Torsion in $\Gamma(\pi_2 K)/\pi_1 K$

Benjamin Ruppik

Geboren am 30. Juni 1994 in Aachen

8. Juni 2016

Bachelorarbeit Mathematik

Betreuer: Dr. Daniel Kasprowski

Zweitgutachter: Prof. Dr. Peter Teichner

Max Planck Institute for Mathematics in Bonn

Mathematisch-Naturwissenschaftliche Fakultät der

Rheinischen Friedrich-Wilhelms-Universität Bonn

Contents

List of Figures

List of Tables

Abstract

In this thesis the invariant $\mathrm{Tors}(\Gamma(\pi_2(K))/\pi_1(K))$ for a finite, 2-dimensional CW-complex K with finite $\pi_1(K)$ is studied. This includes the development of a *Sage*-module to automate the computation.

Since the invariant only depends on the homotopy type of K we can assume K to have only one 0-cell, and that the 2-cells are attached along words to a wedge of circles. This leads to a close connection between such 2-complexes and group presentations.

The term $\mathrm{Tors}(\Gamma(\pi_2(K))/\pi_1(K))$ is important for a specific case distinction in the stable classification of 4-manifolds. The quotient $\Gamma(\pi_2(K))/\pi_1(K)$ allows inference of the existence of homotopies between maps into Eilenberg-MacLane-spaces and maps into lower dimensional skeleta. Understanding the torsion is enough as this is the kernel of a homomorphism from $\Gamma(\pi_2(K))/\pi_1(K)$ to the intersection forms of the manifold.

The explicit calculation in the *Sage*-module is done in steps: We use a homological description of the second homotopy group, and calculate the homology of the universal cover of K with the help of Fox-derivatives. The gamma functor, introduced by Whitehead, is universal with respect to quadratic maps. In the case of a free $\mathbb{Z}$-module (here $\pi_2 K$), we can look at this as a subgroup of a tensor product. There is a diagonal action of the fundamental group on $\Gamma(\pi_n)$ originating from the usual action of π_1 on π_n. The quotient is a finitely generated abelian group, and as such a direct sum of a free and a torsion part. Here we ask whether this torsion part is trivial.

After a short motivation regarding the classification of manifolds this thesis develops most of the theoretical background needed in the computation. Then we will go in detail on the specific implementation in *Sage*. At the end we list concrete results for some example group presentations.

Zusammenfassung

In dieser Bachelorarbeit wird die Invariante $\mathrm{Tors}(\Gamma(\pi_2(K))/\pi_1(K))$ für einen endlichen, 2-dimensionalen CW-Komplex K mit endlicher Fundamentalgruppe genauer untersucht. Im Rahmen davon ist ein *Sage*-Modul entstanden, welches die Berechnung automatisiert.

Da diese Invariante nur vom Homotopietyp von K abhängt, können wir ohne Beschränkung annehmen, dass K nur eine 0-Zelle besitzt und die 2-Zellen entlang von Wörtern an das 1-Skelett angeklebt wurden. Dies führt zu einer Korrespondenz zwischen solchen 2-Komplexen und Präsentationen von Gruppen.

Der Term $\mathrm{Tors}(\Gamma(\pi_2(K))/\pi_1(K))$ tritt in der stabilen Klassifikation von 4-Mannigfaltigkeiten auf. Dabei gibt der Quotient $\Gamma(\pi_2(K))/\pi_1(K)$ Aufschluss darüber, ob bestimmte Abbildungen in einen Eilenberg-MacLane-Raum homotop zu Abbildungen in ein niedrigdimensionaleres Skelett sind. Die Torsion ist interessant, da sie den Kern einer Abbildung von $\Gamma(\pi_2(K))/\pi_1(K)$ in die Schnittformen der Mannigfaltigkeit beschreibt.

Die explizite Berechnung im *Sage*-Programm erfolgt in mehreren Schritten: Zur Beschreibung der zweiten Homotopiegruppe benutzen wir als Hilfsmittel die Homologie der universellen Überlagerung von K. Die Randabbildung kann dort mit sogenannten Fox-Derivationen berechnet werden. Der Gamma-Funktor, von Whitehead eingeführt, ist eine universelle Konstruktion bezüglich quadratischer Abbildungen. Auf einem freien $\mathbb{Z}$-Modul, wie hier $\pi_2(K)$, gibt es eine komplett algebraische Beschreibung als Teilmenge eines Tensorprodukts. Die Wirkung der Fundamentalgruppe auf die höheren Homotopiegruppen kann diagonal auf $\Gamma(\pi_n)$ fortgesetzt werden. Der Quotient ist eine endlich erzeugte abelsche Gruppe, und daher zerfällt er in einen freien und ein einen Torsionsanteil. Wir wollen herausfinden, ob dieser Torsionsanteil trivial ist.

Nach einer kurzen Motivation über die Klassifikation von Mannigfaltigkeiten wird in der Arbeit zunächst der für die explizite Berechnung nötige theoretische Unterbau beschrieben. Danach wird detailliert auf die konkrete Implementierung in *Sage* eingegangen. Zum Abschluss werden einige Ergebnisse für kleine Gruppenpräsentationen (wie für zyklische Gruppen oder Diedergruppen) aufgelistet.

1 Introduction

1.1 Motivation

1.1.1 What are we interested in?

The classification of objects and especially the morphisms between them is ubiquitous in mathematics, no matter if someone is interested in groups, modules, schemes, topological spaces, ...

While dealing with this we ask the essential questions:

- What kind of objects and maps are there? Which "shapes" are possible, which are not? Can we give a comprehensive list?
- Given a specific object, of what type is it? Which properties does it have and where is it on this list?

Often these objects and the maps between them are not easy to understand per se, in particular showing that two things **cannot** be transformed into another is a big challenge. Fortunately to distinguish two objects it is enough to find an invariant that does not agree.

In algebraic topology we build algebraic images of topological spaces, and these algebraic images hopefully are easier to understand than the spaces where they come from. The functors we use for creating these images, which Hatcher aptly calls "lanterns" of algebraic topology [Hat02], also give images of continuous maps between the spaces.

In this text we focus on one very specific tool that is useful in the classification of topological spaces which locally look like 4-dimensional real space.

