Computing simplicial representatives of homotopy group elements*

Marek Filakovský
mfilakov@ist.ac.at

Peter Franek
peter.franek@gmail.com

Uli Wagner
uli@ist.ac.at

Stephan Zhechev
stephan.zhechev@ist.ac.at

IST Austria, Am Campus 1, 3400 Klosterneuburg, Austria

Abstract

A central problem of algebraic topology is to understand the *homotopy groups* $\pi_d(X)$ of a topological space X. For the computational version of the problem, it is well known that there is no algorithm to decide whether the *fundamental group* $\pi_1(X)$ of a given finite simplicial complex X is trivial. On the other hand, there are several algorithms that, given a finite simplicial complex X that is *simply connected* (i.e., with $\pi_1(X)$ trivial), compute the higher homotopy group $\pi_d(X)$ for any given $d \geq 2$.

However, these algorithms come with a caveat: They compute the isomorphism type of $\pi_d(X)$, $d \geq 2$ as an *abstract* finitely generated abelian group given by generators and relations, but they work with very implicit representations of the elements of $\pi_d(X)$. Converting elements of this abstract group into explicit geometric maps from the d-dimensional sphere S^d to X has been one of the main unsolved problems in the emerging field of computational homotopy theory.

Here we present an algorithm that, given a simply connected simplicial complex X, computes $\pi_d(X)$ and represents its elements as simplicial maps from a suitable triangulation of the d-sphere S^d to X. For fixed d, the algorithm runs in time exponential in size(X), the number of simplices of X. Moreover, we prove that this is optimal: For every fixed $d \geq 2$, we construct a family of simply connected simplicial complexes X such that for any simplicial map representing a generator of $\pi_d(X)$, the size of the triangulation of S^d on which the map is defined is exponential in size(X).

*The research leading to these results has received funding from Austrian Science Fund (FWF): M 1980.

1 Introduction

One of the central concepts in topology are the *homotopy groups* $\pi_d(X)$ of a topological space X. Similar to the *homology groups* $H_d(X)$, the homotopy groups $\pi_d(X)$ provide a mathematically precise way of measuring the "d-dimensional holes" in X, but the latter are significantly more subtle and computationally much less tractable than the former. Understanding homotopy groups has been one of the main challenges propelling research in algebraic topology, with only partial results so far despite an enormous effort (see, e.g., [37, 26]); the amazing complexity of the problem is illustrated by the fact that even for the 2-dimensional sphere S^2, the higher homotopy groups $\pi_d(S^2)$ are nontrivial for infinitely many d and *known* only for a few dozen values of d.

For computational purposes, we consider spaces that have a combinatorial description as *finite simplicial complexes*, and we represent maps between them as *simplicial maps*. (Actually, we state our results in terms of finite simplicial complexes mainly for the purposes of exposition; for the more technical parts of this work and the actual agorithms, we use the related but more flexible notion of *simplicial sets*, which will be discussed later.)

A fundamental computational result about homotopy groups is negative: There is no algorithm to decide whether the *fundamental group* $\pi_1(X)$ of a finite simplicial complex X is trivial, i.e., whether every continuous map from the circle S^1 to X can be continuously contracted to a point; this holds even if X is restricted to be 2-dimensional.[1]

On the other hand, given a simplicial complex X that is *simply connected* (i.e., path connected and with $\pi_1(X)$ trivial) there are algorithms that compute the higher homotopy group $\pi_d(X)$, for every given $d \geq 2$. The first such algorithm was given by Brown [5], and newer ones have been obtained as a part of general computational frameworks in algebraic topology; in particular, an algorithm based on the methods of Sergeraert et al. [46, 42] was described by Real [38].

More recently, Čadek et al. [9] proved that, for any fixed d, the homotopy group $\pi_d(X)$ of a given 1-connected finite simplicial complex can be computed in polynomial time. On the negative side, computing $\pi_d(X)$ is #P-hard if d is part of the input [2, 8] (and, moreover, W[1]-hard with respect to the parameter d [30]), even if X is restricted to be a 4-dimensional simplicial complex. These results form part of a general effort to understand the *computational complexity* of topological questions concerning the classification of maps up to *homotopy* (continuous deformation) [7, 6, 8, 15] and related questions, such as the *embeddability problem* for simplicial complexes (a higher-dimensional analogue of graph planarity) [29, 31, 10].

1.1 Our Results: Representing Homotopy Classes by Explicit Maps

By definition, elements of $\pi_d(X)$ are equivalence classes of continuous maps from the d-dimensional sphere S^d to X, with maps being considered equivalent (or lying in the same *homotopy class*) if they are *homotopic*, i.e., if they can be continuously deformed into one another (see Section 5 for more details).

The algorithms of [5] or [9] mentioned above compute $\pi_d(X)$ as an abstract abelian group, in terms of generators and relations.[2] However, they work with very implicit representations of the elements of $\pi_d(X)$.

The main result of this paper is an algorithm that, given an element α of $\pi_d(X)$,

[1] This follows via a standard reduction from a result of Adjan[1] and Rabin [36] on the algorithmic unsolvability of the triviality problem of a group given in terms of generators and relations; we refer to the survey [48] for further background.

[2] That is, they compute integers $r, q_1, \ldots, q_k$ such that $\pi_d(X)$ is isomorphic to $\mathbb{Z}^r \oplus \mathbb{Z}_{q_1} \oplus \ldots \oplus \mathbb{Z}_{q_k}$.

computes a suitable triangulation Σ^d of the sphere S^d and an explicit simplicial map $\Sigma^d \to X$ representing the given homotopy class α.

Apart from the intrinsic importance of homotopy groups, we see this as a first step towards the more general goal of computing explicit maps with specific topological properties; instances of this goal include computing explicit representatives of homotopy classes of maps between more general spaces X and Y (a problem raised in [7]) as well as *computing an explicit embedding* of a given simplicial complex into $\mathbb{R}^d$ (as opposed to *deciding embeddability*). Moreover, these questions are also closely related to *quantitative* questions in homotopy theory [19] and in the theory of embeddings [17]. See Section 1.2 for a more detailed discussion of these questions.

Throughout this paper, we assume that the input X is *simply connected*, i.e., that it is connected and has trivial fundamental group $\pi_1(X)$. As an additional input, our algorithm requires a specific *certificate* of simple connectivity, which we will refer to as an *explicit loop contraction* and which will be formally defined in Definition 4.2 below. The intuitive geometric meaning of this certificate is as follows. There is a standard way of defining the fundamental group $\pi_1(X)$ combinatorially, in terms of combinatorial loops (closed walks in the 1-skeleton of X starting and ending at a chosen root vertex) and combinatorial relations between walks given by the triangles (2-simplices) of X (if a walk passes through part of the boundary of a triangle, we may replace that part of the walk by walking around the triangle the other way). A contraction for a given combinatorial loop is a sequence of such combinatorial relations that reduces the loop to the trivial one. There is a standard set of generating combinatorial loops for $\pi_1(X)$ (one loop for each edge of X not lying in a chosen spanning tree), and an explicit loop contraction corresponds to a choice of a contraction for each of these generating loops. The *size* $\mathrm{size}(c)$ of an explicit loop contraction c is the total number of combinatorial relations in the chosen contractions.

For many simply connected simplicial complexes X of interest, such a certificate is easily obtained; in particular, this is the case if X is some standard triangulation of the sphere S^k, e.g., as the boundary of a $(k+1)$-simplex.[3]

Theorem 1. *There exists an algorithm that, given $d \geq 2$ and a finite simply connected simplicial complex X with an explicit loop contraction c (as defined in Def. 4.2), computes the generators $g_1, \ldots, g_k$ of $\pi_d(X)$ as simplicial maps $\Sigma_j^d \to X$, for suitable triangulations Σ_j^d of S^d, $j = 1, \ldots, k$.*

For fixed d, the time complexity is exponential in the size *(number of simplices) of X and the size of the loop contraction c; more precisely, it is $O(2^{P(\mathrm{size}(X)+\mathrm{size}(c))})$ where $P = P_d$ is a polynomial depending only on d.*

Any element of $\pi_d(X)$ can be expressed as a sum of generators, and expressing the sum of two explicit maps from spheres into X as another explicit map is a simple operation. Hence, the algorithm in Theorem 1 can convert *any* element of $\pi_d(X)$ into an explicit simplicial map.

Theorem 1 also has the following *quantitative* consequence: Fix some standard triangulation Σ of the sphere S^d, e.g., as the boundary of a $d+1$-simplex. By the classical *Simplicial Approximation Theorem* [21, 2.C], for any continuous map $f\colon S^d \to X$, there is a subdivision Σ' of Σ and a simplicial map $f'\colon \Sigma' \to X$ that is homotopic to f. Theorem 1

[3]Note that there is no algorithm that can compute an explicit loop contraction for any given simply connected simplicial complex X; indeed the size of the contractions of the generating loops cannot be bounded by any recursive function in the size of X, since any computable bound could easily be converted into an algorithm for deciding simple connectivity. For this reason, we require the loop contraction to be provided as part of the input.

implies that if f represents a generator of $\pi_d(X)$, then the size of Σ' can be bounded by an exponential function of the number of simplices of X.

Furthermore, we can show that the exponential dependence on the number of simplices in X is inevitable:

Theorem 2. *Let $d \geq 2$ be fixed. Then there is an infinite family of d-dimensional simplicial complexes X such that for any simplicial map $\Sigma \to X$ representing a generator of $\pi_d(X)$, the triangulation Σ of S^d on which f is defined has size at least $2^{\Omega(\text{size}(X))}$.*

Consequently, any algorithm for computing simplicial representatives of the generators of $\pi_d(X)$ for a simply connected simplicial complex X has time complexity at least $2^{\Omega(\text{size}(X))}$.

We suspect that any algorithm that computes representatives of $\pi_d(X)$ *must* necessarily use some explicit certificate of simple connectivity, but so far we have not been able to verify this.

As mentioned above, for most of the paper we will actually work with *simplicial sets* instead of simplicial complexes. For simplicial sets, there is another commonly used certificate that trivially implies simple connectivity and that is easy to verify, namely the property of being 1-*reduced* (see Section 5). For 1-reduced simplicial sets, the running time of the algorithm in Theorem 1 is $O(2^{P(\text{size}(X))})$. In Section 10, we prove an analogue of Theorem 2 for 1-reduced simplicial sets, which shows that the exponential dependence on size(X) is optimal also in that setting.

1.2 Related and Future Work

Computational homotopy theory and applications. This paper falls into the broader area of *computational topology*, which has been a rapidly developing area (see, for instance, the textbooks [11, 51, 32]); more specifically, as mentioned above, this work forms part of a general effort to understand the computational complexity of problems in *homotopy theory*, both because of the intrinsic importance of these problems in topology and because of applications in other areas, e.g., to algorithmic questions regarding embeddability of simplicial complexes [29, 10], to questions in topological combinatorics (see, e.g., [28]), or to the robust satisfiability of equations [16].

A central theme in topology is to understand the set $[X, Y]$ of all homotopy classes of maps from a space X to a space Y. In many cases of interest, this set carries additional structure, e.g., an abelian group structure, as in the case $\pi_d(X) = [S^d, X]$ of higher homotopy groups that are the focus of the present paper.

Homotopy-theoretic questions have been at the heart of the development of algebraic topology since the 1940's. In the 1990s, three independent groups of researchers proposed general frameworks to make various more advanced methods of algebraic topology (such as spectral sequences) *effective* (algorithmic): Schön [45], Smith [47], and Sergeraert, Rubio, Dousson, Romero, and coworkers (e.g., [46, 42, 39, 43]; also see [44] for an exposition). These frameworks yielded general *computability* results for homotopy-theoretic questions (including new algorithms for the computation of higher homotopy groups [38]), and in the case of Sergeraert et al., also a *practical implementation* in form of the Kenzo software package [22].

Building on the framework of *objects with effective homology* by Sergeraert et al., in recent years a variety of new results in computational homotopy theory were obtained [7, 27, 9, 8, 49, 15, 10, 40, 41], including, in some cases, the first *polynomial-time algorithms*, by using a refined framework of *objects with polynomial-time homology* [27, 9] that allows for a computational complexity analysis. For an introduction to this area from a theoretical computer science perspective and an overview of some of these results, see, e.g., [6] and the references therein.

Explicit maps. As mentioned above, the above algorithms often work with rather *implicit* representations of the homotopy classes in $\pi_d(X)$ (or, more generally, in $[X, Y]$) but does not yields explicit maps representing these homotopy classes.

For instance, the algorithm in [38] computes $\pi_d(X)$ as the *homology group* $H_d(F)$ of an auxiliary space $F = F_d(X)$ constructed from X in such a way that $\pi_d(X)$ and $H_d(F)$ are isomorphic as groups.[4]

More recently, Romero and Sergeraert [41] devised an algorithm that, given a 1-reduced (and hence simply connected) simplicial set X and $d \geq 2$, computes the homotopy group $\pi_d(X)$ as the homotopy group $\pi_d(K)$ of an auxiliary simplicial set K (a so-called *Kan completion* of X) with $\pi_d(X) \cong \pi_d(K)$. Moreover, given an element of this group, the algorithm can compute an explicit simplicial map $\Sigma^d \to K$ from a suitable triangulation of S^d to K representing the given homotopy class. In this way, homotopy classes are represented by explicit maps, but as maps to the auxiliary space K, which is homotopy equivalent to but not homeomorphic to the given space X.

By contrast, our general goal is to is represent homotopy classes by maps into the given space; in the present paper, we treat, as an important first instance, the case $\pi_d(X) = [S^d, X]$.

Open Problems and Future Work. Our next goal is to extend the results here to the setting of [7], i.e., to represent, more generally, homotopy classes in $[X, Y]$ by explicit simplicial maps from some suitable subdivision X' to Y (under suitable assumptions that allow us to compute $[X, Y]$).[5]

In a subsequent step, we hope to generalize this further to the *equivariant* setting $[X, Y]_G$ of [10], in which a finite group G of symmetries acts on the spaces X, Y and all maps and homotopies are required to be *equivariant*, i.e., to preserve the symmetries.

As mentioned above, one motivation is the problem of algorithmically constructing embeddings of simplicial complexes into $\mathbb{R}^d$. Indeed, in a suitable range of dimensions ($d \geq \frac{3(k+1)}{2}$), the existence of an embedding of a finite k-dimensional simplicial complex K into $\mathbb{R}^d$ is equivalent to the existence of an $\mathbb{Z}_2$-equivariant map from an auxiliary complex $\tilde{K}$ (the deleted product) into the sphere S^{d-1}, by a classical theorem of Haefliger and Weber [20, 50]. The proof of the Haefliger-Weber Theorem is, in principle, constructive, but in order to turn this construction into an algorithm to compute an embedding, one needs an explicit equivariant map into the sphere S^{d-1}.

Quantitative homotopy theory. Another motivation for representing homotopy classes by simplicial maps and complexity bounds for such algorithms is the connection to *quantitative questions* in homotopy theory [19, 13] and in the theory of embeddings [17]. Given a suitable measure of *complexity* for the maps in question, typical questions are: What is the relation between the complexity of a given null-homotopic map $f : X \to Y$ and the minimum complexity of a nullhomotopy witnessing this? What is the minimum complexity of an embedding of a simplicial complex K into $\mathbb{R}^d$? In quantitative homotopy theory, complexity is often quantified by assuming that the spaces are metric spaces and by considering Lipschitz constants (which are closely related to the sizes of the simplicial representatives of maps and homotopies [13]). For embeddings, the connection is even more direct: a typical

[4]Similarly, the algorithm in [9] constructs an auxiliary chain complex C such that $\pi_d(X)$ is isomorphic to the homology group $H_{d+1}(C)$ and computes the latter.

[5]Similarly as before, the algorithm in [7] computes $[X, Y]$ as the set $[X, P]$ for some auxiliary space P (a stage of a *Postnikov system* for Y) and represents the elements of $[X, Y] \cong [X, P]$ as maps from X to P, but not as maps to Y.

measure is the smallest number of simplices in a subdivision K' or K such that there exisits a simplexwise linear-embedding $K' \hookrightarrow \mathbb{R}^d$.

2 Outline of the Algorithm

In this section we present a high-level description of the main steps and ingredients involved in the algorithm from Theorem 1.

The algorithm in a nutshell.

1. With a slight abuse of notation, we will denote by X^{sc} the simplicial complex given as the original input, and by X another space contructed from X^{sc} in a first preprocessing step, which has more convenient properties and which we will work with for the rest of the algorithm. Specifically, given X^{sc}, we first choose a maximal tree T in its 1-skeleton and contract it to a point, thus producing a *simplicial set* X (see Section 5) with one vertex only (this is a simple example of the additional flexibility that simplicial sets offer). The homotopy groups of X^{sc} and X are isomorphic. The main part of our algorithm will be to compute a simplicial map to X representing a given homotopy class in $\pi_d(X^{sc}) \cong \pi_d(X)$. In the final Step 4 of the algorithm, this map is then converted into a map to the original simplicial complex X^{sc}.

2. In the simplest case when the space X is $(d-1)$-connected (i.e., $\pi_i(X) = 0$ for all $i \leq d-1$.), the classical Hurewicz Theorem [21, Sec. 4.2] yields an isomorphism $\pi_d(X) \cong H_d(X)$ between the dth homotopy group and the dth homology group of X. Computing generators of the homology group is known to be a computationally easy task (it amounts to solving a linear system of equations over the integers). The key is then converting the homology generators into the corresponding homotopy generators, i.e., to compute an inverse of the Hurewicz isomorphism. This was described in the work of Berger [3, 4]. Berger's algorithm requires the explicit loop contraction certificate for simple connectivity of X^{sc} that is part of the hypotheses for Theorem 1. We analyze the complexity of Berger's algorithm in detail and show that it runs in exponential time in the size of X and the size of the explicit loop contraction (assuming that the dimension d is fixed).

3. For the general case, we construct an auxiliary simplicial set F_d together with a simplicial map $\psi_d : F_d \to X$ that has the following properties:

   - F_d is a simplicial set that is $d-1$ connected, and
   - $\psi_d\colon F_d \to X$ induces an isomorphism $\psi_{d*}\colon \pi_d(F_d) \to \pi_d(X)$.

   Our construction of F_d is based on computing stages of the Whitehead tower of X [21, p. 356]; this is similar to Real's algorithm, which computes $\pi_d(X)$ as $H_d(F_d)$ as an abstract abelian group.

   The overall strategy is to use Berger's algorithm on the space F_d and compute generators of $\pi_d(F_d)$ as simplicial maps. Then we use the simplicial map ψ_d to convert a each generator of $\pi_d(F_d)$ into a map $\Sigma^d \to X$, and these maps generate $\pi_d(X)$. The main technical task for this step to show that Berger's algorithm can be applied to F_d. For this, we need to show that the explicit loop contractions for X yield an analogous certificate for simple connectivity of F_d.

4. The final step is the aforementioned conversion of maps to X into maps to X^{sc}. In general, a given a simplicial map $f\colon \Sigma^d \to X$ cannot be directly converted into a

simplicial map to the original simplicial complex X^{sc} defined on the same triangulation $\Sigma^d \to X$ of S^d. Instead, we present a procedure to construct a suitable subdivision $\mathrm{Sd}(\Sigma^d)$ of Σ^d and a simplicial map $f' \colon \mathrm{Sd}(\Sigma^d) \to X^{sc}$ representing the same homotopy class as f, see Lemma 6.4. This completes the construction of a simplicial map representing a generator of $\pi_d(X)$

Our contributions. The main ingredients of the algorithm outlined above are the computability of stages of the Whitehead tower [38] as simplicial sets with polynomial-time homology and Berger's algorithmization of the inverse Hurewicz isomorphism [3, 4].

The idea that these two tools can be combined to compute explicit representatives of $\pi_d(X)$ is rather natural and is also mentioned, for the special case of 1-reduced simplicial sets, in [41, p. 3]; however, there are a number of technical challenges to overcome in order to carry out this program (as remarked in [41, p. 3]: "Clemens Berger's algorithm, quite complex, has never been implemented, severely limiting the current scope of this approach, same comment with respect to the theoretical complexity of such an algorithm."). On a technical level, our main contributions are as follows:

- We give a complexity analysis of Berger's algorithm to compute the inverse of the Hurewicz isomorphism (Theorem 6.2).
- We show that the homology generators of the Whitehead stage F_d can be computed in polynomial time (Lemma 6.1).
- Berger's algorithm requires an explicit algorithm for loop contraction. We show how the explicit loop contraction for the original simplicial complex X^{sc} can be converted to an algorithm for contracting loops in the Whitehead tower stage F_d (Lemma 6.3).

We remark that the Whitehead tower stages are simplicial sets with infinitely many simplices, and we need the machinery of objects with polynomial-time homology to carry out the last two steps.

Structure of the paper. The remainder of the paper is structured as follows: In Section 3, we prove Theorem 2 (the exponential lower bound). In Section 4, we give the formal defintition of the explicit loop contraction that certifies simple connectivity of the input simplicial complex and that is assumed to be part of the input for our algorithm.

In Section 5, we review a number of necessary technical definitions regarding simplicial sets and the frameworks of effective and polynomial-time homology, in particular Kan's simplicial version of loop spaces and polynomial-time loop contractions for infinite simplicial sets.

In Section 6, we formally describe the algorithm from Theorem 1 and state the main technical results, mentioned above, that needed to prove the correctness of the algorithm and the running time bounds. These results are then proved in the subsequent sections: In Section 7, we describe Berger's effective Hurewicz inverse and analyze its running time (Theorem 6.2). In Section 8, we prove that the stages of the Whitehead tower have polynomial-time contractible loops (Lemma 6.3). In Section 9, we show how to convert a map to the simplicial set X into a map to the original simplicial complex X^{sc} (Lemma 6.4). Finally, in Section 10, we prove an analogue of the lower bound in Theorem 2 for simplicial sets.

3 Proof of Theorem 2

Proof of Theorem 2. We begin by constructing for every $d \geq 2$, a sequence of $\{X_k\}_{k\geq 1}$ of $(d-1)$-connected simplicial complexes, such that $H_d(X_k) \simeq \mathbb{Z}$ for all k, and for any choice of a cycle $z_k \in Z_d(X_k)$ generating the homology group, the largest coefficient in z_k grows exponentially in size(X_k).

To illustrate the idea of the construction, we start with $d = 2$. The idea is to glue X_k out of k copies of a triangulated mapping cylinders of a degree 2 map $S^1 \to S^1$, i.e. k Möbius bands, and then fill in the two open ends with one triangle each. The case $k = 1$ is shown in Figure 1. For $k \geq 2$, we simply take k copies of the triangulated Möbius band and identify the middle circle of each one to the boundary of the next one.

