 follows the schema presented in Section 3. A large part of the proof is devoted to the work with summations, for which the Coq/SSReflect library "bigop" has played a key role.

For instance, the first summation splitting (equation (3)) is realized by:

```
rewrite (bigID (mem (boundary (enum_val j)))).
```

where j belongs to S_{n+1}.

The lemma `bigID` states that an iterated operation using a commutative monoidal operator can be split:

$$\sum_{i \in r | P_i} F_i = \sum_{i \in r | P_i \wedge a_i} F_i + \sum_{i \in r | P_i \wedge \sim a_i} F_i$$

It is also possible to split a summation (equation (6)) and at the same time rewrite the first resulting sum to 0 as in:

```
rewrite (bigID (mem (enum_val i))) big1.
```

`big1` states that, when a monoidal operator is iterated over elements that are all equal to the neutral, then the result is also the neutral element:

$$\sum_{i \in r | P_i} 0 = 0$$

Therefore, after the last tactic, the system will require a proof that all the terms of the first resulting summation are zero. `big1` is applied to obtain equations 4 and 7 of Section 3.

Our proof relies on two main reindexations: from ordinals to n-simplices (2) and later on from simplices to vertices (5). To perform the first reindexation, the script has the following shape:

```
rewrite (reindex_onto (enum_rank_in Hx0) enum_val) ; last first.
  by move=> x _ ; exact:enum_valK_in.
```

Where:

- Hx0 is a proof that there exists at least one n-simplex
- `enum_rank_in` enumerates the n-simplices since Hx0 ensures there is at least one
- `enum_val` enumerates the ordinals over which the sum is expressed
- `reindex_onto` reindexes from ordinals to n-simplices, given a bijection between both sets. Indeed, the second line proves that `enum_val` $\circ$ `enum_rank_in` $=$ Id

The second reindexation is based on the injectivity of the face operator:

```
rewrite big_imset ; last exact:face_op_inj2.
```

Rewriting with the lemma `big_imset` triggers a check that the summation is expressed over the image of a set by a function. In our case, the system automatically infers that this function is the face operator `face_op`, and will then ask for a proof of its injectivity.

The lemma `eq_big` and its variants `eq_bigl` and `eq_bigr` allow to rewrite the predicate or the operand of an iterated operation. It is applied in particular to obtain equation 8 of Section 3:

```
rewrite (eq_bigr (fun _ => 1)).
```

The system will of course require a proof that the operand is equal to 1. Then it will rewrite the expression to a constant summation, allowing the use of the lemma `big_const` to replace it with a product (cardinal of the iterated set by the constant value).

Simple arithmetic arguments on cardinals will then complete the proof. The interested reader will find a snapshot of our development online [HPDR10].

6 Conclusions and further work

In this paper we have presented the formalization of simplicial complexes and their incidence matrices as well as the main theorem that gives meaning to the definition of homology groups. The proof assistant used has been Coq as well as the SSReflect extension and the libraries it provides. The verified algorithm is related to a Computer Algebra system for Algebraic Topology called Kenzo [DSS98]. Therefore, our research is placed between the efforts to formalize mathematics and the application of formal methods in software systems.

Some parts of the future work are quite natural. The work presented here is solid enough to undertake the challenge of formalizing the construction of the Smith Normal Form [Veb31] of incidence matrices, that is the diagonalization process which obtains homology groups of finite type objects.

Moreover, if we want to apply our Algebraic Topology methods to real life problems, for instance the study of medical images, we must be completely sure that our programs are safe. Therefore, the process to construct a simplicial complex from a digital image, presented in Section 3, should be formalized, too.

In addition, our proof seems generic enough to achieve the case of working with $\mathbb{Z}$-modules, instead of $\mathbb{Z}/2\mathbb{Z}$-modules, quite easily.

Another topic is related to the executability of our proofs, that is the computational capabilities of the objects we have defined (like the incidence matrices). Two main approaches are possible: code extraction or internal computations. The first one delivers a certified program and takes advantage of the existing extraction machinery of the Coq system. However, technical limitations have to be dealt with to get a usable program in our context. The second approach is somewhat more challenging regarding efficiency. Indeed, reaching inside Coq an execution speed on par with the one obtained by extraction and compilation is difficult because proofs cannot safely be erased from the terms (what extraction does). However, compilation techniques and evaluation strategies mitigating the performance impact are currently being studied and implemented.