1.1.2 Towards $\mathrm{Tors}(\Gamma(\pi_2 K)/\pi_1 K)$

This section at first describes the objects we actually want to classify, and after this explains where the mysterious $\Gamma(\pi_2 K)/\pi_1 K$ term turns up. For the purpose of brevity in this motivation we will not state everything in detail, therefore not all proofs and definitions are given. We are especially negligent with the base points of the spaces. Nevertheless some of the claims will later be revisited in the text.

The main goal of this thesis was the actual computation of this invariant for specific K. One of the products is a computer program exactly for this purpose. A detailed description of the code is given in section 3.

The objects we want to classify are the closed (compact without boundary), connected, oriented 4-manifolds with a fixed fundamental group π, but only up to homeomorphism and connected sum with some $\#_k(\mathbb{S}^2 \times \mathbb{S}^2)$. In the literature this is called the *stable classification of 4-manifolds with fixed fundamental group.*

Example 1.1 (4-manifolds)**.** The first examples one can think of are simply connected spaces: The four-dimensional sphere $\mathbb{S}^4$, the product $\mathbb{S}^2 \times \mathbb{S}^2$, complex projective space $\mathbb{CP}^2$ and connected sums of the above.

But actually any finitely presented group π can be realized as the fundamental group of a closed 4-manifold: Take the surface complex associated to the presentation and embed it in $\mathbb{R}^5$. Thicken it a little, i.e. take a small ϵ-neighborhood such that the result deformation retracts to the original complex. Then the boundary is a closed 4-manifold with fundamental group π. For details on this construction consult [Sti12, Chapter 9.4.1]. $\triangle$

Let us adopt a convention on how the data is presented. Recall that for any group G there is a CW-complex BG with just one nontrivial homotopy group in degree one:

$$\pi_n(BG, *) = \begin{cases} G & n = 1 \\ 0 & \text{else} \end{cases}$$

BG is also called the *Eilenberg-MacLane space* $K(G, 1)$, and it is unique up to weak homotopy equivalence.

Furthermore for any group π there is a correspondence

$$\begin{aligned} [X; B\pi]_* &\cong_{(Set)} \operatorname{Hom}(\pi_1 X, \pi) \\ [f] &\mapsto \pi_1([f]) \end{aligned}$$

where $\pi_1([f])\colon \pi_1(X) \to \pi_1(B\pi) \cong \pi$. So to specify the fixed fundamental group we choose an isomorphism $\phi \in \operatorname{Hom}(\pi_1 X, \pi)$ and use its corresponding map $[f] \in [X; B\pi]_*$.

Now we are ready to construct our first invariant: Since the manifold M is connected, orientable and closed, the fourth homology group is isomorphic to an infinite cyclic group: $H_4(M; \mathbb{Z}) \cong \mathbb{Z}$. As M is already oriented we have chosen a generator $[M] \in \mathbb{Z}$ of this group. We look at the image of this fundamental class under the map induced by $[f] : M \to B\pi$ on fourth homology:

$$\begin{aligned} \left\{ \begin{matrix} \text{closed, connected, oriented 4-manifolds} \\ \text{with fixed fundamental group } \pi \end{matrix} \right\} \Big/ {\scriptstyle \text{homeomorphism} \atop -\#_k(\mathbb{S}^2 \times \mathbb{S}^2)} &\to H_4(B\pi; \mathbb{Z}) \\ [M^4 \xrightarrow{f} B\pi] &\mapsto f_*([M]) \end{aligned}$$

Here we write f_* for the induced map $H_4([f])$ on homology.

Now we will just take the following two facts for granted. They tell us whether the map f can be deformed into something easier, here into a map whose image is contained in the 2-skeleton of $B\pi$.

Fact 1.2. *If $f_*([M]) = 0$, then there exists an $f'\colon M^{(3)} \to B\pi^{(2)}$ with $f' \simeq f \restriction_{M^{(3)}}$.*
The homotopy possibly uses the whole of $B\pi$, not just $B\pi^{(2)}$.

Sometimes this map can be extended to the entirety of M:

Fact 1.3. *Consider the attaching map $\mathbb{S}^3 \xrightarrow{\alpha} M^{(3)}$ of the highest cell. If $[f' \circ \alpha] = 0 \in \pi_3(B\pi^{(2)})/\pi$, then there exists an $f'' \colon M \to B\pi^{(2)}$ with $f'' \simeq f$. Again the homotopy might use the whole of $B\pi$.*

Observe that fact 1.3 would be easy to prove under the stronger assumption $[f' \circ \alpha] = 0 \in \pi_3(B\pi^{(2)})$: Just use the null homotopy to extend the map over $\mathbb{D}^4$. It is surprising that we can still find this extension only by knowing the class of the map in the quotient by the fundamental group action!

Recall that homotopy and homology groups are deeply connected via the *Hurewicz homomorphism* h_n: For each $n \in \mathbb{N}$ we can choose a generator $u_n \in H_n(\mathbb{S}^n) \cong \mathbb{Z}$ to get a map

$$h_n \colon \pi_n(M) \to H_n(M) \tag{1}$$
$$[f] \mapsto H_n([f])(u_n) \tag{2}$$

which is a group homomorphism.

In his paper [Whi50] Whitehead shows that these morphisms, a priori not related for different n, fit into an exact sequence:

Define $\Gamma_n^X \coloneqq \mathrm{Im}(\pi_n(i) \colon \pi_n(X^{(n-1)}) \to \pi_n(X^{(n)}))$ for all $n \geq 2$. This gives a natural exact sequence:

$$\ldots \to \pi_{n+1}(X) \xrightarrow{h_{n+1}} H_{n+1}(X;\mathbb{Z}) \to \Gamma_n^X \to \pi_n(X) \xrightarrow{h_n} H_n(X;\mathbb{Z}) \to \ldots$$

There are easier algebraic descriptions for these groups Γ_n^X. The case we are especially interested in is $\Gamma_3^X = \Gamma(\pi_2(X))$, where $\Gamma(-)$ is *Whitehead's quadratic functor*. We will give a definition of Γ in section 2.3.