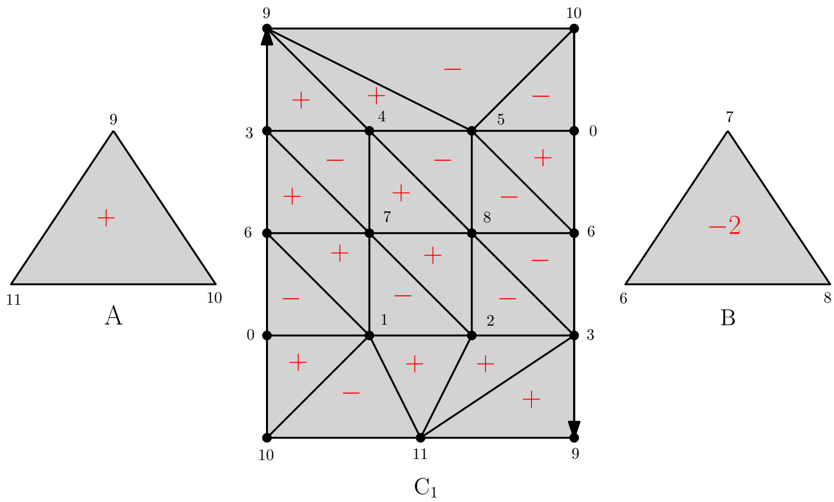


Figure 1: The Möbius band is the mapping cylinder of a degree 2 map $S^1 \to S^1$. The triangulation has four layers because starting from the boundary, which is a triangle, we first need to pass to a hexagon in order to cover the middle triangle twice, obtaining the desired degree 2 map. Connecting k copies of the Möbius band creates a mapping cylinder of a degree 2^k map, using only linearly (in k) many simplices. Gluing the full triangles A and B to the ends of this mapping cylinder finishes the construction of X_k. The red coefficients exhibit a generator ξ of $H_2(X_1) = Z_2(X_1) \simeq \mathbb{Z}$ given as a formal sum of 2-simplices.

In order to prove that X_k is simply connected and has $H_2(X_k) \simeq \mathbb{Z}$, we use standard tecniques, which can be found in details for instance in [21]. First, the mapping cylinder of any map deformation retracts to its target. In the case of X_1 this means that we retract the Möbius band C_1 to its middle circle, obtaining a topological space Y_1 [6], consisting of one circle and two discs, attached to it by maps of degree 1 and 2 respectively. The degree 2 comes from the fact that the boundary of C_1 winds twice around the middle circle. Clearly Y_1 is simply connected and has $H_2(Y_1) \simeq \mathbb{Z}$, so the same is true for X_1. The same argument works for X_k, but this time retracting each Möbius band increases the degree of the attaching map of the capping disc A by a multiplicative factor of 2. This means that the space Y_k will consist of one circle and two discs, attached to it by maps of degree 1 and 2^k respectively. This shows that X_k is simply connected and has $H_2(X_k) \simeq \mathbb{Z}$.

What is left is to investigate the coefficients in a generator of $H_2(X_k)$. Consider first X_1. It is readily verified that the chain exhibited in Figure 1 is a cycle, and that it generates $H_2(X_1) = Z_2(X_1) = \mathbb{Z}$ [7]. Observe that in the generator, given as a formal sum, all the 2-simplices of X_1 appear with coefficients ± 1, except for the triangle B which has a coefficient -2. This a reflection of the fact that through the Möbius the boundary of the triangle A winds twice around the boundary of the triangle B.

[6] The space Y_1 is not a simplicial complex any more.

[7] Since each X_k is 2-dimensional, $H_2(X_k) = Z_2(X_k) = \mathbb{Z}$

In the same way we can construct a generator for the group $H_2(X_k)$, by summing up all 2-simplices of X_k with appropriate signs, starting from the capping triangle A, and multiplying all simplices to the right by a factor of 2 on each iteration of the construction. Thus in the generator of $H_2(X_k)$ we will have a term with coefficient $\pm 2^k$ in the formal sum of 2-simplices.

The same construction can be carried out in arbitrary dimensions. The simplicial complex X_k is obtained by glueing k copies of a triangulated mapping cylinder of a degree 2 map $S^{d-1} \to S^{d-1}$, and the two open ends are filled in with two triangulated d-balls. Using the same argument as for $d = 2$, we observe that X_k is simply connected and has $H_d(X_k) \simeq \mathbb{Z}$. Moreover, taking a generator η_k of this group, on each iteration of the construction, we will necessarily multiply the coefficients of all remaining d-simplices in the formal sum by a multiplicative factor of 2. Thus, η_k will have d-simplices with coefficients 2^k.

The rest of the proof follows easily from this construction. Let $d \geq 2$ and let $\{X_k\}_{k\geq 1}$ be the sequence of simplicial complexes defined above. Since they are $(d-1)$-connected, by the theorem of Hurewicz, $\pi_d(X_k) \simeq H_d(X_k) \simeq \mathbb{Z}$. For each k, let Σ_k be a simplicial complex with $|\Sigma_k| = S^d$, and $f_k : \Sigma_k \to X_k$ a simplicial map representing a generator of $\pi_d(X_k)$. Let $\xi_k = \sum_{\sigma\in\Sigma_k} \pm\sigma$ be a chosen generator of $H_d(\Sigma_k)$, given as a formal sum of all d-simplices of Σ_k with coefficients[8] ± 1. The Hurewicz isomorphism $\pi_d(X_k) \to H_d(X_k)$ maps the representative f_k of $\pi_d(X_k)$ to the sum of simplices

$$f_k \mapsto (f_k)_*(\xi_k) = \sum_{\sigma \text{ is a } d\text{-simplex in } \Sigma_k} \pm f_k(\sigma) \in C_d(X_k)\,,$$

This chain is a cycle, and represents a generator of $H_d(X_k)$.

The complexity of any algorithm that computes $f_k : \Sigma_k \to X_k$ is at least the size of Σ_k. From the construction of X_k, the number of simplices in Σ_k grows exponentially in size(X_k), which completes the proof. □

We will show in Section 10 that any algorithm for computing homotopy classes has at least exponential complexity even if we retrict ourselves to 1-*reduced simplicial sets*. These spaces don't contain any loops and 1-reduceness itself is a trivial certificate of simply connectedness.

4 Explicit Loop Contraction

In this section, we describe the certificate for simply connectedness of a given simplicial complex X^{sc} which is assumed to be a part of the input of our main algorithm. Geometrically this certificate will correspond to an explicit contraction of loops in X^{sc}.

To define loops and their contractions, let us first choose T to be a spanning tree in the 1-skeleton of X^{sc} and R to be a chosen vertex. For each oriented edge $e = (v_1 v_2)$ we define a formal inverse to be $e^{-1} := (v_2 v_1)$ and we also consider degenerate edges (v, v). A *loop* is defined as a sequence $e_1, \ldots, e_k$ of oriented edges in X^{sc} such that

- The end vertex of e_i equals the initial vertex of e_{i+1}, and
- The initial vertex of e_1 and the end vertex of e_k equal R.

Every edge e that is not contained in T gives rise to a unique loop l_e. Further, every loop in X^{sc} is either a concatenation of such l_e's, or can be derived from such concatenation by

[8]The fact that a generator contains only ± 1 coefficients follows from the fact that Σ^d is a triangulation of the manifold S^d and hence the generator of $H_d(\Sigma^d)$ is its fundamental class.

inserting and deleting consecutive pairs (e, e^{-1}) and degenerate edges. Before we formally define our combinatorial version of loop contraction, we need the following definition.

Definition 4.1. *Let S be a set, $U \subseteq S$, $F(S)$ and $F(U)$ be free groups generated by S, U, respectively.*[9] *Let $h_U : F(S) \to F(S)$ be a homomorphism that sends each $u \in U$ to 1 and each $s \in S \setminus U$ to itself. We say that an element x of $F(S)$ equals y modulo U, if $h_U(x) = y$.*

An example of an element that is trivial modulo U is the word $s_1\, u_1\, s_2\, u_2\, s_2^{-1}\, s_1^{-1}$, where $s_i \in S$ and $u_j \in U$.

Definition 4.2. *Let S be the set of all oriented edges and oriented degenerate edges in X^{sc} and assume that a spanning tree T is chosen. Let U be the set of all oriented edges in T, including all degenerate edges. A* contraction of an edge α *is a sequence of vertices $A_0, A_1, \ldots, A_s$ and $B_1, \ldots, B_s$ such that*

- *for each i, $\{A_i, A_{i+1}, B_{i+1}\}$ is a simplex of X^{sc}, and*
- *the element of $F(S)$*

$$(A_0B_1)(B_1A_1)(A_1B_2)(B_2A_2)\ldots(B_sA_s)(A_sA_{s-1})(A_{s-1}A_{s-2})\ldots(A_1A_0) \qquad (1)$$

  equals α modulo U.

  A loop contraction in a simplicial complex *is the choice of a contraction of α for each edge $\alpha \in X^{sc} \setminus T$.*

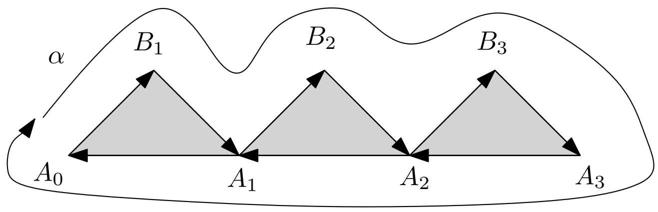


Figure 2: The loop ranging over the boundary of this geometric shape equals α, after ignoring edges in the maximal tree and canceling pairs (e, e^{-1}). The interior of the triangles gives rise to a contraction.

The geometry behind this definition is displayed in Figure 2. The sequence of A_i's and B_j's gives rise to a map from the sequence of (full) triangles into X^{sc}. The big loop around the boundary is combinatorially described by (1). We can continuously contract all of its parts that are in the tree T to a chosen basepoint, as the tree is contractible. Further, we can continuously contract all pairs of edges (e, e^{-1}) and what remains is the original edge α: with all the tree contracted to a point, it will be transformed into a loop that geometrically corresponds to l_α. The interior of the full triangles then constitutes its "filler", hence a certificate of the contractibility of l_α.

A loop contraction in the sense of Definition 2 exists iff the space X^{sc} is simply connected. One could choose different notions of loop contraction. For instance, we could provide, for each α, a simplicial map from a triangulated 2-disc into X^{sc} such that the oriented boundary of the disc would be mapped exactly to l_α. The description from Definition 4.2 could easily be converted into such map. We chose the current definition because of its canonical and algebraic nature that will be exploited later.

This finishes the exposition of our main result and in what follows, we introduce the more technical definitions that will be needed in the proofs.

[9] Formally, elements of $F(S)$ are sequences of symbols s^ϵ for $\epsilon \in \{1, -1\}$ and $s \in S$ with the relation $s^1 s^{-1} = 1$, where 1 represents the empty sequence. The group operation is concatenation.

5 Definitions and Preliminaries

In this section, we give the necessary technical definitions that will be used throughout this paper. In the first part, we recall the standard definitions for simplicial sets and the toolbox of effective homology.

Afterwards, we present Kan's definiton of a loop space and further formalize our definition of (polynomial-time) loop contractions.

5.1 Simplicial Sets and Polynomial-Time Effective Homology

Simplicial sets and their computer representation. A simplicial set X is a graded set X indexed by the non-negative integers together with a collection of mappings $d_i\colon X_n \to X_{n-1}$ and $s_i\colon X_n \to X_{n+1}$, $0 \le i \le n$ called the *face* and *degeneracy* operators. They satisfy the following identities:

$$\begin{array}{lllll} d_i s_i & = d_{i+1} s_i = \mathrm{id}; & d_i s_j & = s_j d_{i-1} & i > j+1; \\ d_i d_j & = d_{j-1} d_i; & d_i s_j & = s_{j-1} d_i & i < j; \\ s_i s_j & = s_{j+1} s_i; & & & i \le j. \end{array}$$

More details on simplicial sets and the motivation behind these formulas can be found in [33, 18].

Simplicial maps between simplicial sets are maps of graded sets which commute with the face and degeneracy operators. The elements of X_n are called *n-simplices*. We say that a simplex $x \in X_n$ is *(non-)degenerate* if it can(not) be expressed as $x = s_i y$ for some $y \in X_{n-1}$. If a simplicial set X is also a graded (Abelian) group and face and degeneracy operators are group homomorphisms, we say that X is a simplicial (Abelian) group.

A simplicial set is called k-reduced for $k \ge 0$, if it has a single i-simplex for each $i \le k$.

For a simplicial set X, we define the chain complex $C_*(X)$ to be a free Abelian group enerated by the elements of X_n with differential $\partial(c) = \sum_{i=0}^{n} (-1)^i d_i(c)$.

A simplicial set is *locally effective*, if its simplices have a specified finite encoding and algorithms are given that compute the face and degeneracy operators. A simplicial map f between locally effective simplicial sets X and Y is *locally effective*, if an algorithm is given that for the encoding of any given $x \in X$ computes the encoding of $f(x) \in Y$.

We define a simplicial set to be *finite* if it has finitely many non-degenerate simplices. Such simplicial set can be algorithmically represented in the following way. The encoding of non-degenerate simplices can be given via a finite list and the encoding of a degenerate simplex $s_{i_k} \dots s_{i_1} y$ for $i_1 < i_2 < \dots < i_k$ and a non-degenerate y can be assumed to be a pair consisting of the sequence $(i_1, \dots, i_k)$ and the encoding of y. The face operators are fully described by their action on non-degenerate simplices and can be given via finite tables. In this way, any simplicial set with finitely many non-degenerate simplices is naturally locally effective. Any choice of an implementation of the encoding and face operators is called a *representation* of the simplicial set. The *size of a representation* is the overall memory space one needs to store the data which represent the simplicial set. In what follows, we will denote by $\mathcal{I}$ the set of all representations of all 1-reduced finite simplicial sets.

Geometric realization. To each simplicial set X we assign a topological space $|X|$ called its geometric realization. The construction is similar to that of simplicial complexes. Let Δ_j be the geometric realization of a standard j-simplex for each $j \ge 0$. For each k, we define $D_i : \Delta_{k-1} \hookrightarrow \Delta_k$ to be the inclusion of a $(k-1)$-simplex into the i'th face of a k-simplex and $S_i : \Delta_k \to \Delta_{k-1}$ be the geometric realization of a simplicial map that sends the vertices $(0, 1, \dots, k)$ of Δ_k to the vertices $(0, 1, \dots, i, i, i+1, \dots, k-1)$. The geometric realization

$|X|$ is then defined to be a disjoint union of all simplices X factored by the relation $\sim$

$$|X| := (\bigsqcup_{n=0}^{\infty} X_n \times \Delta_n)/\sim$$

where $\sim$ is the equivalence relation generated by the relations $(x, D_i(p)) \sim (d_i(x), p)$ for $x \in X_{n+1}$, $p \in \Delta_n$ and the relations $(x, S_i(p)) \sim (s_i(x), p)$ for $x \in X_{n-1}$, $p \in \Delta_n$.

Similarly, a simplicial map between simplicial complexes naturally induces a continuous map between their geometric realizations.

Simplicial complexes and simplicial sets. In any simplicial complex X^{sc}, we can choose an ordering of vertices and define a simplicial sets X^{ss} that consists of all non-decrasing sequences of points in X^{sc}: the dimension of $(V_0, \ldots, V_d)$ equals d. The face operator is d_i omits the i'th coordinate and the degeneracy s_j doubles the j'th coordinate. Moreover, choosing a maximal tree T in the 1-skeleton of X enables us to construct a simplicial set $X := X^{ss}/T$ in which all vertices and edges in the tree, as well as their degeneracies, are considered to be a base-point (or its degeneracies). The geometric realizations of X^{sc} and X are homotopy equivalent and X is 0-reduced, i.e. it has one vertex only.

Homotopy groups. Let (X, x_0) be a pointed topological space. The k-th homotopy group $\pi_k(X, x_0)$ of (X, x_0) is defined as the set of pointed homotopy[10] classes of pointed continuous maps $(S^k, *) \to (X, x_0)$, where $* \in S^k$ is a distinguished point. In particular, the 0-th homotopy group has one element for each path connected component of X. For $k = 1$, $\pi_1(X, x_0)$ is the fundamental group of X, once we endow it with the group operation that concatenates loops starting and ending in x_0. The group operation on $\pi_k(X, x_0)$ for $k > 1$ assigns to $[f], [g]$ the homotopy class of the composition $S^k \xrightarrow{\pi} S^k \vee S^k \xrightarrow{f \vee g} X$ where π factors an equatorial $(k-1)$-sphere containing x_0 into a point. Homotopy groups π_k are commutative for $k > 1$.

If the choice of base-points is understood from the context or unimportant, we will use the shorter notation $\pi_k(X)$. For a simplicial set X, we will use the notation $\pi_k(X)$ for the k'th homotopy group of its geometric realization $|X|$.

An important tool for computing homotopy groups is the *Hurewicz theorem*. It says that whenever X is $(d-1)$-connected, then there is an isomorphism $\pi_d(X) \to H_d(X)$. Moreover, if the element of $\pi_d(X)$ is represented by a simplicial map $f : \Sigma^d \to X$ and $\sum_j k_j \sigma_j$ is a homology generator of $H_d(\Sigma^d)$, then the Hurewicz isomorphism maps $[f]$ to the homology class of the formal sum $\sum_j k_j f(\sigma_j)$ of d-simplices in X.

Effective homology. We call a chain complex C_* *locally effective* if the elements $c \in C_*$ have finite (agreed upon) encoding and there are algorithms computing the addition, zero, inverse and differential for the elements of C_*.

A locally effective chain complex C_* is called *effective* if there is an algorithm that for given $n \in \mathbb{N}$ generates a finite basis $c_\alpha \in C_n$ and an algorithm that for every $c \in C_*$ outputs the unique decomposition of c into a linear combination of c_α's.

Let C_* and D_* be chain complexes. A *reduction* $C_* \Rrightarrow D_*$ is a triple (f, g, h) of maps such that $f : C_* \to D_*$ and $g : D_* \to C_*$ are chain homomorphisms, $h : C_* \to C_*$ has degree 1, $fg = \mathrm{id}$ and $fg - \mathrm{id} = h\partial + \partial h$, and further $hh = hg = fh = 0$.

A locally effective chain complex C_* has *effective homology* (C_* is a *chain complex with effective homology*) if there is a locally effective chain complex $\tilde{C}_*$, reductions $C_* \Lleftarrow \tilde{C}_* \Rrightarrow C_*^{\mathrm{ef}}$ where C_*^{ef} is an effective chain complex, and all the reduction maps are computable.

[10]A homotopy $F : S^k \times I \to X$ is pointed if $F(*, t) = x_0$ for all $t \in I$.

Eilenberg-MacLane spaces. Let $d \geq 1$ and π be an Abelian group. An Eilenberg-MacLane space $K(\pi, d)$ is a topological space with the properties $\pi_d(K(\pi, d)) \simeq \pi$ and $\pi_j(K(\pi, d)) = 0$ for $0 < j \neq d$. It can be shown that such space $K(\pi, d)$ exists and, under certain natural restrictions, has a unique homotopy type. If π is finitely generated, then $K(\pi, d)$ has a locally effective simplicial model [27].

Globally polynomial-time homology and related notions. In many auxiliary steps of the algorithm, we will construct various spaces and maps. To analyse the overall time complexity, we need to parametrize all these objects by the very initial input, which is in our case an encoding of a finite 1-connected simplicial complex (or a finite 0-reduced 1-connected simplicial set) and a loop contraction, such as in Definition 4.2 (or Def. 5.4 in case of simplicial sets).

More generally, let $\mathcal{I}$ be a parameter set so that for each $I \in \mathcal{I}$ an integer $\mathrm{size}(I)$ is defined. We say that F is a parametrized simplicial set (group, chain group, ...), if for each $I \in \mathcal{I}$, a locally effective simplicial set (group, chain group, ...) $F(I)$ is given. The simplicial set F is *locally polynomial-time*, if there exists a locally effective model of $F(I)$ such that for each $k \in \mathbb{N}$ and an encoding of a k-simplex $x \in F(I)$, the encoding of $d_i(x)$ and $s_j(x)$ can be computed in time polynomial in $\mathrm{size}(\mathrm{enc}(x)) + \mathrm{size}(I)$. The polynomial, however, may depend on k. A polynomial-time map between parametrized simplicial sets F and G is an algorithm that for each $k \in \mathbb{N}$, $I \in \mathcal{I}$ and an encoding of an k-simplex x in $F(I)$ computes the encoding of $f(x)$ in time polynomial in $\mathrm{size}(\mathrm{enc}(x)) + \mathrm{size}(I)$: again, the polynomial may depend on k.

Similarly, a locally polynomial-time (parametrized) chain complex is an assignment of a computer representation $C_*(I)$ of a chain complex with a distinguished basis in each gradation, such that all these basis elements have some agreed-upon encoding. A chain $\sum_j k_j \sigma_j$ is assumed to be represented as a list of pairs $(k_j, \mathrm{enc}(\sigma_j))_j$ and has size $\sum_j (\mathrm{size}(k_j) + \mathrm{size}(\mathrm{enc}(\sigma_j)))$, where we assume that the size of an integer k_j is its bit-size. Further, an algorithm is given that computes the differential of a chain $z \in C_k(I)$ in time polynomial in $\mathrm{size}(z) + \mathrm{size}(I)$, the polynomial depending on k. The notion of a polynomial-time chain map is straight-forward.

A *globally polynomial-time chain complex* is a locally polynomial-time chain complex EC that in addition has all chain groups $EC(I)_k$ finitely generated and an additional algorithm is given that for each k computes the encoding of the generators of $EC(I)_k$ in time polynomial in $\mathrm{size}(I)$. Finally, we define a *globally polynomial-time simplicial set* to be a locally polynomial-time parametrized simplicial set F together with reductions $C_*(F) \Lleftarrow \tilde{C} \Rrightarrow EC$ where $\tilde{C}, EC$ are locally polynomial-time chain complexes, EC is a globally polynomial-time chain complex and the reduction data are all polynomial-time maps, as usual the polynomials depending on the grading k.

The name "polynomial-time homology" is motivated by the following:

Lemma 5.1. *Let F be a parametrized simplicial set with polynomial-time homology and $k \geq 0$ be fixed. Then all generators of $H_k(F(I))$ can be computed in time polynomial in* $\mathrm{size}(I)$.

Proof. For the globally polynomial-time chain complex EF and each fixed j, we can compute the matrix of the differentials $d_j : EF(I)_j \to EF(I)_{j-1}$ with respect to the distinguished bases in time polynomial in $\mathrm{size}(I)$: we just evaluate d_k on each element of the distinguished basis of $EF(I)_k$. Then the homology generators of $H_k(EC)$ can be computed using a Smith normal form algorithm applied to the matrices of d_k and d_{k+1}, as is explained in standard textbooks (such as [34]). Polynomial-time algorithms for the Smith normal form are nontrivial but known [25].