One advantage of this second approach lies in the fact that it would enable the reuse of computational results in further formal developments. For instance, the computation of the smith normal form of a matrix could be used for further deductions, in the same system, on the topological object under study. We are currently studying the use of both code extraction and efficient computational techniques in the Coq/SSReflect system, applied to the objects and theories we have presented above.

References

[ABR08] J. Aransay, C. Ballarin, and J. Rubio. A mechanized proof of the Basic Perturbation Lemma. *Journal of Automated Reasoning*, 40(4):271–292, 2008.

[ADFQ03] R. Ayala, E. Domínguez, A.R. Francés, and A. Quintero. Homotopy in digital spaces. *Discrete Applied Mathematics*, 125:3–24, 2003.

[BC04] Y. Bertot and P. Castéran. *Interactive Theorem Proving and Program Development, Coq'Art: the Calculus of Inductive Constructions*. Springer-Verlag, 2004.

[BGBP08] Y. Bertot, G. Gonthier, S.O. Biha, and I. Pasca. Canonical big operators. In *Theorem Proving in Higher-Order Logics (TPHOLS'08)*, volume 5170 of *Lecture Notes in Computer Science*, pages 86–101, 2008.

[DEG99] T. K. Dey, H. Edelsbrunner, and S. Guha. *Contemporary Mathematics*, volume 223, chapter Computational topology. Advances in Discrete and Computational Geometry, pages 190–143. AMS, Providence, 1999.

[DR] C. Domínguez and J. Rubio. Effective Homology of Bicomplexes, formalized in Coq. *To appear in Theoretical Computer Science*.

[DSS98] X. Dousson, F. Sergeraert, and Y. Siret. The Kenzo program. Institut Fourier, Grenoble, 1998. http://www-fourier.ujf-grenoble.fr/~sergerar/Kenzo.

[GDMRSP05] R. Gonzalez-Diaz, B. Medrano, P. Real, and J. Sanchez-Pelaez. Algebraic Topological Analysis of Time-Sequence of Digital Images. In *Proceedings 8th International Conference on Computer Algebra in Scientific Computing (CASC'2005)*, volume 3718 of *Lecture Notes in Computer Science*, pages 208–219, 2005.

[GDR05] R. Gonzalez-Diaz and P. Real. On the Cohomology of 3D Digital Images. *Discrete Applied Math*, 147(2-3):245–263, 2005.

[GM09] G. Gonthier and A. Mahboubi. A Small Scale Reflection Extension for the Coq system. Technical report, Microsoft Research INRIA, 2009. http://hal.inria.fr/inria-00258384.

[GMR+07] G. Gonthier, A. Mahboubi, L. Rideau, E. Tassi, and L. Théry. A modular formalisation of finite group theory. In *Theorem Proving in Higher-Order Logics (TPHOLS'07)*, volume 4732 of *Lecture Notes in Computer Science*, pages 86–101, 2007.

[Gon08] G. Gonthier. *Formal proof - The Four-Color Theorem*, volume 55. Notices of the American Mathematical Society, 2008.

[HPDR10] J. Heras, M. Poza, M. Dénès, and L. Rideau. Incidence simplicial matrices formalized in SSReflect, 2010. http://www.unirioja.es/cu/joheras/ismfissr/.

[Jac89] N. Jacobson. *Basic Algebra II*. W. H. Freeman and Company, 2nd edition, 1989.

[Mac63] S. MacLane. *Homology*. Springer, 1963.

[Mac03] D. Mackenzie. Topologists and Roboticists Explore and Inchoate World. *Science*, 8:756, 2003.

[Mat] Mathematical components team homepage. http://www.msr-inria.inria.fr/Projects/math-components.

[OS03] D. Orden and F. Santos. Asymptotically efficient triangulations of the d-cube. *Discrete and Computational Geometry*, 30(4):509–528, 2003.

[RS06] J. Rubio and F. Sergeraert. Constructive Homological Algebra and Applications, Lecture Notes Summer School on Mathematics, Algorithms, and Proofs. University of Genova, 2006. http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/Genova-Lecture-Notes.pdf.

[SGF03] F. Ségonne, E. Grimson, and B. Fischl. Topological Correction of Subcortical Segmentation. In *Proceedings 6th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'2003)*, volume 2879 of *Lecture Notes in Computer Science*, pages 695–702, 2003.

[tdt10] Coq development team. The Coq Proof Assistant Reference Manual, version 8.3. Technical report, 2010.

[Veb31] O. Veblen. *Analysis Situs*. AMS Coll. Publ., 1931.

[Woo89] J. Wood. Spinor groups and algebraic coding theory. *Journal of Combinatorial Theory*, 50:277–313, 1989.