For the purpose of this motivation let us look at this certain sequence for the space $B\pi^{(2)}$ which turned up in the discussion above. As a further simplification we will actually look at its universal cover $\widetilde{B\pi}$, since the homotopy groups of the two agree.

$$\pi_4\widetilde{B\pi^{(2)}} \xrightarrow{h_4} \underbrace{H_4\widetilde{B\pi^{(2)}}}_{=0} \longrightarrow \Gamma(\pi_2\widetilde{B\pi^{(2)}}) \xrightarrow{\cong} \pi_3\widetilde{B\pi^{(2)}} \xrightarrow{h_3} \underbrace{H_3\widetilde{B\pi^{(2)}}}_{=0} \tag{3}$$

The two groups $H_4\widetilde{B\pi^{(2)}}$ and $H_3\widetilde{B\pi^{(2)}}$ are zero because there are neither 4- nor 3-cells in the cellular chain complex of $\widetilde{B\pi^{(2)}}$. Since the sequence is exact this implies that the map from $\Gamma(\pi_2)$ to π_3 is an isomorphism. Thus instead of $\pi_3 B\pi^{(2)}$ we can calculate $\Gamma(\pi_2 B\pi^{(2)})$. This is sometimes easier since it is just an algebraic construction on the homotopy group one degree lower.

There is one more reduction step possible if we assume π to be a finite group (as we will later always do): One can find a map from the Γ-construction into the intersection forms of the 4-manifold, whose kernel is exactly the torsion:

$$0 \to \mathrm{Tors}(\Gamma(\pi_2 \widetilde{B\pi^{(2)}})/\pi) \to \Gamma(\pi_2 \widetilde{B\pi^{(2)}})/\pi \to \mathrm{Seq}(H^2(B\pi^{(2)}; \mathbb{Z}[\pi]))$$
$$[f' \circ \alpha] \mapsto \lambda$$

where

$$\lambda \colon H^2(M; \mathbb{Z}[\pi]) \otimes H^2(M; \mathbb{Z}[\pi]) \to \mathbb{Z}[\pi]$$

is considered in $\mathrm{Seq}(H^2(B\pi^{(2)}; \mathbb{Z}[\pi])$ via the induced map on cohomology

$$\begin{aligned} H^2(B\pi^{(3)}; \mathbb{Z}[\pi]) &\to H^2(M^{(3)}; \mathbb{Z}[\pi]) \\ &\cong H^2(M; \mathbb{Z}[\pi]) \end{aligned}$$

Given the complex $B\pi^{(2)}$ our goal now is to understand and calculate this torsion.

1.2 Notational conventions

We will denote the category of groups as *(Grp)*, the category of abelian groups as *(Ab)*, the category of rings as *(Ring)*, ...

All rings contain a 1 and are commutative unless otherwise noted. For a commutative ring A the category of modules over A will be called *(A-Mod)*.

Remember that an abelian group G has the unique structure of a $\mathbb{Z}$-module by defining

$$\begin{aligned} n \cdot g &= \underbrace{g + \cdots + g}_{n \text{ times}} && \text{for } g \in G, n \geq 0 \\ n \cdot g &= -((-n) \cdot g) && \text{for } g \in G, n < 0. \end{aligned}$$

As every module has an underlying abelian group we will use the words *abelian group* and *$\mathbb{Z}$-module* interchangeably. Formally this is an equivalence of categories between *(Ab)* and *($\mathbb{Z}$-Mod)*.

We fix some standard notation:

- $\mathbb{S}^n := \{x \in \mathbb{R}^{n+1} \mid \|x\| = 1\}$, the n-dimensional unit sphere
- $\mathbb{D}^n := \{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$, the closed n-disk
- $\mathfrak{S}_n$, the symmetric group with $n!$ elements

We will use $\cong$ to denote isomorphisms in the appropriate category, but also have the special symbols $\simeq$ for homotopy equivalence and $\approx$ for homeomorphism.

The basic category we work in is the category of topological spaces *(Top)*: The objects are topological spaces and the morphisms continuous maps $f\colon X \to Y$.

Often we are only interested in the homotopy type of a map: If X and Y are topological spaces let $[X;Y]$ be the equivalence classes of all continuous functions $X \to Y$ modulo the relation of homotopy. The equivalence class of f will be denoted by $[f]$, composition is defined as $[f] \circ [g] := [f \circ g]$. This yields the category *(hTop)* where the objects are topological spaces and the morphisms between the spaces X and Y are given by $[X;Y]$.

Sometimes we need to keep track of a basepoint: If (X, x_0) and (Y, y_0) are pointed spaces, we look at pointed maps, i.e. continuous maps $f : (X, x_0) \to (Y, y_0)$ with $f(x_0) = y_0$. Then the pointed homotopy category $(hTop_*)$ is defined to be the category whose objects are pointed topological spaces and whose morphisms are equivalence classes of pointed maps modulo pointed homotopy (the homotopy has to fix the basepoint as well). We will use a star to denote the set of morphisms in $(hTop_*)$ as $[X;Y]_*$.

Let R be a commutative ring and G be a group. Define the *group ring* $R[G]$ as the free module with basis the elements of G:

$$\begin{aligned} R[G] &:= \bigoplus_{g\in G} Rg \\ &= \{\sum_{g\in G} r_g g \text{ finite sum } | r_g \in R\} \end{aligned}$$

There is a natural multiplication in the group ring: For two basis vectors e_g and e_h define their product as $e_g \cdot e_h := e_{gh}$ and extend linearly. One has to be careful: This multiplication is commutative if and only if the group G is abelian. The construction of the group ring is functorial, i.e. every group homomorphism $G \xrightarrow{f} H$ extends to a ring homomorphism

$$R[G] \to R[H], \sum r_g g \mapsto \sum r_g f(g).$$

$\square$ will mark the end of a proof, $\triangle$ the end of an example.

2 Theoretical background

2.1 Preparations

Let K be a finite, connected, 2-dimensional CW-complex, meaning that it has finitely many cells just in dimension less than or equal to two. Think of K as being built up in steps: Start with a finite discrete set of points (the 0-cells), then attach some unit intervals to get a graph, and after this glue in 2-disks along edges of the graph. The Seifert-van-Kampen theorem allows us to calculate the fundamental group of such a complex [Hat02, Prop. 2.16]:

The 1-skeleton is homotopic to a wedge of circles, each corresponding to a generator of the group. The 2-cells just add the words along which they are attached as relations. In the following discussion we always assume $\pi_1(K, *)$ to be a finite group.