Let $x_1, \dots, x_m$ be the cycles generating $H_k(EF(I))$. We assume that reductions

$$C_*(F) \overset{(f,g,h)}{\Longleftarrow} \tilde{F} \overset{(f',g',h')}{\Longrightarrow} EF$$

are given and all the reduction maps are polynomial. Thus we can compute the chains

$$fg'(x_1), fg'(x_2), \dots, fg'(x_m)$$

in polynomial time and it is a matter of elementary computation to verify that they constitute a set of homology generators for $H_k(F(I))$. □

5.2 Loop Spaces and Polynomial-Time Loop Contraction

Principal bundles and loop group complexes. In the text we will frequently deal with principal twisted Cartesian products: these are simplicial analogues of principal fiber bundles. The definitions in this section come from Kan's article [24].

We first define the Cartesian product $X \times Y$ of simplicial sets X, Y: The set of n-simplices $(X \times Y)_n$ consists of tuples (x, y), where $x \in X_n, x \in Y_n$. The face and degeneracy operators on $X \times Y$ are given by $d_i(x, y) = (d_i x, d_i y)$, $s_i(x, y) = (s_i x, s_i y)$.

Definition 5.2 (Principal Twisted Cartesian product)**.** *Let B be a simplicial set with a basepoint $b_0 \in B_0$ and G be a simplicial group. We call a graded map (of degree -1) $\tau\colon B_{n+1} \to G_n, n \geq 0$ a* twisting operator *if the following conditions are satisfied:*

- $d_n\tau(\beta) = \tau(d_{n+1}b)^{-1}\tau(d_n b)$
- $d_i\tau(\beta) = \tau(d_{i+1}b)$ *for* $0 \leq i < n$
- $s_i\tau(b) = \tau(s_{i+1}b)$, $i \leq n$, *and*
- $\tau(s_n b) = 1_n$ *for all* $b \in B_n$ *where* 1_n *is the unit element of* G_n.

Let B, G, τ be as above. We will define a twisted Cartesian product $B \times_\tau G$ *to be a simplicial set E with $E_n = B_n \times G_n$, and the face and degeneracy operators are also as in the Cartesian product, i.e. $d_i(g, b) = (d_i g, d_i b)$, with the sole exception of d_n, which is given by*

$$d_n(b, g) := (d_n b, \tau(b) d_n(g)), \qquad (b, g) \in B_n \times G_n.$$

It is not trivial to see why this should be the right way of representing fiber bundles simplicially, but for us, it is only important that it works, and we will have explicit formulas available for the twisting operator for all the specific applications.

We remark that in the literature one can find multiple definitions of twisted operator and twisted product [33, 24, 3]and that they, in essence differ from each other based based on the decision whether the twisting "compresses" the first two or the last two face operators. Here, we follow the same notation as in [3].

Definition 5.3. *Let X be a 0-reduced simplicial set. Then we define GX to be a (non-commutative) simplicial group such that*

- *GX_n has a generator $\overline{\sigma}$ for each $(n+1)$-simplex $\sigma \in X$ and a relation $\overline{s_n y} = 1$ for each simplex in the image of the last degeneracy s_n.*
- *The face operators are given by $d_i\overline{\sigma} := \overline{d_i\sigma}$ for $i < n$ and $d_n\overline{\sigma} := (\overline{d_{n+1}\sigma})^{-1}\overline{d_n\sigma}$*
- *The degeneracy operators are $s_i\overline{\sigma} := \overline{s_i\sigma}$.*

We use the multiplicative notation, with 1 being the neutral element. It is shown in [24] that GX is a discrete simplicial analog of the loop space of X.

For algorithmic puroposes, we assume that an elements $\prod_j \overline{\sigma}_j^{k_j}$ of GX is represented as a list of pairs (σ_j, k_j) and has size $\sum_j \mathrm{size}(\sigma_j) + \mathrm{size}(k_j)$.

Definition 5.4. *Let X be a 0-reduced simplicial set. We say that a map $c_0 : GX_0 \to GX_1$ is a* contraction *of loops in X, if $d_0 c_0(x) = x$ and $d_1 c_0(x) = 1$ for each $x \in GX_0$.*

Now we will describe the connection between Definition 5.4 and Definition 4.2.

Lemma 5.5. *Let X^{sc} be a 1-connected simplicial complex with a chosen orientation of all simplices, X^{ss} the induced simplicial set, T a maximal tree in X^{sc}, and $X := X/T$ the corresponding 0-reduced simplicial set. Assume that a loop contraction in the simplicial complex X^{sc} is given, such as described in Definition 4.2. Then we can algorithmically compute $c_0(\alpha) \in GX_1$ such that $d_0 c_0(\alpha) = \alpha$ and $d_1 c_0(\alpha) = 1$, for every generator α of GX_0. Moreover, the computation of $c_0(\alpha)$ is linear in the size of X^{sc} and the size of the simplicial complex contraction data.*

Proof. For each i, the triangle $\{A_i, A_{i+1}, B_{i+1}\}$ from Def. 4.2 is in the simplicial complex X^{sc}. There is a unique oriented 2-simplex in X^{ss} of the form (V_0, V_1, V_2) (possibly degenerate) such that $\{V_0, V_1, V_2\} = \{A_i, A_{i+1}, B_{i+1}\}$. Let as denote such oriented simplex by σ_i, and its image in GX_1 by $\overline{\sigma}_i$. We will define an element $g_i \in GX_1$ such that it satisfies

$$d_0 g_i \simeq \overline{(A_i, A_{i+1})} \quad \text{and} \quad d_1 g_i \simeq \overline{(A_i, B_{i+1})}\,\overline{(B_{i+1}, A_{i+1})} \tag{2}$$

where $\simeq$ is an equivalence relation that identifies any element $\overline{(U,V)} \in GX_1$ with $\overline{(V,U)}^{-1}$ (note that only one of the symbols (U,V) and (V,U) is well defined in X^{ss}, resp. X.) Explicitly, we can define g_i with these property as follows:

- If $\sigma = (B_{i+1}, A_i, A_{i+1})$, then $g_i := \overline{\sigma}_i$,
- If $\sigma = (A_i, A_{i+1}, B_{i+1})$, then $g_i := s_0\overline{(d_2\sigma)}\,\overline{\sigma}_i\, s_0 d_0(\overline{\sigma}_i)^{-1}$
- If $\sigma = (A_{i+1}, B_{i+1}, A_i)$, then $g_i = s_0 d_0 \overline{\sigma_i}^{-1}\,\overline{\sigma_i}\, s_0(\overline{d_1\sigma_i})^{-1}$
- If $\sigma = (B_{i+1}, A_{i+1}, A_i)$, then $g_i := \overline{\sigma_i}^{-1}$
- If $\sigma = (A_{i+1}, A_i, B_{i+1})$, then $g_i := s_0 d_0 \overline{\sigma_i}\ \overline{\sigma_i}^{-1}\, s_0(\overline{d_2\sigma_i})^{-1}$
- If $\sigma = (A_i, B_{i+1}, A_{i+1})$, then $g_i := s_0(\overline{d_1\sigma_i})\,\overline{\sigma_i}^{-1} s_0 d_0 \overline{\sigma_i}$.

Let $g := g_0 \ldots, g_s$. The assumption (1) together with equation (2) immediately implies that $d_1 g (d_0 g)^{-1} = \overline{\alpha}$. Thus we define $c_0(\overline{\alpha}) := s_0 d_1(g)\, g^{-1}$. Algorithmically, to construct g amounts to going over all the triples (A_i, A_{i+1}, B_{i+1}) from a given sequence of $A_i's$ and B_j's, checking the orientation and computing g_i for every i. □

Polynomial-time loop contraction. Let F be a parametrized simplicial set such that each $F(I)$ is 0-reduced. Using constructions analogous to those defined above, GF is a parametrized locally-polynomial simplicial group whereas we assume a simple encoding of elements of GF_i as follows. If $x = \prod_j \overline{\sigma_j}^{k_j} \in GF(I)_k$ where σ_j are $(k+1)$-simplices in $F(I)$, not in the image of s_k, then we assume that x is stored in the memory as a list of pairs $(k_j, \mathrm{enc}(\sigma_j))$ and has size $\sum_j(\mathrm{size}(k_j) + \mathrm{size}(\sigma_j))$ where some σ_i may be equal to σ_j for $i \neq j$. Face and degeneracy operators are defined in Definition (5.3) and it is easy to see that for any locally polynomial-time simplicial set F, GF is a locally polynomial-time simplicial group.

Definition 5.6. *Let F be a locally polynomial simplicial set. We say that F has* polynomially contractible loops*, if there exists an algorithm that for a 0-simplex $x \in GF(I)$ computes a 1-simplex $c_0(x) \in GF(I)$ such that $d_0 x = x$, $d_1 x = 1 \in GF(I)_0$, and the running-time is polynomial in* $\mathrm{size}(x) + \mathrm{size}(I)$.

6 Proof of Theorem 1

The proof of Theorem 1 is based on a combination of four statements presented here as Lemma 6.1, Theorem 6.2, Lemma 6.3 and Lemma 6.4. Each of them is relatively independent and their proofs are partially delegated to further sections.

First we present an algorithm that, given a 1-connected finite simplicial set X and a positive integer d, outputs a simplicial set F_d and a simplicial map ψ_d such that

- the simplicial set F_d is $d-1$ connected, it has polynomial-time effective homology and polynomially contractible loops.
- the simplicial map $\psi_d \colon F_d \to X$ is polynomial-time and induces an isomorphism $\psi_{d*} \colon \pi_d(F_d) \to \pi_d(X)$.

Whitehead tower. We construct simplicial sets F_d as stages of a so-called *Whitehead tower* for the simplicial set X. It is a sequence of simplicial sets and maps

$$\cdots \longrightarrow F_d \xrightarrow{f_d} F_{d-1} \xrightarrow{f_{d-1}} \cdots \xrightarrow{f_4} F_3 \xrightarrow{f_3} F_2 = X.$$

where f_i induces an isomorphism $\pi_j(F_i) \to \pi_j(F_{i-1})$ for $j > i$ and $\pi_j(F_i) = 0$ for $j < i$. We define $\psi_d = f_d f_{d-1} \dots f_3$. One can see that F_d, ψ_d satisfy the desired properties.

Lemma 6.1. *Let X be a 1-connected finite simplicial set and let $d \geq 2$ be a fixed integer. Then there exists a polynomial-time algorithm that constructs the stages $F_2, \dots, F_d$ of the Whitehead tower of X.*

Simplicial sets $F_k(X)$, considered as simplicial sets parametrized by 1-connected finite simplicial sets, have polynomial-time homology and the maps f_k are polynomial-time simplicial maps.

Proof. The proof is by induction. The basic step is trivial as $F_2 = X$. We describe how to obtain F_{k+1}, f_{k+1} assuming that we have computed F_k, $2 \leq k < d$.

1. We compute simplicial map $\varphi_k \colon F_k \to K(\pi_k(X), k) = K(\pi_k(F_k), k)$ that induces an isomorphism $\varphi_{k*} \colon \pi_k(F_k) \to \pi_k(K(\pi_k(X), k)) \cong \pi_k(X)$. This is done using the algorithm in [9], as $K(\pi_k(X), k)$ is the first nontrivial stage of the Postnikov tower for the simplicial set F_k.

   For the simplicial set $K(\pi_k(X), k)$ and for such simplicial sets there is a classical principal bundle (twisted Cartesian product) (see [33]):

$$\begin{array}{c} K(\pi_k(X), k-1) \\ \downarrow \\ E(\pi_k(X), k-1) = K(\pi_k(X), k) \times_\tau K(\pi_k(X), k-1) \\ \big\downarrow \delta \\ K(\pi_k(X), k) \end{array}$$

2. We construct F_{k+1} and f_{k+1} as a pullback of the twisted Cartesian product:

$$\begin{array}{ccc}
K(\pi_k(X), k-1) & \xrightarrow{\quad\cong\quad} & K(\pi_k(X), k-1) \\
\downarrow & & \downarrow \\
F_{k+1} \coloneqq F_k \times_{\tau'} K(\pi_k(X), k-1) & \dashrightarrow & K(\pi_k(X), k) \times_{\tau} K(\pi_k(X), k-1) \\
{\scriptstyle f_{k+1}} \downarrow & & \downarrow {\scriptstyle \delta} \\
F_k & \xrightarrow{\quad\varphi_k\quad} & K(\pi_k(X), k).
\end{array}$$

It can be shown that the pullback, i.e. simplicial subset of pairs $(x, y) \in F_k \times E(\pi_k(X), k-1)$ such that $\delta(y) = \varphi_k(x)$, can be identified with the twisted product as above [33], where the twisting operator τ' is defined as $\tau\varphi_k$.

To show correctness of the algorithm, we assume inductively, that F_k has polynomial-time effective homology. According to [9, Section 3.8], the simplicial sets $K(\pi_k(X), k-1)$, $E(\pi_k(X), k-1)$, $K(\pi_k(X), k)$ have polynomial-time effective homology and maps φ_k, δ are polynomial-time. Further, they are all obtained by an algorithm that runs in polynomial time.

As F_{k+1} is constructed as a twisted product of F_k with $K(\pi_k(X), k)$, Corollary 3.18 of [9] implies that F_{k+1} has polynomial-time effective homology and f_{k+1} is a polynomial-time map.[11]

The sequence of simplicial sets $F_{k+1} \xrightarrow{f_{k+1}} F_k \xrightarrow{\varphi_k} K(\pi_k(X), k)$ induces the long exact sequence of homotopy groups

$$\cdots \longrightarrow \pi_i(F_{k+1}) \xrightarrow{f_{k+1*}} \pi_i(F_k) \xrightarrow{\varphi_{k*}} \pi_i(K(\pi_k(X), k)) \longrightarrow \pi_{i-1}(F_{k+1}) \longrightarrow \cdots$$

The reason why this is the case follows from a rather technical argument that identifies the simplicial set F_{k+1} with a so called *homotopy fiber* of the map $\varphi_k \colon F_k \to K(\pi_k(X), k)$. In more detail, the category of simplicial sets is right proper [18, II.8.67] and map δ is a so-called Kan fibration [33, 23]. This makes the pullback F_{k+1} coincide with so-called homotopy pullback. Further, the simplicial set $E(\pi_k(X), k-1)$ is contractible, hence the homotopy pullback is a homotopy fiber. The induced exact sequence is due to [35, chapter I.3].

The inductive assumption, together with the fact that φ_k induces an isomorphism $\varphi_{k*} \colon \pi_k(F_k) \to \pi_k(K(\pi_k(X), k))$ imply that f_k induces an isomorphism $\pi_j(F_{k+1}) \to \pi_j(F_k)$ for $j > k$ and $\pi_j(F_{k+1}) = 0$ for $j \leq k$. □

The lemma implies that the simplicial sets F_k have polynomial-time effective homology and maps $\psi_k = f_k f_{k-1} \ldots f_3$ are polynomial-time as they are defined as a composition of polynomial-time maps f_i.

The following theorem is a key ingredient of our algorithm.

Theorem 6.2 (Effective Hurewicz Inverse)**.** *Let $d > 1$ be fixed and F be an $(d-1)$-connected 0-reduced simplicial set parametrized by $\mathcal{I}$ with polynomial-time homology and polynomially contractible loops.*

[11] We remark that the paper [9] uses a different formalization of twised cartesian product than the one employed by us. However, the paper [14], on which the Corollary 3.18 of [9] is based, can be reformulated in context of the definition used here. We do not provide full details, only remark that one has to make a choice of *Eilenberg-Zilber reduction data* that corresponds to the definition of twisted cartesian product.

Then there exists an algorithm that, for a given d-cycle $z \in Z_d(F(I))$, outputs a simplicial model Σ^d of the d-sphere and a simplicial map $\Sigma^d \to F(I)$ whose homotopy class is the Hurewicz inverse of $[z] \in H_d(F(I))$.

Moreover, the time complexity is bounded by an exponential of a polynomial function in $\mathrm{size}(I) + \mathrm{size}(z)$.

We will show at the end of Section 7 that the simplicial set Σ^d and the map $\Sigma^d \to X$ can be used to create a simplicial complex Σ^{sc} with a given orientation of all simplexes, and a map $\Sigma^{sc} \to X$ (still understood to be a map between simplicial sets) representing the same homotopy class. This can be done without changing the given complexity bounds and is explained in Lemma 7.14 at the end of Section 7.

The construction of an effective Hurewicz inverse is the main result of [3] and further details are provided in Section 7. It exploits a combinatorial version of Hurewicz theorem given by Kan in [23] where $\pi_d(F)$ is described in terms of $\pi_{d-1}(\widetilde{GF})$ where $\widetilde{GF}$ is a non-commutative simplicial group that models the loop space of F. Kan showed that the Hurewicz isomorphism can be identified with a map $H_{d-1}(\widetilde{GF}) \to H_{d-1}(\widetilde{AF})$ induced by Abelianization. Berger then describes the inverse of the Hurewicz homomorphism as a composition of the maps $1, 2, 3$ in the diagram

$$\begin{array}{ccc} \pi_d(F) & \xleftarrow{\;h^{-1}\;} & H_d(F) \\ {\scriptstyle 3}\uparrow & & \downarrow{\scriptstyle 1} \\ H_{d-1}(\widetilde{GF}) & \xleftarrow[2]{} & H_{d-1}(\widetilde{AF}). \end{array}$$

Arrow 1 is induced by a chain homotopy equivalence and arrow 3 by Berger's explicit geometric model of the loop space. To algorithmize arrow 2, we need an algebraic machinery that includes an explicit contraction of k-loops in $\widetilde{GF}$ for all $k < d - 1$. Those are based partially on linear computations in the Abelian group $\widetilde{AF}$ and partially on explicit inductive formulas dealing with commutators. The lowest-dimensional contraction operation, however, cannot be algorithmized, without some external input. The possibility of providing it is is the content of the following claim:

Lemma 6.3. *Let $d \geq 2$ be a fixed integer and $\mathcal{I}$ be the set of all 1-connected finite simplicial complexes with an explicit loop contraction. Then the simplicial set F_d from Lemma 6.1, parametrized by $\mathcal{I}$, has polynomial-time contractible loops.*

The proof is technical and details can be found in Section 8.

Another ingredient for the creation of our desired simplicial homotopy generators is a conversion algorithm from a maps into simplicial sets to maps into simplicial complexes. One link that will be exploited is described by the following.

Lemma 6.4. *Let $d > 0$ be fixed. Assume that X^{sc} is a given simplicial complex with a chosen ordering of vertices and a maximal spanning tree T; we denote the underlying simplicial set by X^{ss}. Let $p : X^{ss} \to X := X^{ss}/T$ be the projection to the associated 0-reduced simplicial set. Let Σ be a given d-dimensional simplicial complex with a chosen orientation of each simplex, Σ^{ss} the induced simplicial set, and $f : \Sigma^{ss} \to X$ a simplicial map.*

Then there exists a subdivision $\mathrm{Sd}(\Sigma)$ *and a simplicial map* $f' : \mathrm{Sd}(\Sigma) \to X^{sc}$ *between* simplicial complexes[12] *such that*

$$|\Sigma| = |\mathrm{Sd}(\Sigma)| \stackrel{|f'|}{\to} |X^{sc}| \stackrel{|p|}{\to} |X|$$

[12]The constructed map f does not necessarily preserves orientations: it only maps simplices to simplices.

is homotopic to $|\Sigma^{ss}| \xrightarrow{|f|} |X|$*. Moreover,* f' *can be computed in polynomial time, assuming an encoding of the input* f, Σ, X^{sc}*,* X *and* T*.*

Thus if Σ is a sphere and f corresponds to a homotopy generator, f' is the corresponding homotopy generator represented as a simplicial map between simplicial complexes. We remark that the algorithm we describe works even if d is a part of the input, but the time complexity would be exponential in general, as the number of vertices in our subdivision $\mathrm{Sd}(\Sigma)$ would grow exponentially with d.

Proof of Theorem 1. Assume that a finite simplicial complex X^{sc} is given together with a loop contraction. We can choose an ordering of vertices and convert X^{sc} into a simplicial set. Choosing a spanning tree and contracting it to a point creates a 0-reduced simplicial set X homotopy equivalent to X^{sc}. By Lemma 5.5, we can convert the input data into a list $c_0(\alpha)$ for all generators α of GX_0 in polynomial time. We construct the simplicial set F_d from Lemma 6.1 as simplicial set with polynomial-time effective homology. Hence by Lemma 5.1 we can compute the generators of $H_d(F_d)$ in time polynomial in size(X). Due to Lemma 6.3 and Theorem 6.2, we can convert these homology generators to homotopy generators $\Sigma^d \to F_d$ in time exponential in $P(\mathrm{size}(X))$ where P is a polynomial. Then, we compose the representatives of $\pi_d(F_d)$ with ψ_d to obtain representatives of the generators of $\pi_d(X)$, another polynomial-time operation. This way, we compute explicit homotopy generators as maps into the simplicial set X. Further, we use Lemma 7.14 to create simplicial models Σ_j^{sc} of the d-sphere and maps $(\Sigma_j^{sc})^{ss} \to X$, still considered as maps between simplicial sets. Finally, by Lemma 6.4, we can compute, for each j, a subdivision of the sphere Σ_j^{sc} and a simplicial map into the simplicial complex X^{sc}, in time polynomial in the size of the representatives of $\pi_d(X)$. The overall exponential complexity bound comes from Berger's Effective Hurewicz inverse theorem. □

7 Effective Hurewicz Inverse

Here we will prove Theorem 6.2 by directly describing the algorithm proposed in [3] and analyzing its running time.

Definition 7.1. *Let* G *be a simplicial group. Then the Moore complex* $\tilde{G}$ *is a (possibly non-abelian) chain complex defined by* $\tilde{G}_i := G_i \cap (\bigcap_{j>0} \ker d_j)$ *endowed with the differential* $d_0 : \tilde{G}_i \to \tilde{G}_{i-1}$*.*

It can be shown that $d_0 d_0 = 1$ in $\tilde{G}$ and that $\mathrm{Im}(d_0)$ is a normal subgroup of $\ker d_0$ so that the homology $H_*(\tilde{G})$ is well defined.

Definition 7.2. *Let* F *be a* 0*-reduced simplicial set,* GF *the associated simplicial group from Def. 5.3, and* $\widetilde{GF}$ *its Moore complex. We define* AF *to be the Abelianization of* GF *and* $\widetilde{AF}$ *to be the Moore complex of* AF*. The simplicial group* AF *is also endowed with a chain group structure via* $\partial = \sum_j (-1)^j d_j$*. If* $\sigma \in F_k$*, we will denote by* $\overline{\sigma}$ *the corresponding simplex in* GF_{i-1}*, resp.* AF_{i-1}*.*

Note that, following Def. 5.3, the "last" differential $d_k\overline{\sigma}$ in AF_k equals $\overline{d_k\sigma} - \overline{d_{k+1}\sigma}$. Clearly, the Abelianization map $p : GF \to GF/[GF, GF] = AF$ takes $\widetilde{GF}$ into $\widetilde{AF}$.

Kan showed in [23] that for $d > 1$ and a $(d-1)$-connected simplicial set F, the Hurewicz isomorphism can be identified with the map $H_{d-1}(\widetilde{GF}) \to H_{d-1}(\widetilde{AF})$ induced by Abelianization, whereas these groups are naturally isomorphic to $\pi_d(F)$ and $H_d(F)$, respectively.

Our strategy is to construct maps representing the isomorphisms $1, 2, 3$ in the commutative diagram

$$\begin{array}{ccc} \pi_d(F) & \xleftarrow{\;h^{-1}\;} & H_d(F) \\ {\scriptstyle 3}\uparrow & & \downarrow{\scriptstyle 1} \\ H_{d-1}(\widetilde{GF}) & \xleftarrow[\;2\;]{} & H_{d-1}(\widetilde{AF}). \end{array}$$

Here h stands for the Hurewicz isomorphism, 1 is induced by a homotopy equivalence of chain complexes, 2 is the inverse of $H_{d-1}(p)$ where p is the Abelianization, and 3 represents an isomorphism between the $(d-1)$'th homology of $\widetilde{GF}$ (that models the loop space of F) and $\pi_d(F)$. The algorithms representing $1, 2, 3$ will act on representatives, that is, 1 and 2 will convert cycles to cycles and 3 will convert a cycle to a simplicial map $\Sigma^d \to F$ where $|\Sigma^d| = S^d$. In what follows, we will explicitly describe the effective versions of $1, 2, 3$ and show that the underlying algorithms are polynomial for arrows 1,2 and exponential for arrow 3.

Arrow 1.

Let F be a 0-reduced simplicial set, $C_*(F)$ be the (unreduced) chain complex of F and AF_{*-1} the shifted chain complex of AF defined by $(AF_{*-1})_i := AX_{i-1}$. As a chain complex, AF_{*-1} is a subcomplex of $C_*(F)$ generated by all simplices that are not in the image of the last degeneracy. Let $\widetilde{AF}_{*-1}$ be the Moore complex of AF_{*-1}.

Lemma 7.3. *There exists a polynomial-time strong chain deformation retraction* $(f, g, h) : C_*(F) \to \widetilde{AF}_{*-1}$*. That is,* $f : C_*(F) \to \widetilde{AF}_{*-1}$*,* $g : \widetilde{AF}_{*-1} \to C_*(F)$ *are polynomial-time chain-maps and* $h : C_*(F) \to C_{*+1}(F)$ *is a polynomial map such that* $fg = \mathrm{id}$ *and* $gf - \mathrm{id} = h\partial + \partial h$*.*

Proof. First we will define a chain deformation retraction from $C_*(F)$ to AF_{*-1} represented by $f_0 : C_*(F) \to AF_{*-1}$, $g_0 : AF_{*-1} \to C_*(F)$ and $h_0 : C_*(F) \to C_{*+1}(F)$.

The chain complex AF_{*-1} consists of Abelian groups AF_{k-1} freely generated by k-simplices in F that are not in the image of the last degeneracy s_{k-1}. On generators, we define $f_0(\sigma) = \overline{\sigma}$ whenever σ is a k-simplex not in $\mathrm{Im}(s_{k-1})$ and $f_0(x) = 0$ otherwise. Deciding whether σ is in the image of s_{k-1} amounts to deciding $\sigma = s_{k-1}d_k\sigma$ which can be done in time polynomial in $\mathrm{size}(I) + \mathrm{size}(\sigma)$, the polynomial depending on k. It follows that f_0 is a locally polynomial map.

The remaining maps are defined by $g_0(\overline{\sigma}) := \sigma - s_{k-1}d_k\sigma$ and $h_0(\sigma) := (-1)^k s_k\sigma$. These maps are locally polynomial as well and it is a matter of straight-forward computations to check that f_0 and g_0 are chain maps, $f_0g_0 = \mathrm{id}$ and $g_0f_0 - \mathrm{id} = h_0\partial + \partial h_0$.

Further, we define another chain deformation retraction from AF to $\widetilde{AF}$. For each $p \geq 0$, let A^p be a chain subcomplex of AF defined by

$$(A^p)_k := \{x \in AF_k : d_i x = 0 \quad \text{for} \quad i > \max\{k-p, 0\}\}$$

that is, the kernel of the p last face operators, not including d_0 (d_i refers here to the face operators in AF). Then A^{p+1} is a chain subcomplex of A^p and we define the maps $f_{p+1} : (A^p)_k \to (A^{p+1})_k$ by $f_{p+1}(x) = x - s_{k-p-1}d_{k-p}x$ whenever $k - p > 0$, and $f_{p+1}(x) = x$ otherwise; $g_{p+1} : A^{p+1} \to A^p$ will be an inclusion, and $h_{p+1} : (A^p)_k \to (A^p)_{k+1}$ via $h_{p+1}(x) = (-1)^{k-p}s_{k-p}x$ if $k - p > 0$ and 0 otherwise. A simple calculation shows that f_{p+1}, g_{p+1} are chain maps, $f_{p+1}g_{p+1} = \mathrm{id}$, $g_{p+1}f_{p+1} - \mathrm{id} = h_{p+1}\partial + \partial h_{p+1}$ and it is clear that $f_{p+1}, g_{p+1}, h_{p+1}$ are polynomial-time maps.

By definition, the Moore complex $\widetilde{AF} = \cap_{p>0} A^p$. The strong chain deformation retraction (f, g, h) from $C_*(F)$ to $\tilde{AF}_{*-1}$ is then defined by the infinite compositions

$$f := \dots f_{k+1} f_k \dots f_1 f_0$$
$$g := g_0 g_1 \dots g_k g_{k+1} \dots$$

and the infinite sum

$$h = h_0 + g_1 h_1 f_1 + (g_1 g_2) h_2 (f_2 f_1) + \dots$$

which are all well-defined, because when applying them to an element x, only finitely many of f_j, g_j differ from the identity map and only finitely many h_j are nonzero. These are the maps f, g, h from the lemma and we need to show that if the degree k is fixed, then we can evaluate f, g, h on $C_k(F)$ resp. $\tilde{AF}_{k-1}$ in time polynomial in the input size. However, for fixed k , the definition of f, g, h includes only f_i, g_i, h_i for $i < k$. Then f, g are composed of k polynomial-time maps and h is a sum of k polynomial-time maps. $\square$

The polynomial-time version of arrow 1 is then induced by applying the map f from Lemma 7.3.

Arrow 2.

Lemma 7.4 (Boundary certificate)**.** *Let $d > 1$ be fixed and let F be a $(d-1)$-connected simplicial set. There is an algorithm that, for $j < d-1$ and a cycle $z \in Z_j(\widetilde{AF})$, computes an element $c^A(z) \in \widetilde{AF}_{j+1}$ such that $d_0 c^A(z) = z$. The running time is polynomial in* $\mathrm{size}(z) + \mathrm{size}(I)$.

Proof. First note that the $(d-1)$-connectedness of F implies that $H_{j+1}(F) \simeq H_j(\widetilde{AF})$ are trivial for $j < d-1$, so each cycle in these dimensions is a boundary.

By assumption, F has a polynomial-time homology, which means that there exists a globally polynomial-time chain complex E_*F, a locally polynomial-time chain complex Y and polynomial-time reductions from Y to $C_*(F)$ and E_*F

$$E_*F \overset{\mathrm{P}}{\Lleftarrow} Y \overset{\mathrm{P}}{\Rrightarrow} C_*(F).$$

Let (f', g', h') be the strong deformation retraction from Y to $\widetilde{AF}_{*-1}$ defined as the composition of $Y \Rrightarrow C_*(F)$ and the strong deformation retraction from $C_*(F)$ to $\widetilde{AF}_{*-1}$ described in Lemma 7.3. Further, let f'', g'', h'' be the maps defining the reduction $Y \Rrightarrow E_*F$: all of these maps are polynomial-time.

Let $j < d-1$ and $z \in Z_j(\widetilde{AF})$, $z = \sum_j k_j y_j$. Then the element $f''g'(z)$ is a cycle in $E_{j+1}F$ and can be computed in time polynomial in $\mathrm{size}(z) + \mathrm{size}(I)$. In particular, the size of $f''g'(z)$ is bounded by such polynomial. The number of generators of $E_{j+2}F$ and $E_{j+1}F$ is polynomial in $\mathrm{size}(I)$ and we can compute, in time polynomial in $\mathrm{size}(I)$, the boundary matrix of $\partial : E_{j+2}F \to E_{j+1}F$ with respect to the generators.

Next we want to find an element $t \in E_{j+2}F$ such that $\partial t = f''g'(z)$. Using generating sets for $E_{j+2}F$, $E_{j+1}F$, this reduces to a linear system of Diophantine equations and can be solved in time polynomial in the size of the ∂-matrix and the right hand side $f''g'(z)$ [25].

Finally, we claim that $c^A(z) := f'g''(t) - f'h''g'(z)$ is the desired element mapped to z

by the differential in $\widetilde{AF}$. This follows from a direct computation

$$\begin{aligned}\partial c^A(z) :=&\partial f' g''(t) - \partial f' h'' g'(z) = \\ =&f' g''(\partial t) - \partial f' h'' g'(z) = \\ =&f' g'' f'' g'(z) - \partial f' h'' g'(z) = \\ =&f'(h''\partial + \partial h'' + \mathrm{id}) g'(z) - \partial f' h'' g'(z) = \\ =&f' h'' g' \partial z + \partial f' h'' g'(z) + f' g'(z) - \partial f' h'' g'(z) = \\ =&0 + f' g'(z) = z\end{aligned}$$

The computation of t as well as all involved maps are polynomial-time, hence the computation of $c^A(z)$ is polynomial too. □

The next lemma will be needed as an auxiliary tool later.

Lemma 7.5. *Let S be a countable set with a given encoding, G be the free (non-abelian) group generated by S, and define* $\mathrm{size}(\prod_j s_j^{k_j}) := \sum_j(\mathrm{size}(s_j) + \mathrm{size}(k_j))$. *Let $G' := [G, G]$ be its commutator subgroup.*

Then there exists a polynomial-time algorithm that for an element $g = \prod_j s_j^{k_j}$ in $G' \subseteq G$, computes elements $a_i, b_i \in G$ such that $g = \prod_j [a_j, b_j]$.

In other words, we can decompose commutator elements into simple commutators in polynomial-time at most.

Proof. Let us choose a linear ordering on S and let $g = \prod_j s_j^{k_j}$ be in G': that is, for each j, the exponents $\{k_{j'} : s_{j'} = s_j\}$ sum up to zero. We will present a bubble-sort type algorithm for sorting elements in g. Going from the left to right, we will always swap $s_j^{k_j}$ and s_{j+1}^{k+1} whenever $s_{j+1} < s_j$. Such swap always creates a commutator, but that will immediately be moved to the initial segment of commutators.

More precisely, assume that Init is the initial segment, $x = s_j^{k_j}$ and $y = s_{j+1}^{k_{j+1}}$ should be swapped, Rest represent the segment behind y, and Commutators is a final segment of commutators. The swapping will consists of these steps:

$$\begin{aligned}&\text{Init } x\ y \text{ Rest Commutators} \\ \mapsto& \text{Init } y\ x\ [x^{-1}, y^{-1}] \text{ Rest Commutators} \\ \mapsto& \text{Init } y\ x \text{ Rest } \big([x^{-1}, y^{-1}]\ [[y^{-1}, x^{-1}], \text{Rest}^{-1}] \text{ Commutators}\big)\end{aligned}$$

where the parenthesis enclose a new segment of commutators. Before the parenthesis, x and y are swapped. Each such swap requires enhancing the commutator section by two new commutators of size at most $\mathrm{size}(g)$, hence each such swap has complexity linear in $\mathrm{size}(g)$.

Let as call everything before the commutator section a "regular section". Going from left to right and performing these swaps will ensure that the largest element will be at the end of the regular section. But no later then that, the largest element y_{largest} disappears from the regular section completely, because all of its exponents add up to 0. Again, starting from the left and performing another round of swaps will ensure that the second-largest elements disappear from the regular section; repeating this, all the regular section will eventually disappear which will happen in at most $\mathrm{size}(g)^2$ swaps in total. Each swap has complexity linear in $\mathrm{size}(g)$ and the overall time complexity is not worse than cubic. □

Lemma 7.6. *Assume that F is a parametrized simplicial set with polynomially contractible loops. Let $k > 0$, $\gamma \in GF_k$ be spherical and $\alpha \in GF_k$ is arbitrary. There is a polynomial-time algorithm that computes $\delta \in GF'_{k+1}$ such that $d_0\delta = [\alpha, \gamma]$ and $d_i\delta = 1$ for all $i > 0$.*

In other words, a simple commutator of a spherical element with another element can always be "contracted" in GF'. Our proof roughly follows the construction in Kan [23, Sec. 8]

Proof. For $x \in GF_0$, we will denote by $c_0 x$ the element of $\widetilde{GF}_1$ such that $d_0 c_0 x = x$: this can be computed in polynomial-time by the assumption on loop contractions. For the simplex $\alpha \in GF_k$, we define $(k+1)$-simplices $\beta_0, \ldots, \beta_k$ by $\beta_k := s_0^k c_0 d_0^k \alpha$ and inductively $\beta_{j-1} := (s_j d_j \beta_j) \cdot (s_j \alpha^{-1}) \cdot (s_{j-1}\alpha)$ for $j < k$. Then the following relations hold:[13]

- $d_0 \beta_0 = \alpha$.
- $d_j \beta_j = d_j \beta_{j-1}$, $1 \le j \le k$
- $d_{k+1} \beta_k = 1$.

The second and third equations are a matter of direct computation, while the first follows from the more general relation $d_0^{j+1} \beta_j = d_0^j \alpha$ which can be proved by induction. If k is fixed, then all $\beta_0, \ldots, \beta_k$ can be computed in polynomial time.

The desired element δ is then the alternating product

$$\delta := [\beta_0, s_0\gamma]\,[\beta_1, s_1\gamma]^{-1} \,\ldots\, [\beta_k, s_k\gamma]^{\pm 1}.$$

□

Lemma 7.7. *Under the assumptions of Theorem 6.2, there exists homomorphisms $c_j : GF_j \to GF_{j+1}$ for $0 \le j < d-1$ such that*

1. $d_0 c_j = \mathrm{id}$,
2. $d_i c_j = c_{j-1} d_{i-1}$, $0 < i \le j+1$, *and*
3. $c_j s_i = s_{i+1} c_{j-1}$ *for* $0 < j < d-1$ *and* $0 \le i < j$,
4. $d_1 c_0(x) = 1$ *for all* $x \in GF_0$.

If d is fixed and $x \in GF_j$, $j < d-1$, then $c_j(x)$ can be computed in exponential time.

Proof. The homomorphism c_0 can be constructed directly from the assumption on polynomial contractibility of loops. We have a canonical basis of GF_0 consisting of all non-degenerate 1-simplices of F. For $\sigma \in F_1$, we denote by $\overline{\sigma}$ the corresponding generator of GF_0. The we define $c_0(\prod \overline{\sigma}_j^{k_j})$ to be $\prod b_j^{k_j}$ where b_j is the element of GF_1 such that $d_0 b_j = \overline{\sigma}_j$ and $d_1 b_j = 1$.

In what follows, assume that $1 \le k < d-1$ and c_i have been defined for all $i < k$. We will define c_k in the following steps.

Step 1. Contractible elements.

Let $x \in GF_k$. We will say that x is *contractible* and $y \in GF_{k+1}$ is a *contraction of* x, if $d_0 y = x$ and $d_i y = c_{k-1} d_{i-1} x$ for all $i > 0$.

The general strategy for defining c_k will be to find a contraction h for each basis element ($(k+1)$-simplex) $g \in GF_k$ and define $c_k(g) := h$. This will enforce properties 1 and 2. Moreover, in case when g is degenerate, the contraction will be chosen in such a way that property 3 holds too; otherwise it holds vacuously. Property 4 only deals with c_0 and is satisfied by its construction above.

[13] Kan uses a slightly different convention in [23] but the resulting properties are the same. The sequence $\beta_0, \ldots, \beta_k$ can be interpreted as a discrete path from α to the identity element.

Step 2. Contraction of degenerate elements.

Let $g = s_i y$ be a basis element of GF_k, $y \in GF_{k-1}$. Then g can be uniquely expressed as $s_j z$ where j is the maximal i such that $g \in \mathrm{Im}(s_i)$. We then define $c_k(g) \coloneqq s_{j+1}c_{k-1}(z)$. Note that

$$d_0 c_k(g) = d_0 s_{j+1} c_{k-1}(z) = s_j d_0 c_{k-1}(z) = s_j z = g,$$

so property 1 is satisfied. To verify property 2, first assume that $i \in \{j+1, j+2\}$. Then we have

$$d_i c_k(g) = d_i s_{j+1} c_{k-1}(z) = c_{k-1}(z) = c_{k-1} d_{i-1} s_j z = c_{k-1} d_{i-1} g.$$

This fully covers the case $k = 1$, because then the only possibility is $j = 0$ and $i \in \{1, 2\}$. Further, let $k > 1$. If $i \leq j$, then we have

$$\begin{aligned} d_i c_k g =& d_i c_k s_j z = d_i s_{j+1} c_{k-1}(z) = s_j d_i c_{k-1}(z) = s_j c_{k-2} d_{i-1} z = \\ =& c_{k-1} s_{j-1} d_{i-1} z = c_{k-1} d_{i-1} s_j z = c_{k-1} d_{i-1} g \end{aligned}$$

and if $i > j + 2$, then the computation is completely analogous, using the relation $d_i s_{j+1} = s_{j+1} d_{i-1}$ instead.

So far, we have shown that $c_k(g) \coloneqq s_{j+1} c_{k-1} g$ is a contraction of g. It remains to show property 3. That is, we have to show that if $g = s_j z$ can also be expressed as $s_i y$, then $c_k(s_i y) = s_{i+1} c_{k-1} y$.

The degenerate element g has a unique expression $g = s_{i_u} \dots s_{i_1} s_{i_0} v$ where $i_0 < i_1 < \dots < i_u = j$ and is expressible as $s_i x$ iff $i = i_j$ for some $j = 0, 1, \dots, u$. Choosing such $i < j$, we can rewrite g as $g = s_j s_i v$ for some v and then $g = s_i s_{j-1} v$, so that $y = s_{j-1} v$ and $z = s_i v$. Then we again use induction to show

$$\begin{aligned} c_k(s_i y) =& s_{j+1} c_{k-1}(z) = s_{j+1} c_{k-1} s_i v = s_{j+1} s_{i+1} c_{k-2} v = \\ =& s_{i+1} s_j c_{k-2} v = s_{i+1} c_{k-1} s_{j-1} v = s_{i+1} c_{k-1} y \end{aligned}$$

as required.

Step 3. Decomposition into spherical and conical parts.

We will call an element $\hat{x} \in GF_k$ to be *conical*, if it is a product of elements that are either degenerate or in the image of c_{k-1}. Let $x \in GF_k$ be arbitrary. We define $x_k \coloneqq x$ and inductively $x_{i-1} \coloneqq x_i (s_{i-1} d_i x_i)^{-1}$. In this way we obtain $x_0, \dots, x_n$ such that x_i is in the kernel of d_j for $j > i$ and $x = x_0 y$ where y is a product of degenerate simplices. Further, let $x^s \coloneqq x_0 (c_{k-1} d_0 x_0)^{-1}$. A simple computation shows that x^s is *spherical*, that is, $d_i x^s = 1$ for all i. We obtain an equation $x = x^s \hat{x}$ where $\hat{x} = (c_{k-1}(d_0 x_0) y$; this is a decomposition of x into a spherical part x^s and a conical element $\hat{x}$.

We will define c_k on non-degenerate basis elements $g = \overline{\sigma}$ by first decomposing $g = g^S \hat{g}$ into a spherical and conical part, finding contractions h_1 of g^S and h_2 of $\hat{g}$, and defining $c_k(g) \coloneqq h_1 h_2$. Then $c_k(g)$ is a contraction of g and hence satisfies properties 1 and 2: property 3 is vacuously true once g is non-degenerate.

Step 4. Contraction of the conical part.

Let $\hat{x} \coloneqq c_{k-1}(d_0 x_0)\, y$ be the conical part defined in the previous step. By construction, $x_0 \in \tilde{GF}_k$ and y is a product of degenerate elements $s_{i_1} u_1 \dots s_{i_l} u_l$. We define the contraction of $c_{k-1}(d_0 x_0)$ to be

$$\tilde{c}_k(c_{k-1}(d_0 x_0)) \coloneqq s_0 c_{k-1}(d_0 x_0).$$

Note that this satisfies property 1 as $d_0 \tilde{c}_k c_{k-1}(d_0 x_0) = c_{k-1}(d_0 x_0)$. For property 2, we first verify

$$d_1 \tilde{c}_k c_{k-1}(d_0 x_0) = d_1 s_0 c_{k-1}(d_0 x_0) = c_{k-1}(d_0 x_0) = c_{k-1} d_0 c_{k-1}(d_0 x_0).$$

Not let $i \geq 2$. If $k = 1$, then the remaining face operator is d_2 and we have

$$d_2\tilde{c}_1 c_0(d_0x_0) = d_2 s_0 c_0(d_0x_0) = s_0 d_1 c_0(d_0x_0) = 1 = c_0 d_1 c_0(d_0x_0)$$

using axiom 4 for c_0. Finally, if $i \geq 2$ and $k \geq 2$, we have

$$\begin{aligned} d_i\tilde{c}_k c_{k-1}(d_0x_0) =& d_i s_0 c_{k-1}(d_0x_0) = s_0 d_{i-1} c_{k-1}(d_0x_0) = s_0 c_{k-1} d_{i-2} d_0 x_0 = \\ =& s_0 c_{k-1} d_0 d_{i-1} x_0 = s_0 c_{k-1} d_0 1 = 1 = c_{k-1} c_{k-2} d_0 d_{i-1} x_0 = \\ =& c_{k-1} c_{k-2} d_{i-2} d_0 x_0 = c_{k-1} d_{i-1} c_{k-1}(d_0x_0), \end{aligned}$$

where we exploited the fact that $x_0 \in \widetilde{GF}_k$ and hence $d_u x_0 = 1$ for $u \geq 2$.

The contraction of degenerate elements y has already been defined in Step 2, so we can define a contraction of $c_{k-1}(d_0x_0)y$ to be $s_0 c_{k-1}(d_0x_0)\, c_k(y)$.

Step 5. Contraction of commutators.

Let $g' \in GF'_k$ be an element of the commutator subgroup. By Lemma 7.5, we can algorithmically decompose g' into a product of simple commutators, so to find a contraction of g', it is sufficient to find a contraction of each simple commutator $[x, y]$ in this decomposition.