The motivation has given enough justification why studying

$$\mathrm{Tors}(\Gamma(\pi_2 K)/\pi_1 K)$$

is worth our time; we will now develop the technical details needed. Emphasis will be put on how all the calculations could be implemented in a computer program, for further discussion see section 3.

Writing the expression as the composition

$$(\mathrm{Tors}(-) \circ -/\pi_1 K \circ \Gamma(-) \circ \pi_2 -)(K) \tag{4}$$

we get a roadmap for the individual parts.

Two preliminary observations to start off: A CW-complex is locally contractible, thus locally path-connected, and as such connected if and only if it is path-connected. As a consequence different choices of the base point give isomorphic fundamental groups. We will often drop the base point from the notation (as it has already been done).

$\pi_n(-)$ only depends on the homotopy type of a space, thus we are only interested in K up to homotopy equivalence. Combine this with the following proposition:

Proposition 2.1 ([Fen83, Corollary 1.1.3]). *Any finite connected CW-complex has the homotopy type of a CW-complex with just one 0-cell.*

Proof. Let X be such a CW-complex, if it already has just one 0-cell we are done. Otherwise suppose σ^0 and ρ^0 are two distinct 0-cells of X, joined by a 1-cell σ^1 (here we use that X is path connected). The subcomplex $A := \sigma^0 \cup \sigma^1 \cup \rho^0$ is contractible, and as the pair (X, A) satisfies the homotopy extension property there is a homotopy equivalence $X \simeq X/A$. Now X/A has the structure of a CW-complex with one less 0-cell, and we can continue inductively. $\square$

2.1.1 2-complexes $\leftrightarrow$ group presentations

In the case of a 2-complex K with just one 0-cell there is a correspondence which will be crucial for representing K in a format suitable as input for a computer program.

The following lemma tells us that we can change attaching maps by a homotopy:

Lemma 2.2 (The cell homotopy lemma, [Fen83, Lemma 1.1.2]). *Let X be a CW-complex and let the attaching map $\phi : \mathbb{S}^{i-1} \to X^{(i)}$ of an i-cell be changed by a homotopy. Then K is deformed into another complex which has the same homotopy type.*

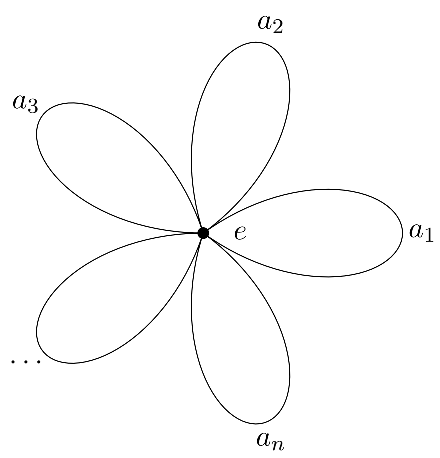

Figure 1: The 1-skeleton of a cell complex with just one 0-cell

Let e be the 0-cell of K to which the 1-cells $a_1, a_2, \dots$ are attached. The resulting 1-skeleton is homeomorphic to a wedge of n circles (see figure 1),

$$K^{(1)} \approx \bigvee_{i=1}^{n} \mathbb{S}^1.$$

Let $r_1, r_2, \dots$ be the 2-cells (r is the abbreviation for *relation*), and $\phi : \mathbb{S}^1 \to K^{(1)}$ one of the attaching maps. Then by the cell homotopy lemma 2.2 we can change ϕ by a homotopy without altering the homotopy type of K. The circle $\mathbb{S}^1$ is compact, so is its image $\phi(\mathbb{S}^1) \subseteq K^{(1)}$ and thus meets only finitely many of the 1-cells. If we redistribute the time spent in each of the loops via a homotopy we can assume that the inverse image $\phi^{-1}(\{e\})$ consists of m-th roots of unity, i.e.

$$\phi^{-1}(\{e\}) = \{1, \omega, \omega^2, \dots, \omega^m\}, \omega = e^{\frac{2\pi i}{m}}.$$

Denote by (ω^i, ω^{i+1}) the open arc between ω^i and ω^{i+1} on $\mathbb{S}^1$. By another homotopy we can force $f \restriction (\omega^i, \omega^{i+1})$ to be bijective.

The 1-cell traversed by $f(\omega^i, \omega^{i+1})$ will be called $a_{\alpha(i)}$. We then associate the word (also called r)

$$r = a_{\alpha(1)}^{\epsilon(1)} a_{\alpha(2)}^{\epsilon(2)} \dots a_{\alpha(m)}^{\epsilon(m)}$$

to the 2-cell r. Here $\epsilon(i)$ is $+1$ for a anticlockwise and -1 for a clockwise traversal. Up to cyclic reordering of their letters these words r determine K uniquely up to homotopy equivalence.

Definition 2.3 (Group presentations)**.** A *group presentation* is a pair of sets, written as $\langle A \mid R\rangle$, such that to each $r \in R$ there is associated a word

$$w(r) = x_1 x_2 \dots x_k, x_i \in A \cup A^{-1}$$

Here A^{-1} is the collection of formal inverses of the elements in A. Even though the relators r and the relations $w(r)$ are technically distinct we will often confuse them.

The previous discussion gives the correspondence

$$\{\text{connected 2-complex with one 0-cell}\} \longleftrightarrow \{\text{group presentations}\} \tag{5}$$

and in the future we will talk about the two things as if they were the same. As mentioned before the Seifert-van-Kampen-theorem tells us the fundamental group of such a complex has exactly the presentation associated to it in equation 5.