Let $x = x^S\, \hat{x}$ and $y = y^S\, \hat{y}$ be the decompositions into spherical and conical parts described in Step 3. Using the notation ${}^b a := bab^{-1}$, we can decompose $[x, y]$ as follows [3, p. 60]:

$$[x, y] = \left([x, y][\hat{y}, x]\right)\left([x, \hat{y}][\hat{y}, \hat{x}]\right)[\hat{x}, \hat{y}] = [{}^{xy}x^{-1}, {}^{xy}(y^{-1}\hat{y})]\,[{}^{x}\hat{y}, {}^{x}(x^{-1}\hat{x})]\,[\hat{x}, \hat{y}]. \tag{3}$$

Both $x^{-1}\hat{x}$ and $y^{-1}\hat{y}$ are spherical simplices and so are their conjugations. It follows that equation (3) can be rewritten to $[x, y] = [\alpha_1, \gamma_1]\,[\alpha_2, \gamma_2]\,[\hat{x}, \hat{y}]$ where γ_1 and γ_2 are spherical. All of these decompositions are done by elementary formulas and are polynomial-time in the size of x and y.

By Lemma 7.6 we can find an elements $\lambda_i \in \widetilde{GF}_{k+1}$ such that $d_0\lambda_i = [\alpha_i, \gamma_i]$, $i = 1, 2$, in polynomial time. Further, both $\tilde{x}$ and $\tilde{y}$ are conical and they are in the form $\tilde{x} = c_0(d_0x_0)x^d$ where $x_0 \in \widetilde{GF}_k$ and x^d is degenerate; similar decomposition holds for y. In Step 4 we showed how to compute elements c^x and c^y such that c^x, c^y is a contraction of $\hat{x}$, $\hat{y}$, respectively. Then $[c^x, c^y]$ is a contraction of $[\hat{x}, \hat{y}]$ and $\lambda_1\lambda_2[c^x, c^y]$ is a contraction of $[x, y]$.

Step 6. Contraction of spherical elements.

The last missing step is to compute a contraction of the spherical element g^S where g^S is the spherical part of a basis element $g \in GF_k$.

Let us denote by p the projection $GF \xrightarrow{p} AF$. The projection $z := p(g^S)$ is in the kernel of all face operators and hence a cycle in $\widetilde{AF}_k$. By Lemma 7.4, we can compute $t := c^A_k(z) \in \widetilde{AF}_{k+1}$ such that $d_0 t = z$, in polynomial time. Let $h \in GF_{k+1}$ be any p-preimage[14] of t. Let $h_k := h$ and inductively define $h_{j-1} := h_j(s_{j-1}d_jh_j)^{-1}$ for $j < k$. Then h_0 is in the kernel of all faces except d_0, that is, $h_0 \in \widetilde{GF}_{k+1}$. It follows that $p(h_0) \in \widetilde{AF}_{k+1}$ is in the kernel of all faces except d_0. We claim that $p(h_0) = t$.This can be shown as follows: assume that $p(h_j) = t$, then $p(h_{j-1}) = p(h_j) + p(s_{j-1}d_jh_j^{-1}) = t + s_{j-1}d_jt = t + 0 = t$.

[14]For $t = \sum_j k_j\overline{\sigma}_j$, we may choose $h = \prod_j \overline{\sigma}_j^{k_j}$ (choosing any order of the simplices).

We have the following commutative diagram:

$$\begin{array}{ccccc}
 & & h_0 & \longmapsto & t \\
\widetilde{GF}'_{k+1} & \hookrightarrow & \widetilde{GF}_{k+1} & \xrightarrow{p} & \widetilde{AF}_{k+1} \\
\downarrow d_0 & & \downarrow d_0 & & \downarrow d_0 \\
\widetilde{GF}'_{k} & \hookrightarrow & \widetilde{GF}_{k} & \xrightarrow{p} & \widetilde{AF}_{k} \\
 & & g^S & \longmapsto & z
\end{array}$$

Both g^S and d_0h_0 are mapped by p to the same element z: it follows that $g^S(d_0h_0)^{-1}$ is mapped by p to zero and hence is an element of the commutator subgroup. Let $\tilde{h}$ be the contraction of $g^S(d_0h_0)^{-1}$, computed in Step 5, and finally let $h := \tilde{h}h_0$. Then h is an element of $\widetilde{GF}_{k+1}$ and a direct computation shows that $d_0\tilde{h} = g^S$ as desired.

This completes the construction of c_k: for each non-degenerate basis element g of GF_k, $c_k(g)$ is defined to be the product of the contraction of g^S and the contraction[15] of $\hat{g}$.

All the subroutines described in the above steps are polynomial-time. Thus we showed that if there exists a polynomial-time algorithm for c_{k-1}, then there also exists a polynomial-time algorithm for c_k. The existence of a polynomial-time c_0 follows from the assumption on polynomial loop contractibility and d is fixed, thus there exists a polynomial-time algorithm that for $x \in GF_j$ computes $c_j(x)$ for each $j < d-1$. □

Lemma 7.8 (Construction of arrow 2)**.** *Under the assumption of Theorem 6.2, let $z \in Z_{d-1}(\widetilde{AF})$ be a cycle. Then there exists a polynomial-time algorithm that computes a cycle $x \in Z_{d-1}(\widetilde{GF})$ such that the Abelianization of x is z.*

The assignment $z \mapsto x$ is hence an effective inverse of the isomorphism $H_{d-1}(\widetilde{GF}) \to H_{d-1}(\widetilde{AF})$ on the level of representatives.

Proof. Let c_{d-2} be the contraction from Lemma 7.7 and $z \in Z_{d-1}(\widetilde{AF})$ be a cycle. First choose $y \in GF_{d-1}$ such that $p(y) = z$. Creating the sequence $y_n := y$, $y_{j-1} := y_j s_{j-1} d_j y_j^{-1}$ for decreasing j, yields an element $y_0 \in \widetilde{GF}_{d-1}$ that is still mapped to z by p, similarly as in Step 4 of Lemma 7.7. The equation $pd_0(y_0) = d_0p(y_0) = d_0z = 0$ shows that d_0y_0 is in the commutator subgroup $\widetilde{GF}'_{d-2}$. We define $x := y_0c_{d-2}(d_0y_0)^{-1}$: this is already a cycle in $\widetilde{GF}_{d-1}$ and $p(x) = p(y_0) = z$. □

Arrow 3.

The construction of map 3 is one of the main results from [4] and involves further technical definitions. Here, we describe the main points of the construction only while details are given in later sections.

Given a 0-reduced simplicial set F, there exists a simplicial group $\overline{\Omega}F$ that is a discrete analog of a loopspace of F i.e. $\pi_{d-1}(\overline{\Omega}F) \cong \pi_d(F)$. Further, there is a homomorphism of simplicial groups $t\colon GF \to \overline{\Omega}F$ that induces an isomorphism on the level of homotopy groups. This is described in [4, Proposition 3.3].

The homomorphism t is given later by formula (5) and the simplicial set $\overline{\Omega}F$ is described in the next section. Here, we remark that the size of $t(g)$ is exponential in size of g.

[15] The connectivity assumption on F was exploited in the existence of the contraction c_j^A on the Abelian part.

Finally, Lemma 7.13 describes an algorithm that for a spherical element $\gamma \in \overline{\Omega}F_{d-1}$ constructs a simplicial map $\gamma_{\mathsf{sph}}\colon \Sigma^d(\gamma) \to F$ such that $\pi_{d-1}(\overline{\Omega}F) \ni [\gamma] \simeq [\gamma_{\mathsf{sph}}] \in \pi_d(F)$.

The size of γ_{sph} is polynomial in size(γ). Hence, given a spherical $g \in \widetilde{GF}_{d-1}$, the algorithm produces $t(g)_{\mathsf{sph}}\colon \Sigma^d(t(g)) \to F$ that is exponential with respect to size(g).

Berger's model of the loop space.

Definition 7.9 (Oriented multigraph on X_n). *Let X be a 0-reduced simplicial set. We define a directed multigraph $MX_n = (V_n, E_n)$, where the set of vertices $V_n = X_n$ and the set of edges E_n is given by*

$$E_n = \{[x, i]^\epsilon \mid x \in X_{n+1}, 0 \le i \le n, \epsilon \in \{1, -1\}\}.$$

We define maps source , target : $E_n \to V_n$ *by setting* source $[x, i] = d_{i+1}x$, target $[x, i] = d_i x$ *and* source $[x, i]^{-1}$ = target $[x, i]$ *and* target $[x, i]^{-1}$ = source $[x, i]$.

An edge $[x, i]^\epsilon \in E_n$ is called compressible*, if $x = s_i x'$ for some $x' \in V_n = X_n$.*

Definition 7.10 (Paths). *Let $X \in$ sSet. A sequence of edges in MX_n*

$$\gamma = [x_1, i_1]^{\epsilon_1}[x_2, i_2]^{\epsilon_2} \cdots [x_k, i_k]^{\epsilon_k} \tag{4}$$

*is called an n-*path*, if* target $[x_j, i_j]^{\epsilon_j}$ = source $[x_{j+1}, i_{j+1}]^{\epsilon_{j+1}}$, $1 \le j < k$.

Moreover, for every $x \in V_n = X_n$ we define a path of length zero 1_x with the property source $1_x = x =$ target 1_x *and relations $a1_x = a$ whenever* target $a = x$ *and $1_x b = b$ whenever* source $b = x$.

The set of paths on MX_n is denoted by IX_n. Let $\gamma \in IX_n$ by as in (4). We define source γ = source $[x_1, i_1]^{\epsilon_1}$ *and* target γ = target $[x_k, i_k]^{\epsilon_k}$. *The* inverse *of γ,denoted γ^{-1}, is defined as*

$$\gamma^{-1} = [x_k, i_k]^{-\epsilon_k} \cdots [x_1, i_1]^{-\epsilon_1}.$$

if $\gamma = 1_x$, then $\gamma^{-1} = \gamma$. Note that each path is either equal to 1_x for some x or can be represented in a form such as (4), without any units.

For algorithmic purposes, we assume that a path $\gamma = [x_1, i_1]^{\epsilon_1}[x_2, i_2]^{\epsilon_2} \cdots [x_k, i_k]^{\epsilon_k}$ is represented as a list of triples (x_j, i_j, ϵ_j) and has size

$$\text{size}(\gamma) := \sum_j \text{size}(x_j) + \text{size}(i_j) + \text{size}(\epsilon_j),$$

which is bounded by a linear function in $\sum_j \text{size}(x_j)$.

Given an edge $[x, i]^\epsilon \in MX_n$, we define operators

$$d_0, \dots d_n\colon E_n \to IX_{n-1} \text{ and } s_0, \dots, s_n\colon E_n \to IX_{n+1}$$

called *face* and *degeneracy* operators, respectively. These are given as follows

$$d_j[x, i]^\epsilon = \begin{cases} [d_j x, i-1]^\epsilon, & j < i; \\ 1_{d_i d_{i+1} x}, & i = j; \\ [d_{j+1} x, i]^\epsilon, & j > i. \end{cases} \qquad s_j[x, i]^\epsilon = \begin{cases} [s_j x, i+1]^\epsilon, & j < i; \\ [s_i x, i+1][s_{i+1} x, i])^\epsilon, & i = j; \\ [s_{j+1} x, i]^\epsilon, & j > i. \end{cases}$$

One can now extend the definition of face and degeneracy operators to paths, i.e. we define operators $d_0, \dots d_n\colon IX_n \to IX_{n-1}$ and $s_0, \dots, s_n\colon IX_n \to IX_{n+1}$

$$d_j\gamma = \begin{cases} d_j([x_1, i_1]^{\epsilon_1}) d_j([x_2, i_2]^{\epsilon_2}) \cdots d_j([x_k, i_k]^{\epsilon_k}) & \text{if} \quad \gamma = [x_1, i_1]^{\epsilon_1}[x_2, i_2]^{\epsilon_2} \cdots [x_k, i_k]^{\epsilon_k}, \\ 1_{d_j x} & \text{if} \quad \gamma = 1_x, x \in X_n. \end{cases}$$

$$s_j\gamma = \begin{cases} s_j([x_1,i_1]^{\epsilon_1})s_j([x_2,i_2]^{\epsilon_2})\cdots s_j([x_k,i_k]^{\epsilon_k}) & \text{if} \quad \gamma = [x_1,i_1]^{\epsilon_1}[x_2,i_2]^{\epsilon_2}\cdots[x_k,i_k]^{\epsilon_k} \\ 1_{s_jx} & \text{if} \quad \gamma = 1_x, x \in X_n. \end{cases}$$

With the operators defined above, one can see that IX is in fact a simplicial set.

For any $\gamma, \gamma' \in IX$ such that $\mathsf{target}\,\gamma = \mathsf{source}\,\gamma'$, we define a composition $\gamma \cdot \gamma'$ in an obvious way.

If the simplicial set X is 0-reduced, we denote the unique basepoint $* \in X_0$. Abusing the notation, we denote the iterated degeneracy of the basepoint $\underbrace{s_0 \cdots s_0 *}_{k-\text{times}} \in X_k$ by $*$ as well. With that in mind, we define simplicial subsets PX, ΩX of IX as follows:

$$PX = \{\gamma \in IX \mid \mathsf{target}\,\gamma = *\} \quad \Omega X = \{\gamma \in IX \mid \mathsf{source}\,\gamma = * = \mathsf{target}\,\gamma\}.$$

We remark that simplicial sets $PX, \Omega X$ intuitively capture the idea of pathspace and loopspace in a simplicial setting.

Definition 7.11. *A path $\gamma = [x_1,i_1]^{\epsilon_1}[x_2,i_2]^{\epsilon_2}\cdots[x_k,i_k]^{\epsilon_k} \in IX$ is called* reduced*, if for every $1 \le j < k$ the following condition holds:*

$$(x_j = x_{j+1} \,\&\, i_j = i_{j+1}) \Rightarrow \epsilon_j = \epsilon_{j+1}.$$

e.g. an edge in the path γ is never followed by its inverse.

An edge $[x,i]^\epsilon \in E_n$ is called compressible*, if $x = s_ix'$ for some $x' \in V_n = X_n$. A path is* compressed *if it does not contain any compressed edge.*

We define relation $\sim_R$ on IX (or rather on each IX_n) as a relation generated by

$$[x,i]^\epsilon[x,i]^{-\epsilon} \sim_R 1_{\mathsf{source}\,([x,i]^\epsilon)}, \quad n \in \mathbb{N}_0, [x,i]^\epsilon \in E_n.$$

Similarly, we define $\sim_C$ on IX as a relation generated by

$$[x,i]^\epsilon \sim_C 1_{\mathsf{source}\,([x,i]^\epsilon)}, \quad \text{if}\, [x,i]^\epsilon \in E_n \,\text{is compressible.}$$

We finally define $\overline{I}X = (IX/\sim_C)/\sim_R$. Similarly, one defines $\overline{P}X, \overline{\Omega}X$.

For $\gamma, \gamma' \in IX_n$, we write $\gamma \sim \gamma'$ if they represent the same element in $\overline{I}X_n$. the symbol $\overline{\gamma}$, denotes the (unique) compressed and reduced path such that $\gamma \sim \overline{\gamma}$. One can see $\overline{I}X$ ($\overline{P}X, \overline{\Omega}X$) as the set of reduced and compressed paths in $IX(PX, \Omega X)$.

In a natural way, we can extend the definition of face and degeneracy operators d_i, s_i on sets $\overline{I}X(\overline{P}X,\overline{\Omega}X)$ by setting $d_i\gamma = \overline{d_i\gamma}$ and $s_i\gamma = \overline{s_i\gamma}$. One can check that this turns $\overline{I}X$, $\overline{P}X$ and $\overline{\Omega}X$ into simplicial sets.

Similarly, we define operation $\cdot\colon \overline{\Omega}X_n \times \overline{\Omega}X_n \to \overline{\Omega}X_n$ by $\gamma \cdot \gamma' \mapsto \overline{\gamma\gamma'}$, i.e. we first compose the loops and then assign the appropriate compressed and reduced representative. With the operation defined as above, $\overline{\Omega}X$ is a simplicial group.

Homomorphism $t\colon GX \to \overline{\Omega}X$. We first describe how to any given $x \in X_n$ assign a path $\gamma_x \in \overline{P}X_n$ with the property $\mathsf{source}\,\gamma_x = x$ and $\mathsf{target}\,\gamma_x = *$:

For $x \in X_n$, $n > 0$, the 0-reducedness of X gives us $d_{i_1}d_{i_2}\cdots d_{i_n}x = *$, here $i_j \in \{0,\ldots,j\}$, $0 < j \le n$. In particular, $d_0d_1\cdots d_{n-1}x = *$. Using this, we define

$$\gamma_x = [s_nx, n-1][s_ns_{n-1}d_{n-1}x, n-2]\cdots[s_ns_{n-1}\cdots s_1d_1d_2\cdots d_{n-1}x, 0].$$

Ignoring the degeneracies, one can see the sequence of edges as a path

$$x \to d_{n-1}x \to d_{n-2}d_{n-1}x \to \cdots \to d_0d_1\cdots d_{n-1}x.$$

We define the homomorphism t on the generators of GX_n, i.e. on the elements $\overline{x}$, where $x \in X_{n+1}$ as follows:

$$t(\overline{x}) = \overline{\gamma^{-1}_{d_{n+1}x}[x,n]\gamma_{d_n x}}. \tag{5}$$

This is an element of $\overline{\Omega} X_n$.

The algorithm representing the map t has *exponential time complexity* due to the fact that an element $\overline{\sigma}^k$ with size $\mathrm{size}(\sigma) + \mathrm{size}(k)$ is mapped to

$$\underbrace{\overline{\gamma^{-1}_{d_{n+1}x}[x,n]\gamma_{d_n x} \ \ldots \ \gamma^{-1}_{d_{n+1}x}[x,n]\gamma_{d_n x}}}_{k \text{ times}}$$

which in general can have size proportional to k. Assuming an encoding of integers such that $\mathrm{size}(k) \simeq \ln(k)$, this amounts to an exponential increase.

Universal preimage of a path. Intuitively, one can think of the simplicial set IX of paths as of a discretized version of space of continuous maps $|X|^{[0,1]}$. In particular, $\gamma \in IX_{d-1}$ is a walk through a sequence of d-simplices in X that connect $\mathsf{source}\,\gamma$ with $\mathsf{target}\,\gamma$. However, in the continuous case an element $\mu \in |X|^{[0,1]}$ corresponds to a continuous map $\mu \colon [0,1] \to |X|$. We want to push the parallels further, namely, given any nontrivial[16] $\gamma \in IX_{d-1}$, we aim to define a simplicial set $\mathsf{Dom}(\gamma)$ and a simplicial map $\gamma_{\mathsf{map}} : \mathsf{Dom}(\gamma) \to X$ with the following properties:

1. $|\mathsf{Dom}(\gamma)| = D^d$
2. γ_{map} maps $\mathsf{Dom}(\gamma)$ to the set of simplices contained in the path γ.

We will utilize the following construction given in [4].

Definition 7.12. *Let $\gamma \in IX_{d-1}$. We define $\mathsf{Dom}(\gamma)$ and γ_{map} as follows. Suppose, that $\gamma = [y_1, i_1]^{\epsilon_1}[y_2, i_2]^{\epsilon_2} \cdots [y_k, i_k]^{\epsilon_k}$. For every edge $[y_j, i_j]^{\epsilon_j}$, let α_j be the simplicial map $\Delta^d \to y_j$ sending the nondegenerate d simplex in Δ^d to y_j.*

We define $\mathsf{Dom}(\gamma)$ as a quotient of the disjoint union of k copies of Δ^d:

$$\mathsf{Dom}(\gamma) = \bigsqcup_{i=1}^{k} \Delta^d / \sim$$

where each copy of Δ^d corresponds to a domain of a unique α_j and the relation is given by

$$(\alpha_j)^{-1}\mathsf{target}\,([y_j, i_j]^{\epsilon_j}) \sim (\alpha_{j+1})^{-1}\mathsf{source}\,([y_{j+1}, i_{j+1}]^{\epsilon_{j+1}}).$$

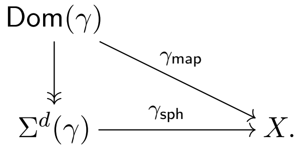
The map γ_{map} is induced by the collection of maps $\alpha_1, \ldots, \alpha_k$:

$$\begin{array}{ccc} \bigsqcup_{i=1}^{k} \Delta^d & & \\ \big\downarrow & \searrow^{\alpha_1,\ldots,\alpha_k} & \\ \mathsf{Dom}(\gamma) & \xrightarrow{\gamma_{\mathsf{map}}} & X. \end{array}$$

We recall that simplicial set $\overline{I}X$ was defined as the set of "reduced and compressed" paths in IX. Similarly, one introduces a reduced and compressed versions of the construction Dom. As a final step we then get

[16]By nontrivial we mean that $\gamma \neq 1_x$ for any $x \in X_{d-1}$.

Lemma 7.13 (Section 2.4 in [4])**.** *Let $\gamma \in \overline{\Omega}X_{d-1}$ such that $d_i\gamma = 1 \in \overline{\Omega}X$ for all i. Then the map $\gamma_{\mathsf{map}}\colon \mathsf{Dom}(\gamma) \to X$ factorizes through a simplicial set model of the sphere $\Sigma^d(\gamma)$ as follows:*

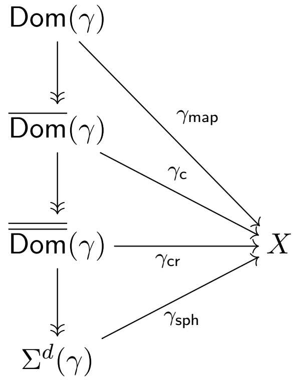


Further, $\pi_{d-1}(\overline{\Omega}X) \ni [\gamma] \simeq [\gamma_{\mathsf{sph}}] \in \pi_d(X)$.

We will not give the proof of correctness of Lemma 7.13 (it can be found in [4]). Instead, in the next section, we only describe the algorithmic construction of $\gamma_{\mathsf{sph}}\colon \Sigma^d(\gamma) \to X$ and give a running time estimate.

Algorithm from Lemma 7.13.

The algorithm accepts an element $\gamma \in \overline{\Omega}X_{d-1}$ such that $d_i\gamma = 1 \in \overline{\Omega}X$ i.e. a spherical element. We divide the algorithm into four steps that correspond to the four step factorization in the following diagram:



$\mathsf{Dom}(\gamma)$: We interpret γ as an element in IX and construct $\gamma_{\mathsf{map}}\colon \mathsf{Dom}(\gamma) \to X$. This is clearly linear in the size of γ.

$\overline{\mathsf{Dom}}(\gamma)$: The algorithm checks, whether an edge $[y,j]^\epsilon$ in $d_{i_1}d_{i_2}\dots d_{i_\ell}\gamma$, where $0 \le i_1 < i_2 < \dots < i_\ell < (d-\ell-2)$ is *compressible*, i.e. $y = s_j d_j y$. If this is the case, add a corresponding relation on the preimages: $\alpha^{-1}(y) \sim s_j d_j \alpha^{-1}(y)$. Factoring out the relations, we get a map $\gamma_{\mathsf{c}}\colon \overline{\mathsf{Dom}}(\gamma) \to X$.

Although the number of faces we have to go through is exponential in d, this is not a problem, since d is deemed as a constant in the algorithm and so is 2^d. Hence the number of operations is again linear in the size of γ.

$\overline{\overline{\mathsf{Dom}}}(\gamma)$: Let $k < d$. We know that $\overline{d_k\gamma} = 1_*$, so after removing all compressible elements from the path $d_k\gamma$, it will contain a sequence of pairs $([y_i,j_i]^{\epsilon_i}, [y_i,j_i]^{-\epsilon_i})$ such that, after removing all $[y_u,j_u]^{\pm 1}$ for all $u < v$, then $[y_v,j_v]^{\epsilon_v}$ and $[y_v,j_v]^{-\epsilon_v}$ are next to each other.[17] Each such pair $([y_i,j_i]^{\epsilon_i}, [y_i,j_i]^{-\epsilon_i})$ corresponds to a pair of indices (l_i, m_i) corresponding to the positions of those edges in $d_k\gamma$. These sequences are not unique, but can be easily found in time linear in length(γ). Then we glue $\alpha_{l_i}^{-1}(y_i)$ with $\alpha_{m_i}^{-1}(y_i)$ for all i. Performing such identifications for all k defines the new simplicial set $\overline{\overline{\mathsf{Dom}}}(\gamma)$.

$\Sigma^d(\gamma)$: It remains to identify $\alpha^{-1}(\mathsf{source}\,\gamma)$ and $\alpha^{-1}(\mathsf{target}\,\gamma)$ with the appropriate degeneracy of the (unique) basepoint. The resulting space $|\Sigma^d(\gamma)|$ is a d-sphere.

[17] For example, $[a,1][b,2][b,2]^{-1}[a,1]^{-1}$ can be split into a sequence $([b,2],[b,2]^{-1}), ([a,1],[a,1]^{-1})$.

Converting Σ^d into a simplicial complex.

Let $f : \Sigma^d \to X$ be a simplicial representative of an element $\alpha \in \pi_d(X)$ produced by Berger's algorithm. The simplicial set Σ^d consists of a finite collection of standard d-simplices, glued together along facets, and with identifications on the boundary turning it into a sphere. In general, the boundary consists of simplices, identified to the base point $*$, and pairs of simplices aa^{-1} which are identified. In Figure 3 is shown an example of Σ^2.

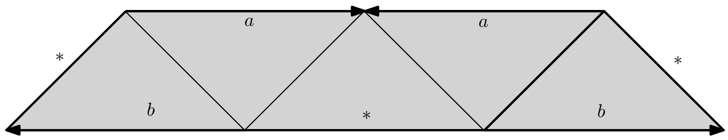


Figure 3: A band of 2-simplices forming a 2-sphere.

This model is not a simplicial complex, but can be easily turned into one. From the construction we present below, it will be clear that the number of simplices we add in order to achieve that is at most polynomial in size(Σ).

First we will show how this is done in dimension 2. We proceed in two steps. We begin by adding additional simplices in order to eliminate cancelling pairs, so the whole boundary of the sphere is identified to a point. Let aa^{-1} be a pair of cancelling edges. Without loss of generality, we can assume that either they are neighbouring edges, or they are separated by a degeneracy of the basepoint $*$ [18]. The former case is illustrated in Figure 4 and the latter case in Figure 5. In order to eliminate the pair of neighbouring cancelling edges aa^{-1}, we add two simplices A_1 and A_2 to Σ^2, both having faces $(a, a, s_0 d_1 a)$, so they are situated as in Figure 4. The simplicial set we obtain has in its boundary instead of the cancelling pair aa^{-1}, two edges identified to the basepoint $*$. However, it might not be a sphere any more because of the initial identification aa^{-1}. The final step is to replace this pair by new edges a_1 and a_2, and change the faces of all 2-simplices accordingly. Thus, we obtain a simplicial set $\bar{\Sigma}^2$ from Σ^2, by eliminating the pair aa^{-1}. We define a map $\bar{f} : \bar{\Sigma}^2 \to X$ in the obvious way, so it is compatible with $f : \Sigma^2 \to X$. In our example in Figure 4, that would mean that we $\bar{f}(a_1) = \bar{f}(a_2) = a$ and $\bar{f}(A_1) = \bar{f}(A_2) = s_0 f(a)$. The simplices we added were chosen in such a way, that they could be mapped to corresponding degeneracies in X in a way, compatible with the map f.

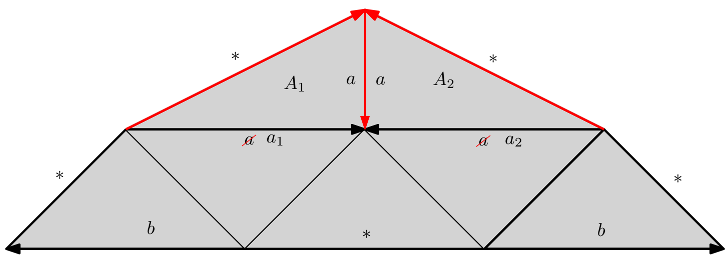


Figure 4: Eliminating the neighbouring cancelling pair aa^{-1}.

The process of eliminating a cancelling pair bb^{-1}, which is separated by a degenerate simplex, is demonstrated in Figure 5. We proceed in the same way, but the elimination requires additional steps.

[18]There might be other pairs separating them, but we would deal with them first.

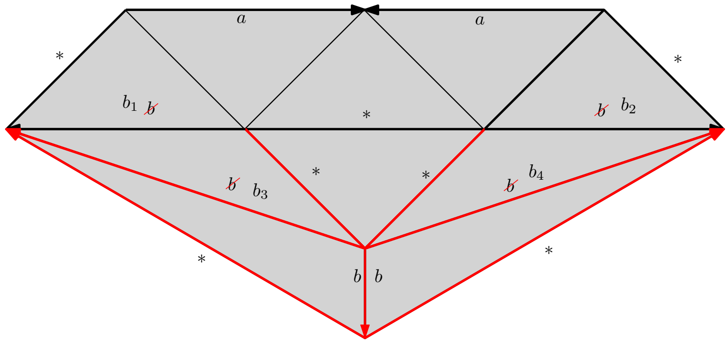
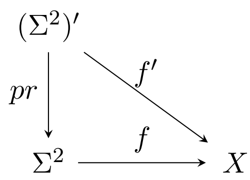


Figure 5: Eliminating the cancelling pair bb^{-1}, separated by a degenerate edge.

Eliminating all pairs of cancelling edges, we obtain a model $(\Sigma^2)'$ of the 2-sphere, consisting of a 2-disc with all its boundary identified to a point, and a map $f' : (\Sigma^2)' \to X$. Moreover, they fit in the commutative diagram:



Here $pr : (\Sigma^2)' \to \Sigma^2$ identifies back all the edges and 2-simplices we changed in order to obtain $(\Sigma^2)'$ out of Σ^2. This map is clearly a homotopy equivalence, so f' will represent the same element of $\pi_2(X)$.

This concludes the first step.

In the second step, we first change the boundary of $(\Sigma^2)'$ so that it consists of non-degenerate edges only, and has no identifications. However, we still map it to the basepoint of X under the map $f' : (\Sigma^2)' \to X$. The changed $(\Sigma^2)'$ is now a simplicial complex, with geometric realisation homeomorphic to a 2-disc. In order to turn it into a sphere, we add one more vertex v, and glue in the triangulated cone over $\partial(\Sigma^2)'$. We denote the newly obtained simplicial complex by $(\Sigma^2)^{sc}$. Finally, we define $f^{sc} : (\Sigma^2)^{sc} \to X$ to simply send $\partial(\Sigma^2)'$ together with the whole cone on it to the base point of X, and coincide with f' on the rest of $(\Sigma^2)^{sc}$.

For the general case $d \geq 3$, we proceed in virtually the same way. Assume Berger's algorithm produces a representative $f : \Sigma^d \to X$ of an element in $\pi_d(X)$. Pick a pair of cancelling facets aa^{-1} of Σ^d. Once again, either they are neighbouring or they are separated by degeneracies of the base point. In the former case, we will add to Σ^d two additional d-simplices A_1 and A_2 which would cancel the pair, then we change the initial facets to a_1 and a_2, in order to avoid unwanted identifications. According to Berger's algorithm, the sphere Σ^d will have at least 2 of its boundary simplices being degeneracies of the base point $*$ namely, the ones corresponding to the source and target of the element of ΩX to which Σ^d corresponds. Thus, after we eliminate all cancelling pairs of facets aa^{-1} of Σ^d, we will obtain a simplicial set $(\Sigma^d)'$, which consists of a triangulated d-disc having its boundary identified to a point. Changing this boundary and glueing in the cone over it will produce the desired simplicial complex $(\Sigma^d)^{sc}$, and the map $f^{sc} : (\Sigma^d)^{sc} \to X$ can be defined in the same way as in dimension 2.

The construction above can be summarised in the following lemma.

Lemma 7.14. *Let X be a 0-reduced simplicial set, Σ be the model of the d-sphere and $f : \Sigma \to X$ the output of Berger's algorithm above. Then we can compute a simplicial complex Σ^{sc} with prescribed orientations of all simplices, and maps*

- $pr : \Sigma^{sc} \to \Sigma$ *and*
- $f^{sc} : \Sigma^{sc} \to X$,

such that $|f| \circ |pr|$ is homotopic to $|f^{sc}|$.

8 Polynomial-Time Loop Contraction in F_d

In this section, we show that simplicial sets F_k, $2 \le k \le n$ constructed algorithmically in Section 6 have polynomial-time contractible loops, thus proving Lemma 6.3. We first give the contraction on F_2 and show that the contraction $F_i, i > 3$ follows from the contraction on F_3. The majority of the effort in this section is then concentrated on the description of the contraction c_0 on F_3.

Notation. We will further use the following shorthand notation: For a 0-reduced simplicial set X we will denote the iterated degeneracy $s_0 \cdots s_0 *$ of its unique basepoint $*$ by $*$ and we set $\pi_i = \pi_i(X)$. For any Eilenberg-Maclane space $K(\pi_i, i-1)$, $i \ge 2$, we denote its basepoint and its degeneracies by 0. From the context, it will always be clear which simplicial set we refer to.

Loop contraction on F_2. Assuming that X is a 0-reduced, 1-connected simplicial set with a given algorithm that computes the contraction on loops $c_0 \colon (GX)_0 \to (GX)_1$, the contraction c_0 on F_2 is automatically defined, as $X = F_2$.

Loop contraction on F_i, $i > 3$. Suppose we have defined the contraction on the generators of $G_0(F_3)$. i.e. for any $(x, k) \in (X \times_{\tau'} K(\pi_2, 1))_1$ we have

$$c_0(\overline{(x,k)}) = \overline{(x_1,k_1)}^{\epsilon_1} \cdots \overline{(x_n,k_n)}^{\epsilon_n} \qquad (x_j,k_j) \in (F_3)_2, \epsilon_j \in \mathbb{Z}, 1 \le j \le n$$

such that $d_0 c_0(\overline{(x,k)}) = \overline{(x,k)}$ and $d_1 c_0(\overline{(x,k)}) = 1$. In detail, we get the following:

$$\overline{(x,k)} = d_0 c_0(\overline{(x,k)}) = \overline{(d_0x_1, d_0k_1)}^{\epsilon_1} \cdots \overline{(d_0x_n, d_0k_n)}^{\epsilon_n} \tag{6}$$

$$\begin{aligned} 1 = d_1 c_0(\overline{(x,k)}) = {} & \left(\overline{(d_2x_1, \tau'(x_1)d_2k_1)}^{-1} \cdot \overline{(d_1x_1, d_1k_1)}\right)^{\epsilon_1} \cdots \\ & \left(\overline{(d_2x_n, \tau'(x_n)d_2k_n)}^{-1} \cdot \overline{(d_1x_n, d_1k_n)}\right)^{\epsilon_n} \end{aligned} \tag{7}$$

We now aim to give a reduction on the generators of $G_0(F_i)$, $i > 3$. Simplicial set F_i is an iterated twisted product of the form

$$\big(((X \times_{\tau'} K(\pi_2, 1)) \times_{\tau'} K(\pi_3, 2)) \times_{\tau'} \cdots \times_{\tau'} K(\pi_{i-2}, i-3)\big) \times_{\tau'} K(\pi_{i-1}, i-2)$$

As simplicial sets $K(\pi_{i-1}, i-2)$ are 1-reduced for $i > 3$, we can identify elements of $(F_i)_1$ with vectors $(x, k, 0, \ldots, 0)$, where $k \in K(\pi_2, 1)_1, x \in X_1$. We further shorthand the series of $i-3$ zeros in the vector with $\mathbf{0}$. Hence generators $G_0(F_i)$ are of the form $\overline{(x, k, \mathbf{0})}$. The 1-reducedness also implies that $\tau'(\alpha) = 0$ whenever $\alpha \in (F_i)_2$, $i > 2$.

Finally, we set

$$c_0(\overline{(x,k,\mathbf{0})}) = \overline{(x_1,k_1,\mathbf{0})}^{\epsilon_1} \cdots \overline{(x_n,k_n,\mathbf{0})}^{\epsilon_n} \qquad (x_j,k_j,\mathbf{0}) \in (F_i)_2, \epsilon_j \in \mathbb{Z}, 1 \le j \le n$$

The (almost) freeness of $G_0(F_i)$, the fact that $K(\pi_{i-1}, i-2)$ are 1-reduced for $i > 3$ and equations (6), (7) give that $d_0 c_0(\overline{(x,k,\mathbf{0})}) = \overline{(x,k,\mathbf{0})}$ and $d_1 c_0(\overline{(x,k,\mathbf{0})}) = 1$.

Before the definition of contraction on simplicial set F_3, we remind the basic facts involving the simplicial model of Eilenberg-MacLane spaces we are using.

Eilenberg–MacLane spaces. As noted in Section 5, given a group π and an integer $i \geq 0$ an Eilenberg–MacLane space $K(\pi, i)$ is a space satisfying

$$\pi_j(K(\pi, i)) = \begin{cases} \pi & \text{for } j = i, \\ 0 & \text{else.} \end{cases}$$

In the rest of this section, by $K(\pi, i)$ we will always mean the simplicial model which is defined in [33, page 101]

$$K(\pi, i)_q = Z^i(\Delta^q; \pi),$$

where $\Delta^q \in \mathsf{sSet}$ is the standard q-simplex and Z^i denotes the cocycles. This means that each q–simplex is regarded as a labelling of the i–dimensional faces of Δ^q by elements of π such that they add up to $0 \in \pi$ on the boundary of every $(i+1)$-simplex in Δ^q, hence elements of $K(\pi, q)_q$ are in bijection with elements of π. The boundary and degeneracy operators in $K(\pi, k)$ are given as follows: For any $\sigma \in K(\pi, i)_q$, $d_j(\sigma) \in K(\pi, k)_{q-1}$ is given by a restriction of $\sigma \in K(\pi, i)$ to the j-th face of Δ^q. To define the degeneracy we first introduce mapping $\eta_j \colon \{0, 1, \ldots, q+1\} \to \{0, 1, \ldots, q\}$ given by

$$\eta_j(\ell) = \begin{cases} \ell & \text{for } \ell \leq j, \\ \ell - 1 & \text{for } \ell > j. \end{cases}$$

Every mapping η_j defines a map $C^*(\eta_j) \colon C^*(\Delta^q) \to C^*(\Delta^{q+1})$.The degeneracy $s_j\sigma$ is now defined to be $C^*(\eta_j)(\sigma)$ (see [33, 23]).

It follows from our model of Eilenberg-MacLane space, that elements of $K(\pi_2, 1)_2$ can be identified with labellings of 1-faces of a 2-simplex by elements of π_2 that sum up to zero.

As π_2 is an Abelian group, we use the additive notation for π_2. We identify the elements of $K(\pi_2, 1)_2$ with triples (k_0, k_1, k_2), $k_i \in \pi_2$, $0 \leq i \leq 2$, such that $k_0 - k_1 + k_2 = 0 \in \pi_2$.

Loop contraction on F_3. Let X be a 0-reduced, 1-connected simplicial set with a given algorithm that computes the contraction on loops $c_0 \colon (GX)_0 \to (GX)_1$.

In the rest of the section, we will assume $x \in X_1$. Then by our assumptions $c_0\overline{x} = \overline{y_1}^{\epsilon_1} \cdots \overline{y_n}^{\epsilon_n}$, where $y_i \in X_2, \epsilon_i \in \mathbb{Z}$, $1 \leq i \leq n$. Let $k_i = \tau'(y_i)$.

We first show that in order to give a contraction on elements of the form $\overline{(x, 0)}$ and $\overline{(x, k)}$, it suffices to have the contraction on elements of the form $\overline{(*, k)}$:

Contraction on element $(x, 0)$. Let $\overline{(x, 0)} \in G_0(F_3)$. We define

$$c_0\overline{(x, 0)} = \prod_{i=1}^{n} \left(c_0\overline{(*, k_i)}^{-1} \overline{(s_1 d_2 y_i, (k_i, k_i, 0))} \cdot \overline{(y_i, 0)} \right)^{\epsilon_i}.$$

Contraction on element (x, k). Suppose $\overline{(x, k)} \in (GF_3)_0$. The formula for the contraction is given using the formulae on contraction on $\overline{(x, 0)}$ and $\overline{(*, k)}$:

$$c_0\overline{(x, k)} = \overline{(s_0 x, (k, 0, -k))} \cdot s_0\overline{(x, 0)}^{-1} \cdot s_0\overline{(*, -k)} \cdot c_0(\overline{(*, -k)})^{-1} \cdot c_0(\overline{(x, 0)})$$

Contraction on element $(*, k)$. We formalize the existence of the contraction as Proposition 8.4 given at the end of this section. Due to the fact that the proof is rather technical, we need to define and prove some preliminary results first:

Definition 8.1. *Let $Z = \{z \in (GF_3)_1 \mid d_0 z = 1\}$ and let $W = \{d_1 z \mid z \in Z\}$ We define an equivalence relation $\sim$ on the elements of W in the following way: We say that $w \sim w'$ if there exists $z \in Z$, $\alpha, \beta \in (GF_3)_1$ such that $d_1 z = w$, $\alpha z \beta \in Z$ and $d_1(\alpha z \beta) = w'$.*

Lemma 8.2. *Let $w \in W$ such that*

1. $w = \overline{(x,k)}^{\epsilon} \cdot \alpha$, *where* $\alpha \in (GF_3)_1$ *Then* $w = \overline{(x,k)}^{\epsilon} \cdot \alpha \sim \alpha \cdot (x,k)^{\epsilon} = w'$.
2. $w = \overline{(*,k)}^{\epsilon} \cdot \alpha$, *where* $\alpha \in (GF_3)_0$. *Then* $w \sim w' = \overline{(*,-k)}^{-\epsilon} \cdot \alpha$.
3. $w = \overline{(*,-k)}^{-1}(x,0) \cdot \alpha$, *where* $\alpha \in (GF_3)_0$. *Then* $w \sim w' = \overline{(x,k)} \cdot \alpha$.
4. $w = \overline{(x,0)}^{-1}\overline{(x,k)} \cdot \alpha$, *where* $\alpha \in (GF_3)_0$. *Then* $w \sim w' = \overline{(*,k)} \cdot \alpha$.
5. $w = \overline{(*,-l)}^{-1}\overline{(*,k)} \cdot \alpha$, *where* $\alpha \in (GF_3)_0$. *Then* $w \sim w' = \overline{(*,k+l)} \cdot \alpha$.

Proof. In all cases, we assume $z \in Z$ such that $d_1 z = w$ and we give a formula for $z' \in Z$ with $d_1 z' = w'$:

1. $z' = s_0\overline{(x.k)}^{-\epsilon} \cdot z \cdot s_0\overline{(x,k)}^{\epsilon}$.
2. $z' = \overline{(*,(k,0,-k))}^{\epsilon} \cdot (s_0\overline{(*,k)})^{-\epsilon} \cdot z$.
3. $z' = (s_0\overline{(x,k)}) \cdot \overline{(s_0 x, (k,0,-k))}^{-1} \cdot z$.
4. $z' = (s_0\overline{(*,k)})\overline{(s_1 x, (k,k,0))}^{-1} \cdot z$.
5. $z' = \overline{(s_0(*,k+l))(*,(k+l,k,-l))}^{-1} \cdot z$.

□

Lemma 8.3. *Let $z \in (GF_3)_1$, $z \in Z$ with*

$$d_1 z = w = \overline{(*,-k_1)}^{-1} \cdot \overline{(x_1,0)}^{\epsilon_1} \cdots \overline{(*,-k_n)}^{-1} \cdot \overline{(x_n,0)}^{\epsilon_n}$$

where $\overline{x_1}^{\epsilon_1} \cdots \overline{x_n}^{\epsilon_n} = 1$ in GX_0, $x_i \in X$, $k_i \in \pi_2(X)$, $\epsilon_i \in \{1,-1\}$, $1 \le i \le n$. Then $w \sim \overline{(\sum_{i=1}^{n} k_i, *)}$.

Proof. We achieve the proof using a sequence of equivalences given in Lemma 8.2. Without loss of generality we can assume that $x_1 = x_2^{-1}$ and $\epsilon_1, \epsilon_2 = 1$ (If this is not the case, we can use rule (1) and/or relabel the elements). Using (1) gives us

$$\begin{aligned} w =& \overline{(*,-k_1)}^{-1} \cdot \overline{(x_2,0)}^{-1} \cdot \overline{(*,-k_2)}^{-1} \cdot \overline{(x_2,0)} \cdots \overline{(*,-k_n)}^{-1} \cdot \overline{(x_n,0)}^{\epsilon_n} \\ \sim& \overline{(*,-k_2)}^{-1} \cdot \overline{(x_2,0)} \cdots \overline{(*,-k_n)}^{-1} \cdot \overline{(x_n,0)}^{\epsilon_n} \cdot \overline{(*,-k_1)}^{-1} \cdot \overline{(x_2,0)}^{-1}. \end{aligned}$$

Then successive use of (3),(1),(4), (1) and finally (5) gives us

$$\begin{aligned} w \sim& \overline{(x_2,k_2)} \cdots \overline{(*,-k_n)}^{-1} \cdot \overline{(x_n,0)}^{\epsilon_n} \cdot \overline{(*,-k_1)}^{-1} \cdot \overline{(x_2,0)}^{-1}. \\ \sim& \overline{(x_2,0)}^{-1} \cdot \overline{(x_2,k_2)} \cdots \overline{(*,-k_n)}^{-1} \cdot \overline{(x_n,0)}^{\epsilon_n} \cdot \overline{(*,-k_1)}^{-1} \\ \sim& \overline{(*,k_2)} \cdots \overline{(*,-k_n)}^{-1} \cdot \overline{(x_n,0)}^{\epsilon_n} \cdot \overline{(*,-k_1)}^{-1} \\ \sim& \overline{(*,k_1+k_2)} \cdot \overline{(*,-k_3)}^{-1} \cdot \overline{(x_3,0)} \cdots \overline{(*,-k_n)}^{-1} \cdot \overline{(x_n,0)}^{\epsilon_n} \end{aligned}$$

multiple use or rules (2) and (1) and gives us

$$w \sim \overline{(*, -k_1-k_2-k_3)}^{-1} \cdot \overline{(x_3,0)} \cdots \overline{(*,-k_n)}^{-1} \cdot \overline{(x_n,0)}^{\epsilon_n}$$

So far, we have produced some element $z' \in Z \subseteq (GF_3)_1$ such that $d_0 z' = 1$,

$$d_1 z' = \overline{(*, -k_1-k_2-k_3)}^{-1} \cdot \overline{(x_3,0)} \cdots \overline{(*,-k_n)}^{-1} \cdot \overline{(x_n,0)}^{\epsilon_n}$$

and further $\overline{x_3}^{\epsilon_3} \cdots \overline{x_n}^{\epsilon_n} = 1$ in GX_0.