Example 2.4 (Group presentations)**.** To illustrate this connection take a look at some examples:

- $\langle a \mid r, s\rangle$, where $w(r) = w(s) = a$ corresponds to a 1-sphere $\mathbb{S}^1$ with two 2-cells glued in by degree 1 maps. This is just a CW-decomposition of the 2-sphere $\mathbb{S}^2$, the presentation is that of the trivial group.
- $\langle \emptyset \mid 1\rangle$ is also the 2-sphere $\mathbb{S}^2$, because here we are collapsing the boundary of a 2-cell to a point.
- $\langle a \mid 1\rangle$ is $\mathbb{S}^1 \vee \mathbb{S}^2$.
- $\langle a, b \mid [a, b]\rangle$, where the commutator of two elements is defined as $[a, b] := aba^{-1}b^{-1}$. This presentation corresponds to the 2-torus $\mathbb{S}^1 \times \mathbb{S}^1$. The group is the torus group $\mathbb{Z} \times \mathbb{Z}$
- $\langle a_1, \dots, a_g, b_1, \dots b_g \mid [a_1, b_1] \cdot \dots \cdot [a_g, b_g]\rangle =: M_g$ is the closed orientable surface of genus g, i.e. a connected sum of g tori.
  $M_g \approx \#_g(\mathbb{S}^1 \times \mathbb{S}^1)$
- $\langle a_1, \dots, a_g \mid a_1^2 a_2^2 \dots a_g^2\rangle =: N_g$ is the closed non-orientable surface of genus g, a connected sum of g projective spaces.
  $N_g \approx \#_g\mathbb{RP}^2$

△

The importance of the correspondence 5 lies in the fact that we now have a combinatorial description of the complexes K we are interested in: By 2.1 we can limit our attention to complexes with just one 0-cell and those can be completely described by a group presentation — something we can use as the input for a computer program!

Remark 2.5 (Convention). For the rest of this text K will always denote a finite, connected, 2-dimensional CW-complex with finite fundamental group $\pi_1(K,*)$. Without loss of generality we assume K to have just one 0-cell e.

We will use letters such as $X, Y, \ldots$ when talking about general topological spaces or CW-complexes.

2.2 A higher homotopy group: $\pi_2(-)$

2.2.1 Homotopy groups and their connection to homology

Definition 2.6 (Homotopy groups π_n)**.** The *absolute homotopy groups* of a topological space X with basepoint x_0 are given by the set

$$\pi_n(X, x_0) \coloneqq [(\mathbb{S}^n, *); (X, x_0)]_*$$

where $* = (1, 0, \ldots, 0)$ (this choice is arbitrary).

Define the group structure as follows: There is a collapse map $p : \mathbb{S}^n \to \mathbb{S}^n \vee \mathbb{S}^n$ by identifying the equator $\{x \in \mathbb{S}^n \mid x_{n+1} = 0\}$ to a point. For $[f], [g] \in \pi_n(X, x_0)$ let the product be $[f] \cdot [g] = [h]$ where h is the composition $\mathbb{S}^n \xrightarrow{p} \mathbb{S}^n \vee \mathbb{S}^n \xrightarrow{f \vee g} X$.

For more details and basic properties see for example [Hat02, 4.1].

Our challenge is to actually compute π_2 for a 2-dimensional CW-complex. Unfortunately the calculation of the higher homotopy groups ($n > 1$) turns out to be difficult, because there is no longer a Seifert-van-Kampen-style theorem.

In our case we can use the tools of homology to calculate the second homotopy group. At first recall some facts from covering space theory: Every CW-complex X is *locally contractible*, meaning that every point $x \in X$ has a local base of contractible neighborhoods ([Hat02, Proposition A.4]). In particular a locally contractible space is *locally simply connected*, i.e. every point $x \in X$ has a local base of simply connected neighborhoods. This in turn implies that X is *semi-locally simply-connected*, i.e. every $x \in X$ has a neighborhood U such that the map $\pi_1(U, x) \to \pi_1(X, x)$ induced by the inclusion $U \subset X$ is trivial.

According to [Hat02, Chapter 1.3], path-connected, locally path-connected and semi-locally simply-connected is sufficient for X to have an universal covering space $\widetilde{X}$.

The following proposition claims we can use any cover of X to calculate the higher homotopy groups:

Proposition 2.7 ([Hat02, Proposition 4.1])**.** *Let X be a connected topological space. A covering space projection $p\colon (\widetilde{X}, \widetilde{x_0}) \to (X, x_0)$ induces isomorphisms $\pi_n(p)\colon \pi_n(\widetilde{X}, \widetilde{x_0}) \to \pi_n(X, x_0)$ for all $n \geq 2$.*

Sketch. The map $\pi_n(p)$ is injective for all n because we can lift homotopies in X to $\widetilde{X}$. Surjectivity follows because every map $(\mathbb{S}^n, *) \to (X, x_0)$ gives a map $(\mathbb{S}^n, *) \to (\widetilde{X}, \widetilde{x_0})$ since the sphere $\mathbb{S}^n$ is simply-connected for $n \geq 2$. $\square$

Homotopy and homology groups are connected via the Hurewicz homomorphism we already met in the motivation (see equation 1).

Theorem 2.8 (Hurewicz, [Hat02, Theorem 4.32])**.** *If a space X is $(n-1)$-connected, $n \geq 2$, then $\widetilde{H}_i(X) = 0$ for $i < n$ and the Hurewicz homomorphism $h_n \colon \pi_n(X, x_0) \to H_n(X)$ gives an isomorphism $\pi_n(X, x_0) \cong H_n(X)$.*

By induction over n one can see that for a space X with $\pi_1(X) = \{1\}$ the first nonzero homotopy and homology groups appear in the same dimension and are isomorphic via the Hurewicz homomorphism.

The combination of 2.7 and 2.8 now allows us to compute $\pi_2(K)$: K has a universal cover $\widetilde{K}$ with $\pi_2(K) \cong \pi_2(\widetilde{K})$, and the Hurewicz theorem tells us that $\pi_2(\widetilde{K}) \cong H_2(\widetilde{K})$. If we now are able to understand $\widetilde{K}$ well enough to compute its cellular homology, we get a description of the second homotopy group. This is our goal for the next subsection.

2.2.2 Understanding $\widetilde{K}$ and its homology

Here we closely follow [Fen83, 4.1].

Since the domain of the characteristic maps of the cells in K are simply connected, we can lift those maps:

$$\begin{tikzcd} & \widetilde{K} \arrow[d, "p"] \\ \mathbb{D}^n \arrow[ur, dashed, "\exists"] \arrow[r, "\Phi"] & K \end{tikzcd} \tag{6}$$

Recall the fact that this actually gives a cell decomposition of $\widetilde{K}$. The unique 0-cell of K will be called e and acts as the basepoint in K. Choose once and for all a lift $\widetilde{e} \in \widetilde{K}$ of the cell e, this will be the basepoint we take in $\widetilde{K}$.

Every characteristic map $\Phi : \mathbb{D}^p \to K$ of a p-cell σ in K with $\Phi(1, 0, \ldots, 0) = e$ has a unique lift $\widetilde{\Phi} : \mathbb{D}^p \to \widetilde{K}$ with $\widetilde{\Phi}(1, 0, \ldots, 0) = \widetilde{e}$; this lift is the characteristic map of the p-cell $\widetilde{\sigma}$ in $\widetilde{K}$.

The action of the fundamental group $\pi_1(K, e)$ on the universal cover permutes these cells, and every other lift of σ is of the form $g\widetilde{\sigma}$ for a $g \in \pi_1(K, e)$. In particular there are $\#\pi_1(K, e)$ distinct lifts of each cell.

From now on we will write $\pi \coloneqq \pi_1(K, e)$.

Remember that we can compute the cellular homology of a CW-complex X as the homology of the cellular chain complex.

Definition 2.9 (Cellular chain complex)**.** The *cellular chain complex* $(C_\bullet^{\mathrm{CW}}, d_\bullet^{\mathrm{CW}})$ of a CW-complex X is defined as

$$C_n(X) := \bigoplus_{\{\text{n-cells of X}\}} \mathbb{Z} \tag{7}$$

$$d_n \colon C_n(X) \to C_{n-1}(X) \text{ defined on a basis element } \sigma^n \text{ as}$$

$$\sigma^n \mapsto \sum_{\substack{\sigma^{n-1} \\ \text{(n-1)-cell of } X}} [\sigma^{n-1}, \sigma^n]\sigma^{n-1} \tag{8}$$

Here $[\sigma^i, \sigma^{i+1}] \in \mathbb{Z}$ is the incidence number of σ^i with σ^{i+1}, defined as the degree of a map of spheres:

Definition 2.10 (Incidence number)**.** $[\sigma^i, \sigma^{i+1}]$ is the degree of the composition

$$\mathbb{S}^i \xrightarrow{\phi_{\sigma^{i+1}}} X^{(i)} \xrightarrow{q} X^{(i)} / \left(X^{(i)} \setminus \mathring{\sigma}^i \right) \approx \mathbb{S}^i \tag{9}$$

where $\phi_{\sigma^{i+1}}$ is the attaching map of the $(i+1)$-cell σ^{i+1} and q is the quotient map that collapses everything except the interior of σ^i to a point. The quotient is identified with a sphere via the characteristic map Φ_{σ^i} of the i-cell σ^i.

Example 2.11. Let $K = \langle A \mid R \rangle$ be a 2-dimensional CW-complex with one 0-cell e.

- $[e, a] = 0$ for all $a \in A$, because the endpoints of a are both on the unique 0-cell e.
- For a 1-cell a and a 2-cell r define

  $$\epsilon_a(r) := \text{ total exponent of } a \text{ in the word } w(r).$$

  For example $\epsilon_a(aba^{-1}b^{-2}) = 0$ and $\epsilon_b(aba^{-1}b^{-2}) = -1$.

  Then $[a, r] = \epsilon_a(r)$ because the total exponent is a signed count of the local degrees of the attaching map of the relation r.

$\triangle$

Let us collect all the lifts in $\widetilde{K}$ belonging to $\widetilde{\sigma}$ together:

$$\begin{aligned}
C_p(\widetilde{K}) &= \bigoplus_{\substack{\tau \\ \text{n-cell of } \widetilde{K}}} \mathbb{Z}_\tau \\
&\cong \bigoplus_{\substack{\sigma \\ \text{n-cell of } K}} \left(\bigoplus_{\substack{\sigma' \\ \text{lift of } \sigma}} \mathbb{Z}_{\sigma'} \right) \\
&\cong \bigoplus_{\substack{\sigma \\ \text{n-cell of } K}} \left(\bigoplus_{g\in\pi} \mathbb{Z}_{g\widetilde{\sigma}} \right) \\
&\cong \bigoplus_{\substack{\sigma \\ \text{n-cell of } K}} \mathbb{Z}[\pi]_{\widetilde{\sigma}}
\end{aligned}$$

Everything above is a $\mathbb{Z}$-module isomorphism. Thus we can write the chains in $C_p(\widetilde{K})$ in the form $\sum \lambda_i \widetilde{\sigma}_i$, where σ_i varies over the p-cells of K and the coefficients λ_i are in the group ring $\mathbb{Z}[\pi]$.

On the level of the chain complex we can also denote this as

$$C_p(\widetilde{K}) \cong \mathbb{Z}[\pi] \otimes_{\mathbb{Z}} C_p(K). \tag{10}$$

The more difficult part is the description of the new boundary maps $\widetilde{d_p} : C_p(\widetilde{K}) \to C_{p-1}(\widetilde{K})$. Here we have to calculate the degrees of the lifts of the attaching maps. We will only do this for $p \leq 2$ as in our explicit calculation only the second boundary map is involved.

By $\mathbb{Z}[\pi]$-linearity of the boundary maps we only need to know the values on the lifted cells $\widetilde{\sigma}$.