It follows that the construction described above can be applied iteratively until all elements $\overline{(x_i,0)}$ are removed and we obtain $w \sim \overline{(-\sum_{i=1}^n k_i, *)}^{-1} \sim \overline{(\sum_{i=1}^n k_i, *)}$. □

Proposition 8.4. *Let $k \in \pi_2(X)$. Then there is an algorithm that computes an element $z \in (GF_3)_1$ such that $d_0 z = \overline{(*,k)}$ and $d_1 z = 1$.*

Proof. Given an element $k \in \pi_2 \cong H_2(X)$, one can compute a cycle $\gamma \in Z_2(X)$ such that

$$[\gamma] = k \in \pi_2(X) \cong H_2(X) \cong H_2(K(\pi_2, 2)) \cong \pi_2(K(\pi_2,2)),$$

were the middle isomorphism is induced by φ_2 and the other isomorphisms follow from Hurewicz theorem.

If one considers $\gamma \in \widetilde{AX}_1$ then by Lemma 7.8 one can algorithmically compute a spherical element $\gamma' = \overline{y_1}^{\epsilon_1} \cdots \overline{y_n}^{\epsilon_n} \in \widetilde{GX}_1$ where $y_i \in X_2$ and $\tau' y_i = k_i \in \pi_2(X)$, such that $d_0\gamma' = 1 = d_1\gamma'$ and $\sum_{i=1}^n \epsilon_i \cdot k_i = k$.

We define $z' \in (GF_3)_1$ by

$$z' = (\prod_{i=1}^n \overline{(s_0 d_0 y_i, (k_i, 0, -k_i))}^{\epsilon_i}) \cdot (\prod_{i=1}^n \overline{(y_i, (k_i, 0, -k_i))}^{\epsilon_i})^{-1}.$$

Observe that $d_0(z') = 0$ and

$$d_1 z' = \left(\overline{(*,-k_1)}^{-1} \cdot \overline{(d_0 y_1, 0)}\right)^{\epsilon_1} \cdots \left(\overline{(*,-k_n)}^{-1} \cdot \overline{(d_0 y_n, 0)}\right)^{\epsilon_n}.$$

We apply Lemma 8.3 on z' and get an element $z'' \in (GF_3)_1$ with the property $d_0 z'' = 1$ and $d_1 z'' = \overline{(*,k)}$. We define $z = s_0\overline{(*,k)} \cdot (z'')^{-1}$. Thus $d_0 z = \overline{(*,k)}$ and $d_1 z = 1$. □

Computational complexity. We first observe that that formulas for c_0 on a general element $\overline{(x,k)}$ depend polynomially on the size of $c_0(\overline{x})$ and the size of contractions on $\overline{(*,k)}$. Hence it is enough to analyse the complexity of the algorithm described in Proposition 8.4:

The computation of γ' is obtained by the polynomial-time Smith normal form algorithm presented in [25] and the polynomial-time algorithm in Lemma 7.8. The size of z' depends polynomially (in fact linearly) on size of γ'. The algorithm described in Lemma 8.3 runs in a linear time in the size of z'.

To sum up, the algorithm computes the formula for contraction on the elements of GF_i in time polynomial in the input (size X + size $c_0(GX)$).

9 Reconstructing a Map to the Original Simplicial Complex

This section contains the proof of Lemma 6.4.

Edgewise subdivision of simplicial complexes. In [12], the authors present, for $k \in \mathbb{N}$, the *edgewise subdivision* $\mathrm{Esd}_k(\Delta^m)$ of an m-simplex Δ^m that generalizes the two-dimensional sketch displayed in Figure 6. This subdivision has several nice properties: in particular, the number of simplices of $\mathrm{Esd}_k(\Delta^m)$ grows polynomially with k. Explicitly, the subdivision can be represented as follows.

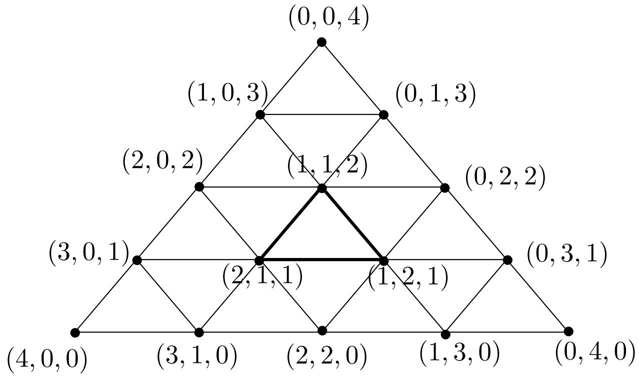


Figure 6: Edgewise subdivision of a 2-simplex for $k = 4$. In this case, there exists a copy of the 2-simplex completely in the "interior", defined by vertices $(2, 1, 1)$, $(1, 2, 1)$ and $(1, 1, 2)$. All other vertices are at the "boundary": more formally, their coordinatates contain a zero.

- The vertices of $\mathrm{Esd}_k(\Delta^m)$ are labeled by coordinates $(a_0, \ldots, a_m)$ such that $a_j \geq 0$ and $\sum_j a_j = k$.
- Two vertices $(a_0, \ldots a_m)$ and $(b_0, \ldots, b_m)$ are *adjacent*, if there is a pair $j < k$ such that $|b_j - a_j| = |b_k - a_k| = 1$ and $a_i = b_i$ for $i \neq j, k$.
- Simplices of $\mathrm{Esd}_k(\Delta^m)$ are given by tuples of vertices such that each vertex of a simplex is adjacent to each other vertex.

We define the *distance* of two vertices to be the minimal number of edges between them. An edgewise k-subdivision of Δ^m induces an edgewise k-subdivision of all faces, hence we may naturally define an edgewise subdivision of any simplicial complex.

Constructing the map $\mathbf{Esd}_k(\Sigma) \to X^{sc}$. Let R be a chosen root in the tree T. We denote the tree-distance of a vertex W from R by $\mathrm{dist}_T(W)$. Let

$$l := \max\{\mathrm{dist}_T(V) : \ V \text{ is a vertex of } X^{sc}\}$$

be the maximal tree-distance of some vertex from R. For each vertex V of X^{sc}, there is a unique path in the spanning tree that goes from R into V. Further, we define the maps $M(j) : (X^{sc})^{(0)} \to (X^{sc})^{(0)}$ from vertices of X^{sc} into vertices of X^{sc} such that

- $M(j)(V) := V$ if $j \geq \mathrm{dist}_T(V)$, and
- $M(j)(V)$ is the vertex on the unique tree-path from R to V that has tree-distance j from R, if $j < \mathrm{dist}_T(V)$.

If, for example, $R - U - V - W$ is a path in the tree, then $M(0)(W) = R$, $M(1)(W) = U$ etc. Clearly, $M(l) = M(l + 1) = \ldots$ is the identity map, as l equals the longest possible tree-distance of some vertex.

Assume that d is the dimension of Σ and $k := l(d+1)+1$. We will define $f' : \mathrm{Esd}_k(\Sigma) \to X^{sc}$ simplexwise. Let $\tau \in \Sigma$ be an m-simplex and $f(\tau) = \tilde{\sigma} \in X$ be its image in the simplicial set X. If σ is the degeneracy of the base-point, then we define $f'(x) := R$ for all vertices x of $\mathrm{Esd}_k(\tau)$: in other words, f' will be constant on the subdivision of τ. Otherwise, $\tilde{\sigma}$ is not the degeneracy of a point and has a unique lift $\sigma \in X^{ss}$. Let $(V_0, \ldots, V_m)$ be the vertices of σ (order given by orientation): these vertices are not necessarily different, as σ may be degenerate.

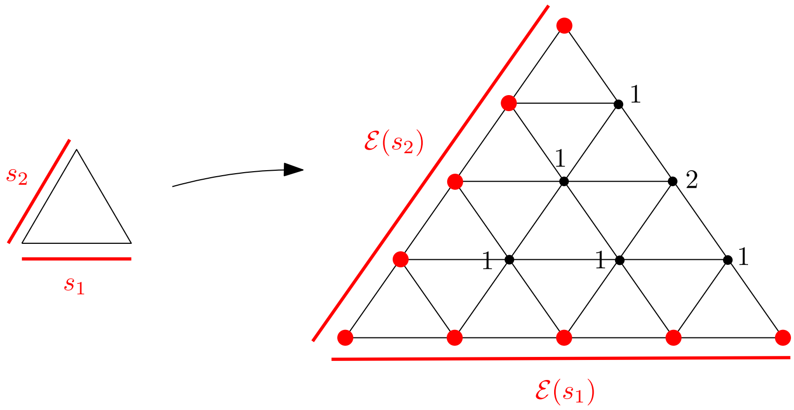


Figure 7: Illustration of extended faces. Here $S = \{s_1, s_2\}$ corresponds to the lower- and left-face of a 2-simplex. The extended faces $\mathcal{E}(s_1)$ and $\mathcal{E}(s_2)$ are sets of vertices of $\mathrm{Esd}_k(\Delta^2)$ that are on the lower- and left- boundary. The corresponding extended tree $\mathcal{E}(T)$ is the union of all these vertices. The integers indicate edge-distances dist_{ET} of vertices in $\mathrm{Esd}_k(\Delta^2)$ from $\mathcal{E}(T)$.

In the algorithm, we will need to know which faces of σ are in the tree T. We formalize this as follows: let $S \subseteq 2^m$ be the family of all subsets of $\{0, 1, \ldots, m\}$ such that

- For each $\{i_0, \ldots, i_j\} \in S$, $\{V_{i_0}, \ldots, V_{i_j}\}$ is in the tree (that is, it is either an edge or a single vertex),
- Each set in S is maximal wrt. inclusion.

Elements of S correspond to maximal faces of σ that are in the tree, in other words, to faces of $\tilde{\sigma}$ that are degeneracies of the base-point.

Definition 9.1. *Let Δ^m be an oriented m-simplex, represented as a sequence of vertices $(e_0, \ldots, e_m)$. For any face $s \subseteq \{e_0, \ldots, e_m\}$, we define the* extended face $\mathcal{E}(s)$ in $\mathrm{Esd}_k(\Delta^m)$ *to be the set of vertices $(x_0, \ldots, x_m)$ in $\mathrm{Esd}_k(\Delta^m)$ that have nonzero coordinates only on positions i such that $e_i \in S$.*

The geometric meaning of this is illustrated by Figure 7.

Definition 9.2. *For $S \subseteq 2^m$, we define the* extended tree $\mathcal{E}(T)$ *to be the union of the extended faces $\mathcal{E}(s)$ in $\mathrm{Esd}_k(\Delta^m)$ for all $s \in S$. The edge-distance of a vertex x in $\mathrm{Esd}_k(\Delta^m)$ from $\mathcal{E}(T)$ will be denoted by $\mathrm{dist}_{ET}(x)$.*

In words, $\mathcal{E}(T)$ it is the union of all vertices in parts of the boundary of $\mathrm{Esd}_k(\Delta^m)$ that correspond to the faces of σ that are in the tree, see Fig. 7. The number $\mathrm{dist}_{ET}(x)$ is the distance to x from those boundary parts that correspond to faces of σ that are in the tree.

To define a simplicial map from $\mathrm{Esd}_k(\tau)$ to X^{sc}, we need to label vertices of $\mathrm{Esd}_k(\tau)$ by vertices of X^{sc} such that the induced map takes simplices in $\mathrm{Esd}_k(\tau)$ to simplices in X^{sc}. Recall that $V_0, \ldots, V_m$ are the vertices of σ. For $x = (x_0, \ldots, x_m)$, we denote by $\arg\max x$ the smallest index of a coordinate of x among those with maximal value (for instance, $\arg\max (4, 2, 1, 4, 0) = 0$, as the first 4 is on position 0). The geometric meaning of $V_{\arg\max x}$ is illustrated by Figure 8.

Now we define the map $f' : \mathrm{Esd}_k(\tau) \to X^{sc}$ by mapping each vertex x via the formula

$$x = (x_0, \ldots, x_m) \mapsto M(\mathrm{dist}_{ET}(x))(V_{\arg\max\ x}). \tag{8}$$

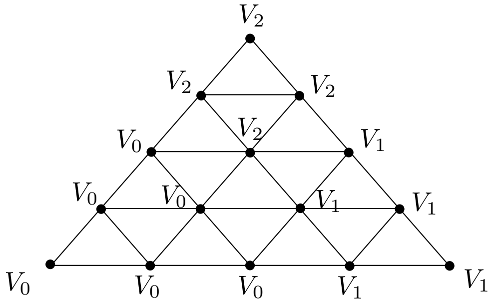


Figure 8: Labelling vertices of $\mathrm{Esd}_k(\Delta^2)$ by $V_{\arg\max x}$.

Geometrically, most vertices x will be simply mapped to V_j for which the j'th coordinate of x is dominant. In particular, a unique m-simplex "most in the interior of $\mathrm{Esd}_k(\tau)$" with coordinates

$$\begin{pmatrix} j+1 \\ j \\ \cdots \\ j \\ \hline j+1 \\ \cdots \\ j+1, \end{pmatrix}^T, \begin{pmatrix} j \\ j+1 \\ \cdots \\ j \\ \hline j+1 \\ \cdots \\ j+1, \end{pmatrix}^T, \ldots, \begin{pmatrix} j \\ j \\ \cdots \\ j+1 \\ \hline j+1 \\ \cdots \\ j+1, \end{pmatrix}^T, \begin{pmatrix} j \\ j \\ \cdots \\ j \\ \hline j+2 \\ \cdots \\ j+1, \end{pmatrix}^T, \ldots, \begin{pmatrix} j \\ j \\ \cdots \\ j \\ \hline j+1 \\ \cdots \\ j+2, \end{pmatrix}^T \tag{9}$$

for suitable j will be labeled by $V_0, V_1, \ldots, V_m$; in other words, it will be mapped to σ.[19]

However, vertices x close to those boundary parts of $\mathrm{Esd}_k(\tau)$ that correspond to the tree-parts of σ, will be mapped closer to the root R and all the extended tree $\mathcal{E}(T)$ will be mapped to R. One illustration is in Figure 9.

Computational complexity. Assuming that we have a given encoding of Σ, f, X, X^{sc} and a choice of T and R, defining a simplicial map $f' : \mathrm{Esd}_k(\Sigma) \to X^{sc}$ is equivalent to labelling vertices of $\mathrm{Esd}_k(\Sigma)$ by vertices of X^{sc}. Clearly, the maximal tree-distance l of some vertex depends only polynomially on the size of X^{sc} and can be computed in polynomial time, as well as the maps $M(0), \ldots, M(l)$. Whenever $j > l$, we can use the formula $M(j) = \mathrm{id}$. Further, $k = l(d+1)+1$ is linear in l, assuming the dimension d is fixed. If $\tau \in \Sigma$ is an m-simplex, then the number of vertices in $\mathrm{Esd}_k(\tau)$ is polynomial[20] in k, and their coordinates can be computed in polynomial time. Finding the lift σ of $f(\tau) = \tilde{\sigma}$ is at most a linear operation in $\mathrm{size}(X^{sc}) + \mathrm{size}(\tilde{\sigma})$. Converting $\sigma \in X^{ss}$ into an ordered sequence $(V_0, V_1, \ldots, V_m)$ amounts to computing its vertices $d_0 d_1 \ldots \hat{d_i} \ldots, d_m \sigma$, where d_i is omitted. Collecting information on faces of σ that are in the tree and the set of vertices $\mathcal{E}(T)$ is straight-forward: note that assuming fixed dimensions, there are only constantly many faces of each simplex to be checked. If $s = \{i_0, \ldots, i_j\}$ is a face, then the edge-distance of a vertex x from $\mathcal{E}(s)$ equals to $\sum_u x_{i_u}$. Applying formula (8) to x requires to compute the edge-distance of x from $\mathcal{E}(T)$: this equals to the minimum of the edge-distances of x from $\mathcal{E}(s)$ for all faces s of σ that are in the tree. Computing $\arg\max x$ is a trivial operation. Finally, the number of simplices τ of Σ is bounded by the size of Σ, so applying (8) to each vertex of $\mathrm{Esd}_k(\Sigma)$ only requires polynomially many steps in $\mathrm{size}(\Sigma, f, X^{sc}, T, X)$.

[19] If $\dim(\tau) = d$ is maximal, then $j = l$ and this most-middle simplex has particularly nice coordinates $(l+1, l, \ldots, l), \ldots, (l, \ldots, l, l+1)$.

[20] Here the assumption on the fixed dimension d is crucial.

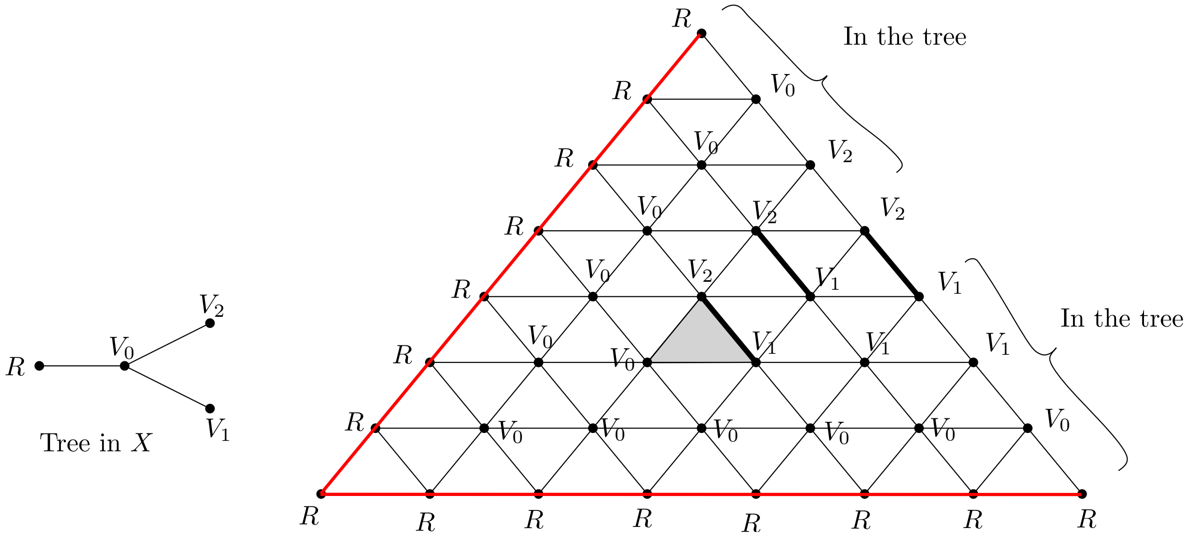


Figure 9: Example of the labelling induced by formula (8). We assume that $f(\tau) = \tilde{\sigma}$ where σ is a simplex of X^{sc} with three different vertices $V_0V_1V_2$. In this example, the tree connects $R - V_0 - V_1$ as well as $R - V_0 - V_2$ and the edge V_1V_2 is *not* in the tree. On the right, we give the induced labelling of vertices of $\mathrm{Esd}_k(\tau)$ which determines a simplicial map to X^{sc}. The bottom and left faces of σ are in the tree, hence the bottom and left extended faces in $\mathrm{Esd}_k(\tau)$ are all mapped into R. The right face of σ is the edge V_1V_2 that is not in the tree: the corresponding right extended face in $\mathrm{Esd}_k(\tau)$ is mapped to a loop $R - V_0 - V_1 - V_2 - V_0 - R$, where V_1V_2 is the only part that is *not* in the tree. The most interior simplex in $\mathrm{Esd}_k(\tau)$ is highlighted and is the only one mapped to σ.

Correctness. What remains is to prove that formula (8) defines a well-defined simplicial map and that $|\mathrm{Esd}_k(\Sigma)| \to |X^{sc}| \to |X|$ is homotopic to $|\Sigma| \to |X|$.

Lemma 9.3. *The above algorithm determines a well-defined simplicial map* $\mathrm{Esd}(\Sigma) \to X^{sc}$.

Proof. First we claim that formula (8) defines a global labelling of vertices of $\mathrm{Esd}_k(\Sigma)$ by vertices of X^{sc}. For this we need to check that if τ' is a face of τ, then (8) maps vertices of $\mathrm{Esd}_k(\tau')$ compatibly. This follows from the following facts, each of them easily checkable:

- If τ' is spanned by vertices of τ corresponding to $s \subseteq \{0, \dots, m\}$, then a vertex $x' := (x_0, \dots, x_j)$ in $\mathrm{Esd}_k(\tau')$ has coordinates x in $\mathrm{Esd}_k(\tau)$ equal to zero on positions $\{0 \dots, m\} \setminus s$ and to $x_0, \dots, x_m$ on other positions, successively.
- $\arg\max x = \arg\max x'$
- The extended tree $\mathcal{E}'(T)$ in $\mathrm{Esd}_k(\tau')$ equals the intersection of the extended tree in $\mathrm{Esd}_k(\tau)$ with $\mathcal{E}(\tau')$
- The distance $\mathrm{dist}_{ET}(x')$ in $\mathrm{Esd}_k(\tau')$ equals $\mathrm{dist}_{ET}(x)$ in $\mathrm{Esd}_k(\tau)$.

Further, we need to show that this labelling defines a well-defined simplicial map, that is, it maps simplices to simplices. We claim that each simplex in $\mathrm{Esd}_k(\tau)$ is mapped either to some subset of $\{V_0, \dots, V_m\}$ or to some edge in the tree T, or to a single vertex.

We will show the last claim by contradiction. Assume that some simplex is *not* mapped to $\{V_0, \dots, V_m\}$, and also it is *not* mapped to an edge of the tree and *not* mapped to a single vertex. Then there exist two vertices x and y in this simplex that are labeled by U and W in X^{sc}, such that either U or W is not in $\{V_0,\dots,V_m\}$, UW is not in the tree, and $U \neq W$.