Let us use the following conventions for the fundamental group: The Latin letters $a, b, c, \ldots \in A$ will denote both the 1-cells of K and the generators of the free group $F = \pi_1(K^{(1)}, e) \cong \pi_1(\bigvee_n \mathbb{S}^1, e) \cong *_n\mathbb{Z}$. Their images under the map $\pi_1(K^{(1)}, e) \to \pi_1(K, e)$ induced by the inclusion $K^{(1)} \subseteq K$ will be denoted by the corresponding Greek letters $\alpha, \beta, \gamma, \ldots$.

p=1: Consider the 1-cell a and its lift $\widetilde{a}$ to the point $\widetilde{e}$. The endpoint of this lift is exactly the definition of $a\widetilde{e}$, and so it is easy to calculate the boundary as the difference of the endpoints:

$$\widetilde{d_1}(\widetilde{a}) = a\widetilde{e} - \widetilde{e} = (a-1)\widetilde{e}.$$

See figure 2 for an illustration.

p=2: Let $r \subseteq K$ be a 2-cell with boundary word $w(r)$, and $\widetilde{r}$ its unique lift to the point $\widetilde{e}$. Then we can write

$$\widetilde{d_2}(\widetilde{r}) = \sum_{a\in A} c_a \widetilde{a}$$

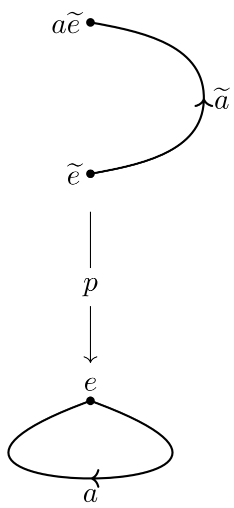

Figure 2: Lifting a 1-cell

for some coefficients $c_a \in \mathbb{Z}[\pi]$. After we determine these coefficients we can give the linear boundary map as a matrix and use it for calculations in the computer program.

As we are only interested in the incidence number of the lifted attaching map $\widetilde{\phi_r} : \mathbb{S}^1 \to \widetilde{K}$ we just have to count how often and in which orientation the lift of the attaching map meets the lifts of the 1-cells.

This has a purely algebraic description. Consider the following definition:

Definition 2.12 (Fox-derivatives, [Fox53])**.** Let F be a free group and $a \in F$ a generator. The *Fox-derivatives with respect to a* are the operators on the group ring $\mathbf{d_a} : \mathbb{Z}[F] \to \mathbb{Z}[F]$ satisfying

$$\mathbf{d_a}(b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases} \qquad \text{where } b \text{ is generator} \tag{11}$$

$$\mathbf{d_a}(ts) = \mathbf{d_a}(t) + t\mathbf{d_a}(s) \qquad \text{where } t, s \text{ are words} \tag{12}$$

These two rules completely determine the values of $\mathbf{d_a}$.

Example 2.13 (Fox-derivatives, [Fen83, 4.1])**.** Let $w = aba^{-1}b$, then

$$\mathbf{d_a}(w) = 1 - aba^{-1} \qquad \mathbf{d_b}(w) = a + aba^{-1}$$

△

Now we claim that the coefficients c_a can be calculated as the image of the operators $\mathbf{d_a}$ in $\mathbb{Z}[\pi]$. More precisely, define ∂_a as the composition

$$\partial_a = (\mathbb{Z}[F] \xrightarrow{\mathbf{d_a}} \mathbb{Z}[F] \to \mathbb{Z}[\pi]) \tag{13}$$

where the second map arises from the surjection

$$F = \pi_1(K^{(1)}, e) \to \pi_1(K, e) = \pi.$$

Proposition 2.14. *The images of the Fox-derivatives actually compute the boundary map, i.e. with the above definition of ∂_a we have*

$$\widetilde{d_2}(\widetilde{r}) = \sum_{a \in A} (\partial_a w)\widetilde{a} \tag{14}$$

Proof. At first a simplification because we can lift the 2-cells one at a time. Consider just the 1-skeleton of K with the single relation r attached, $L := K^{(1)} \cup r$. We will see how the attaching map $\phi_r : \mathbb{S}^1 \to K^{(1)}$ of r unfolds in the universal cover.

In example 2.11 we saw that the coefficient c_a for the 1-cell $\widetilde{a}$ is the total exponent with regard to a in the word describing the lifted attaching map $\widetilde{\phi_r}$. Here we have to count both the cell $\widetilde{a}$ and all its translates $g\widetilde{a}$ for $g \in \pi$. We proceed by induction on the length of the boundary word $w = w(r)$.

Base case 1: $w = b$, where b is some generator.
Here it is obviously correct: If $a = b$ the lifted relation is attached to the lifted 1-cell $\widetilde{a}$ by a degree one map. If $a \neq b$ the lifted relation does not meet the cell $\widetilde{a}$ or any of its translates, thus the degree is zero.

Base case 2: $w = a^{-1}$, the inverse of the generator a.
The Fox-derivative yields

$$\begin{aligned} 0 = \mathbf{d_a}(e) = \mathbf{d_a}(bb^{-1}) &= \mathbf{d_a}(b) + b\mathbf{d_a}(b^{-1}) \\ \Leftrightarrow \mathbf{d_a}(b^{-1}) &= -b^{-1}\mathbf{d_a}(b) \end{aligned}$$

So for $b \neq a, a^{-1}$ we get zero from the Fox-derivative, correspondingly the lift does not meet any of the relevant cells. For $b = a^{-1}$ the Fox-derivative gives $\mathbf{d_a}(a^{-1}) = -a^{-1}$. This fits together with the geometric lift: The lifted attaching map traverses the lift of the inverse a^{-1} in the opposite orientation, from the 0-cell $\widetilde{e}$ towards $a^{-1}\widetilde{e}$.

Inductive step: $w = ts$, the concatenation of the words t and s.
We assume the derivations are correct on words of length shorter than w. Instead of lifting the path w in one piece we can proceed in steps: First lift the initial path t to the point $\widetilde{e}$, after this the path s, but here we have to lift s to the endpoint $\tau\widetilde{e}$ of the lift of t (for an illustration take a look at figure 3). In the universal cover this lift of s is written as $\tau\widetilde{s}$. The degree of the entire lift is the sum of the two degrees of the individual lifts. By induction hypothesis for t this is given by the Fox-derivative $\mathbf{d_a}(t)$. It would be $\mathbf{d_a}(s)$ if we had lifted s to the start point $\widetilde{e}$, but here we have to multiply this with t, as we actually lifted the second part of the path to the start point $\tau\widetilde{e}$. Together we get that the degree is given by $\mathbf{d_a}(ts) = \mathbf{d_a}(t) + t\mathbf{d_a}(s)$, which justifies the second rule in definition 2.12.