The fact that at least one of $\{U,W\}$ does not belong to $\{V_0\dots,V_m\}$, implies that $\mathrm{dist}_{ET}(x) < l$ or $\mathrm{dist}_{ET}(y) < l$ (as $M(j)$ maps each $V_{\arg\max x}$ on itself for $j \geq l$).

Without loss of generality, assume that $\arg\max x = 0$ and $\arg\max y = 1$. Then the coordinates of x and y are either

$$x = (j+1, j, x_3, \dots, x_m), \quad y = (j, j+1, x_3, \dots, x_m)$$

such that $x_i \leq j+1$ for all $i \geq 3$, or

$$x = (j, j, x_3, \dots, x_m), \quad y = (j-1, j+1, x_3, \dots, x_m)$$

for some j such that $x_i \leq j$ for all $i \geq 3$.

We claim that $V_0 \neq V_1$ and that the edge V_0V_1 is *not* in the tree. This is because there exists a tree-path from R via U to V_0 and also a tree-path from R via W to V_1 (and $U \neq W$): both $V_0 = V_1$ as well as a tree-edge V_0V_1 would create a circle. In coordinates, this means that vertices $(*,*,0,0,\dots,0)$ are not contained in $\mathcal{E}(T)$, apart of $(k,0,0,\dots,0)$ and $(0,k,0,\dots,0)$. So, any vertex in $\mathcal{E}(T)$ has a zero on either the zeroth or the first coordinate. This immediately implies that $\mathrm{dist}_{ET}(x) \geq j$ and $\mathrm{dist}_{ET}(y) \geq j$. Keeping in mind that coordinates of x (and y) has to sum up to $k = l(d+1)+1$, the smallest possible value of j is $j = l$ (if $m = d$ is maximal), in which case $x = (l+1, l, l, \dots, l)$ and $y = (l, l+1, \dots, l)$. This choice, however, would contradict the fact that either $\mathrm{dist}_{ET}(x) < l$ or $\mathrm{dist}_{ET}(y) < l$. Therefore we have a strict inequality $j > l$. Finally, we derive a contradiction having either $\mathrm{dist}_{ET}(x) \geq j > l > \mathrm{dist}_{ET}(x)$, or a similar inequality for y.

This completes the proof that each simplex is either mapped to a subset of $\{V_0,\dots,V_m\}$ or to an edge in the tree or to a single vertex: the image is a simplex in X^{sc} in either case. $\square$

Lemma 9.4. *The geometric realisations of $pf' : \mathrm{Esd}_k(\Sigma) \to X$ and $f : \Sigma \to X$ are homotopic.*

Proof. First we reduce the general case to the case when all maximal simplices in Σ (wrt. inclusion) have the same dimension d. If this were not the case, we could enrich any lower-dimensional maximal simplex $\tau = \{x_0,\dots,x_j\} \in \Sigma$ by new vertices $y^\tau_{j+1},\dots,y^\tau_d$ and produce a maximal d-simplex $\tilde{\tau} = \{x_0,\dots,x_j,y^\tau_{j+1},\dots,y^\tau_d\}$. Thus we produce a simplicial complex $\tilde{\Sigma} \supseteq \Sigma$ with the required property. Whenever $f(\tau)$ is mapped to $\tilde{\sigma}$ where $\sigma = (V_0,\dots,V_j)$, we define $f(\tilde{\tau})$ to be $s_j^{d-j}\tilde{\sigma}$, a degenerate simplex with lift $(V_0,\dots,V_j,V_j,\dots,V_j)$. The map $f' : \tilde{\Sigma} \to X^{sc}$ is constructed from $f : \tilde{\Sigma} \to X$ as above and if we prove that $|f|$ is homotopic to $|pf'|$ as maps $|\tilde{\Sigma}| \to |X|$, it immediately follows that they are homotopic as maps $|\Sigma| \to |X|$ as well.

Further, assume that all maximal simplices have dimension d. Let $\tau \in \Sigma$ be a d-dimensional simplex and let τ^{int} be the simplex in $\mathrm{Esd}_k(\tau)$ spanned by the vertices

$$(l+1, l, \dots, l), \dots, (l, \dots, l, l+1),$$

that is, the simplex in the interior of τ that is mapped by pf' to $\tilde{\sigma}$. Let $H_\tau(\cdot,1) : |\tau| \to |\tau|$ be a linear map that takes $|\tau|$ to $|\tau^{int}|$ and H_τ a linear homotopy $|\tau| \times [0,1] \to |\tau|$ between the identity and $H_\tau(\cdot,1)$. The composition $|pf'|H_\tau$ then gives a homotopy $|\tau| \times [0,1] \to |X|$ between the restrictions $(|pf'|)|_\tau$ and $(|f|)|_\tau$. For a general $x \in |\Sigma|$, there exists a maximal d-simplex $|\tau|$ such that $x \in |\tau|$ and we define a homotopy

$$(x,t) \mapsto |pf'|H_\tau(x,t).$$

It remains to show that this map is independent on the choice of τ.

Let as denote the (ordered) vertices of τ by $\{v_0, v_1, \ldots, v_d\}$ and let $\delta \subseteq \tau$ be one of its faces: further, let W_i be the vertex of τ^{int} with barycentric coordinates $(l, \ldots, l, l+1, l, \ldots, l)/k$ in $|\tau|$ such that the $l+1$ is in position i. The homotopy H_τ sends points in $|\delta|$ onto the span of points W_i for which $v_i \in \delta$. For each $y \in |\delta|$ and $j \notin \delta$, the j-th coordinate of $H_\tau(y,t)$ is between 0 and l/k. It follows that each $z := H_\tau(x,t)$ is contained in the interior of a unique simplex Δ of $\mathrm{Esd}_k(\tau)$ such that $v_{\arg\max x} \in \delta$ for all vertices x of Δ. For $y \in |\delta|$, the barycentric coordinates of $H_\tau(y,t)$ in positions $j \notin \delta$ are all equal to $t\,(k/l)$.

Let $i_0 < i_1 \ldots < i_k$ be the indices such that $v_{i_j} \in \delta$ and $j_1 < \ldots < j_{d-k}$ be the remaining indices. Let $\tau' = (v'_0, \ldots, v'_d)$ be another d-simplex containing δ as a face. We change the orientation of τ' by ordering its vertices so that vertices of δ are in positions $i_0, \ldots, i_k$—such as it is in τ—and other vertices are on the remaining positions. Let σ, σ' be the lift of $f(\tau)$, $f(\tau')$ respectively, and V_i, V'_i the i-th vertex of σ, σ' respectively.

We define a "mirror" map $m : |\tau| \to |\tau'|$, which to a point with barycentric coordinates $(x_0, \ldots, x_d)$ with respect to τ assigns a point in $|\tau'|$ with the same barycentric coordinates with respect to τ'. Clearly, $H_{\tau'}(y,t) = m(H_\tau(y,t))$ for $y \in |\tau|$ and whenever z is in the interior of a simplex $\Delta \in \mathrm{Esd}_k(\tau)$, then $m(z)$ is in the interior of $m(\Delta)$, where vertices of Δ and $m(\Delta)$ have the same barycentric coordinates with respect to τ and τ', respectively. If, moreover, Δ is such that each of its vertices r have coordinates $\leq l/k$ on positions $j_1, \ldots, j_{d-k}$, then $V_{\arg\max r} = V'_{\arg\max m(r)}$. These properties are still true even if we didn't change the orientation of τ': the coordinates in τ' would be permuted as well as the $(V'_0, \ldots, V'_d)$ but each vertex r of Δ, resp. $m(r)$ of $m(\Delta)$, would still have a unique dominant coordinate with index o, resp. p such that $v_o = v'_p$ is the corresponding vertex in δ and $V_{\arg\max r} = V'_{\arg\max m(r)}$ would still hold.[21]

To summarize these properties, $H_\tau(y,t)$ and $H_{\tau'}(y,t)$

- have the same coordinates wrt. τ, τ', respectively,
- are in the interior of simplices $\Delta \in \mathrm{Esd}_k(\tau)$, $\Delta' \in \mathrm{Esd}_k(\tau')$ whose vertices have the same coordinates wrt. τ, τ', respectively,
- the $\arg\max$ labeling induces the same labeling of vertices of Δ, Δ' by vertices of δ, respectively.

The map pf' takes each k-simplex Δ in $\mathrm{Esd}_k(\tau)$ with vertices t_u labeled by $V_{\arg\max t_u}$ onto $p(V_{\arg\max t_0}, \ldots, V_{\arg\max t_k})$. It follows that $|pf'|H_\tau(y,t) = |pf'|H_{\tau'}(y,t)$ for each $y \in |\delta|$ and $t \in [0,1]$. □

10 Proof of Theorem 2 for Simplicial Sets

Here we show a variant of Theorem 2 for simplicial sets. In particular, we can guarantee an exponential lower bound even for the special case of 1-reduced simplicial sets, where the simply connectedness certificate is provided automatically.

Theorem 2.A. *Let $d \geq 2$ be fixed. Then any algorithm that, for a given $(d-1)$-reduced simplicial set X, computes generators of $\pi_2(X)$ as a simplicial map $\Sigma_k \to X$ where Σ_k is a simplicial set with $|\Sigma_k| = S^d$, has complexity at least exponential in the size of X.*

The proof is analogous to the proof of Theorem 2 for simplicial sets, but is not immediately implied by it.

[21] We used the orientation change only to define m in a more readable way.

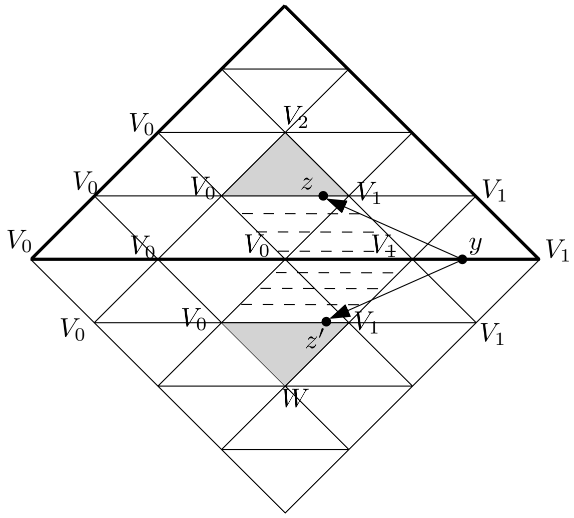


Figure 10: The homotopy H_τ takes y linearly into z and $H_{\tau'}$ takes y into z'. Due to the symmetry represented by the horizontal line, $|pf'|$ maps $H_\tau(y,t)$ into the same point of X as $|pf'|H_{\tau'}(y,t)$.

Lemma 10.1. *Let $d \geq 2$. There exists a sequence $\{X_k\}_{k\geq 1}$ of $(d-2)$-reduced $(d-1)$-connected simplicial sets, such that $H_d(X_k) \simeq \mathbb{Z}$ for all k and for any choice of cycles $z_k \in Z_d(X_k)$ generating the homology, the largest coefficient in z_k grows exponentially*[22] *in* size(X_k).

Proof of Theorem 2 based on Lemma 10.1. Let $\{X_k\}_{k\geq 1}$ be the sequence of simplicial sets from Lemma 10.1. Since they are $(d-1)$-connected, by the theorem of Hurewicz, $\pi_d(X_k) \simeq H_d(X_k) \simeq \mathbb{Z}$. For each k, let Σ_k be a simplicial sets with $|\Sigma_k| = S^d$, and $f_k : \Sigma_k \to X_k$ a simplicial map representing a generator of $\pi_d(X_k)$. Let $\xi = ...$ The generator of $H_d(\Sigma_d)$ contains each non-degenerate d-simplex with a coefficient ± 1. The Hurewicz isomorphism $\pi_d(X_k) \to H_d(X_k)$ maps such a representative to the formal sum of simplices

$$f_k \mapsto \sum_{\sigma \text{ is a } d-\text{simplex in } (\Sigma_k)} \pm f_k(\sigma) \in C_d(X_k)\,,$$

which represents a generator of $H_d(X_k)$ It follows from Lemma 10.1 that the number of d-simplices in Σ_k grows exponentially in size(X_k). Moreover, the complexity of any algorithm that computes $f_k : \Sigma_k \to X_k$ is at least the size of Σ_k, which completes the proof. □

It remains to define the sequence from Lemma 10.1:

Proof of Lemma 10.1. For every $k \geq 1$ we define the simplicial sets X_k to have one vertex $*$, no non-degenerate simplices up to dimension $d-2$, k non-degenerate $(d-1)$-simplices $\sigma_1, \ldots, \sigma_k$ that are all spherical (that is, for all i, j, $d_i\sigma_j = *$ is the degeneracy of the only vertex of X_k), and $k+1$ d-simplices $A, B_1, B_2, \ldots, B_{k-1}, C$ such that

- $d_0A = \sigma_1$, $d_jA = *$ for $j > 0$,
- $d_0B_i = \sigma_i$, $d_1B_i = \sigma_{i+1}$, $d_2B_i = \sigma_i$ and $d_jB_i = *$ for $j > 2$, and
- $d_0C = \sigma_k$, $d_jC = *$ for $j > 0$.

[22]With a slight abuse of language, we assume that each X_k not only a simplicial set but also its algorithmic representation with a specified size such as explained in Section 5.

X_k does not have any non-degenerate simplices of dimension larger than d. The relations of a simplicial set are satisfied, because $d_i d_j$ is trivial in all cases.

The boundary operator in the associated normalised chain complex $C_*(X_i)$ acts on basis elements as

- $\partial A = \sigma_1$
- $\partial B_i = 2\sigma_i - \sigma_{i+1}$, and
- $\partial C = \sigma_k$.

To see that X_k is $(d-1)$-connected, it is enough to prove that $H_{d-1}(X_k)$ is trivial (by Hurewicz theorem). This is true, because σ_1 is the boundary of A and for $i > 1$, σ_i is the boundary of the chain $2^{i-1}A - 2^{i-2}B_1 - \ldots - 2B_{i-2} - B_{i-1}$.

There are no non-degenerate $(d+1)$-simplices, so $H_d(X_k) \simeq Z_d(X_k)$ and a simple computation shows that every cycle is a multiple of

$$2^{k-1}A - 2^{k-2}B_1 - 2^{k-3}B_2 - \ldots - B_{k-1} - C.$$

An elementary representation of X_k has size that grows linearly with k. We see that the coefficients of homology generators grow exponentially with k, so they grow exponentially with size(X_k).

Acknowledgements. We would like to thank Marek Krčál, Lukáš Vokřínek and Sergey Avvakumov for helpful conversations and comments.

References

[1] S. I. Adyan. Algorithmic unsolvability of problems of recognition of certain properties of groups. *Dokl. Akad. Nauk SSSR (N.S.)*, 103:533–535, 1955.

[2] D. J. Anick. The computation of rational homotopy groups is #℘-hard. Computers in geometry and topology, Proc. Conf., Chicago/Ill. 1986, Lect. Notes Pure Appl. Math. 114, 1–56, 1989.

[3] C. Berger. *An effective version of the Hurewicz theorem.* Theses, Université Joseph-Fourier - Grenoble I, 1991. URL: `https://tel.archives-ouvertes.fr/tel-00339314`.

[4] Clemens Berger. Un groupoïde simplicial comme modèle de l'espace des chemins. *Bulletin de la Societé mathematique de France*, 123(1):1–32, 1995.

[5] E. H. Brown (jr.). Finite computability of Postnikov complexes. *Ann. Math. (2)*, 65:1–20, 1957.

[6] M. Cadek, M. Krcál, J. Matoušek, Lukás Vokřínek, and Uli Wagner. Extending continuous maps: polynomiality and undecidability. In *STOC*, pages 595–604, 2013.

[7] M. Čadek, M. Krčál, J. Matoušek, F. Sergeraert, L. Vokřínek, and U. Wagner. Computing all maps into a sphere. *J. ACM*, 61(3):17:1–17:44, June 2014.

[8] M. Čadek, M. Krčál, J. Matoušek, L. Vokřínek, and U. Wagner. Extendability of continuous maps is undecidable. *Discr. Comput. Geom.*, 51(1):24–66, 2013.

[9] M. Čadek, M. Krčál, J. Matoušek, L. Vokřínek, and U. Wagner. Polynomial-time computation of homotopy groups and Postnikov systems in fixed dimension. *Siam Journal on Computing*, 43(5):1728–1780, 2014.

[10] Martin Čadek, Marek Krčál, and Lukáš Vokřínek. Algorithmic solvability of the lifting-extension problem. *Discrete & Computational Geometry*, pages 1–51, 2017.

[11] H. Edelsbrunner and J. L. Harer. *Computational topology*. American Mathematical Society, Providence, RI, 2010.

[12] Herbert Edelsbrunner and Daniel R. Grayson. Edgewise subdivision of a simplex. In *Proceedings of the Fifteenth Annual Symposium on Computational Geometry*, SCG '99, pages 24–30, New York, NY, USA, 1999. ACM.

[13] Steve Ferry and Shmuel Weinberger. Quantitative algebraic topology and lipschitz homotopy. *Proceedings of the National Academy of Sciences*, 110(48):19246–19250, 2013.

[14] M. Filakovský. Effective chain complexes for twisted products. Preprint, 2012. URL: `arXiv:1209.1240`.

[15] M. Filakovský and L. Vokřínek. Are two given maps homotopic? An algorithmic viewpoint, 2013. Preprint. URL: `arXiv:1312.2337`.

[16] P. Franek and M. Krčál. Robust satisfiability of systems of equations. *J. ACM*, 62(4):26:1–26:19, 2015.

[17] Michael Freedman and Vyacheslav Krushkal. Geometric complexity of embeddings in $\mathbb{R}^d$. *Geometric and Functional Analysis*, 24(5):1406–1430, 2014.

[18] P. G. Goerss and J. F. Jardine. *Simplicial homotopy theory*. Birkhäuser, Basel, 1999.

[19] M. Gromov. Quantitative homotopy theory. *Prospects in Mathematics: Invited Talks on the Occasion of the 250th Anniversary of Princeton University (H. Rossi, ed.)*, pages 45–49, 1999.

[20] A. Haefliger. Plongements différentiables dans le domaine stable. *Comment. Math. Helv.*, 37:155–176, 1962/1963.

[21] A. Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, 2001.

[22] J. Heras, V. Pascual, J. Rubio, and F. Sergeraert. fKenzo: a user interface for computations in algebraic topology. *J. Symb. Comput.*, 46(6):685–698, 2011.

[23] D. Kan. The Hurewicz Theorem. In *International Symposium of Algebraic Topology, Autonomous University of Mexico and UNESCO*, pages 225–231, 1958.

[24] D. M. Kan. A combinatorial definition of homotopy groups. *Annals of Mathematics*, 67(2):282–312, 1958.

[25] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Computing*, 8:499–507, 1981.

[26] S. O. Kochman. *Stable homotopy groups of spheres. A computer-assisted approach.* Lecture Notes in Mathematics 1423. Springer-Verlag, Berlin etc., 1990.

[27] M. Krčál, J. Matoušek, and F. Sergeraert. Polynomial-time homology for simplicial Eilenberg–MacLane spaces. *Foundat. of Comput. Mathematics*, 13:935–963, 2013.

[28] Isaac Mabillard and Uli Wagner. Eliminating higher-multiplicity intersections, II. The deleted product criterion in the r-metastable range. In *Proc. 32nd International Symposium on Computational Geometry (SoCG 2016)*, pages 51:1–51:12, 2016.

[29] J. Matoušek, M. Tancer, and U. Wagner. Hardness of embedding simplicial complexes in $\mathbb{R}^d$. *J. Eur. Math. Soc.*, 13(2):259–295, 2011.

[30] J. Matoušek. Computing higher homotopy groups is $W[1]$-hard. *Fundamenta Informaticae*, 2014.

[31] Jiří Matoušek, Eric Sedgwick, Martin Tancer, and Uli Wagner. Embeddability in the 3-sphere is decidable. In *Proceedings of the Thirtieth Annual ACM Symposium on Computational Geometry*, SOCG'14, pages 78–84, New York, NY, USA, 2014.

[32] S. Matveev. *Algorithmic Topology and Classification of 3-Manifolds.* Springer, 2007.

[33] J. P. May. *Simplicial Objects in Algebraic Topology.* Chicago Lectures in Mathematics. University of Chicago Press, Chicago, IL, 1992. Reprint of the 1967 original.

[34] J. R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, Reading, MA, 1984.

[35] D.G. Quillen. *Homotopical Algebra.* Lecture Notes in Mathematics. Springer Berlin Heidelberg, 1967.

[36] M. O. Rabin. Recursive unsolvability of group theoretic problems. *Ann. of Math. (2)*, 67:172–194, 1958.

[37] D. C. Ravenel. *Complex Cobordism and Stable Homotopy Groups of Spheres (2nd ed.).* Amer. Math. Soc., 2004.

[38] P. Real. An algorithm computing homotopy groups. *Mathematics and Computers in Simulation*, 42:461—465, 1996.

[39] A. Romero, J. Rubio, and F. Sergeraert. Computing spectral sequences. *J. Symb. Comput.*, 41(10):1059–1079, 2006.

[40] A. Romero and F. Sergeraert. Effective homotopy of fibrations. *Applicable Algebra in Engineering, Communication and Computing*, 23(1-2):85–100, 2012.

[41] A. Romero and F. Sergeraert. A Bousfield–Kan algorithm for computing the effective homotopy of a space. *Foundations of Computational Mathematics*, pages 1–32, 2016.

[42] J. Rubio and F. Sergeraert. Constructive algebraic topology. *Bull. Sci. Math.*, 126(5):389–412, 2002.

[43] J. Rubio and F. Sergeraert. Algebraic models for homotopy types. *Homology, Homotopy and Applications*, 17:139–160, 2005.

[44] J. Rubio and F. Sergeraert. Constructive homological algebra and applications. Preprint, `arXiv:1208.3816`, 2012.

[45] R. Schön. *Effective Algebraic Topology.* Memoirs of the American Mathematical Society. American Mathematical Society, 1991.

[46] F. Sergeraert. The computability problem in algebraic topology. *Adv. Math.*, 104(1):1–29, 1994.

[47] J. R. Smith. m-Structures determine integral homotopy type. Preprint, `arXiv:math/9809151v1`, 1998.

[48] R. I. Soare. Computability theory and differential geometry. *Bull. Symbolic Logic*, 10(4):457–486, 2004.

[49] L. Vokřínek. Decidability of the extension problem for maps into odd-dimensional spheres. *Discrete & Computational Geometry*, 57(1):1–11, 2017.

[50] C. Weber. Plongements de polyèdres dans le domaine metastable. *Comment. Math. Helv.*, 42:1–27, 1967.

[51] A. J. Zomorodian. *Topology for Computing*, volume 16 of *Cambridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, Cambridge, 2005.