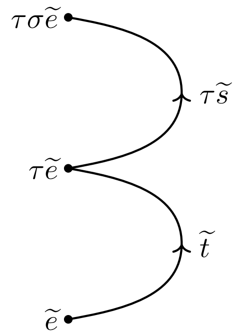

Figure 3: Lifting a concatenation of paths

Remember our conventions:
s, t are words in the 1-cells (generators of the free group $\pi_1(K^{(1)}, e)$).
σ is the image of s and τ the image of t in the fundamental group $\pi = \pi_1(K, e)$.

Remember that until now we are working in the subcomplex L with just one relation at a time. To get the coefficients c_a we just look at the complete complex K and thus at the image of the Fox-derivatives in the group ring over the fundamental group. $\square$

Remark 2.15 (Derivations, [Fen83, 4.2]). One can put this calculus of differentials in a broader context. The formula for the boundary of a lifted 2-cell justifies the following definition:

Definition 2.16. If G is any group let $\epsilon : \mathbb{Z}[G] \to \mathbb{Z}[G]$ be the linear map defined on the group elements by $g \mapsto 1$. ϵ is called the *augmentation map* (we will meet it again in 2.38).

A map $\mathbf{d} : \mathbb{Z}[G] \to \mathbb{Z}[G]$ is called a *derivation* if

- $\mathbf{d}$ is linear with respect to addition, i.e. $\mathbf{d}(\lambda + \mu) = \mathbf{d}(\lambda) + \mathbf{d}(\mu)$.
- $\mathbf{d}(\lambda\mu) = \mathbf{d}(\lambda)\epsilon(\mu) + \lambda\mathbf{d}(\mu)$.

From the definition of a derivation alone one can derive many consequences, but we will not pursue this any further.

To illustrate these homology calculations let us look more specifically at our situation and some concrete examples:

Our complex is given in the form of a group presentation $K = \langle A \mid R \rangle$. As both K and $\widetilde{K}$ just have cells in dimension less or equal than 2 we know that the chain groups will be zero from degree 3 on. The nonzero part of the chain complex is:

$$\begin{array}{c} 0 \\ \big\downarrow \\ \mathbb{Z}[\pi]^{\oplus \#\text{relations}} \\ \Big\downarrow \widetilde{d_2} \text{ given by Fox-derivatives} \\ \mathbb{Z}[\pi]^{\oplus \#\text{generators}} \\ \Big\downarrow \widetilde{d_1} \text{ given by } \widetilde{a} \mapsto (a-1)\widetilde{e} \\ \mathbb{Z}[\pi] \\ \big\downarrow \\ 0 \end{array} \tag{15}$$

The second homology can be calculated as

$$\begin{aligned} H_2(C_\bullet) &= \frac{\ker(\widetilde{d_2} : C_2(\widetilde{K}) \to C_1(\widetilde{K}))}{\operatorname{Im}(\widetilde{d_3} : C_3(\widetilde{K}) \to C_2(\widetilde{K}))} \\ &= \frac{\ker(\widetilde{d_2} : C_2(\widetilde{K}) \to C_1(\widetilde{K}))}{0} \\ &= \ker(\widetilde{d_2}) \end{aligned}$$

As $\mathbb{Z}$-module the group algebra is just a free module, and thus $C_2 = \bigoplus \mathbb{Z}[\pi]$ is free as a direct sum of free modules. Therefore the kernel is free abelian, because it is a subgroup of a free abelian group. This is very useful because after choosing a $\mathbb{Z}$-basis we have a representation of the homology which can be manipulated by a computer program.

Remark 2.17 (Rank of $\pi_2 K$)*.* There is a little trick to quickly calculate the $\mathbb{Z}$-rank of the second homotopy group. The sequence

$$0 \longrightarrow \pi_2(K) \longrightarrow (\mathbb{Z}[\pi]^{\oplus \#\text{rel.}}) \xrightarrow{\widetilde{d_2}} (\mathbb{Z}[\pi]^{\oplus \#\text{gen.}}) \xrightarrow{\widetilde{d_1}} \mathbb{Z}[\pi] \longrightarrow \mathbb{Z} \longrightarrow 0 \tag{16}$$

is exact: $\pi_2(K) \cong H_2(\widetilde{K})$ is the kernel of the second boundary map, $\mathbb{Z}$ the cokernel of the first (the image of $\widetilde{d_1}$ is generated by elements of the form $1-g$, thus in the cokernel all the group elements are identified with 1). The homology at the term $\mathbb{Z}[\pi]^{\oplus \#\text{gen.}}$ is trivial since $H_1(\widetilde{K}) \cong \pi_1(\widetilde{K})^{ab} \cong \{1\}$, thus it is exact there as well.

Moreover we know that the rank is additive on short exact sequences, therefore the alternating sum of the ranks in sequence 16 is zero. Considering everything as $\mathbb{Z}$-modules and using $\operatorname{rk}_{\mathbb{Z}}(\mathbb{Z}[\pi]) = \#\pi$ we get

$$\begin{aligned} 0 &= \operatorname{rk}_{\mathbb{Z}}(\pi_2 K) - \operatorname{rk}_{\mathbb{Z}}(\mathbb{Z}[\pi]^{\oplus \#\text{rel.}}) + \operatorname{rk}_{\mathbb{Z}}(\mathbb{Z}[\pi]^{\oplus \#\text{gen.}}) - \operatorname{rk}_{\mathbb{Z}}(\mathbb{Z}[\pi]) + \operatorname{rk}_{\mathbb{Z}}(\mathbb{Z}) \\ &= \operatorname{rk}_{\mathbb{Z}}(\pi_2 K) - (\#\text{rel.}) \cdot \#\pi + (\#\text{gen.}) \cdot \#\pi - \#\pi + 1 \end{aligned}$$

This yields the easy formula

$$\mathrm{rk}_{\mathbb{Z}}(\pi_2 K) = (\#\text{rel.} - \#\text{gen.} + 1) \cdot \#\pi - 1.$$

Example 2.18 (Cyclic group of order 2)**.** Let $K = \langle a \mid a^2 \rangle$, this is a presentation of the cyclic group with 2 elements, i.e. $K \cong_{(Grp)} \mathbb{Z}/(2)$. Topologically it is 2-dimensional real projective space, $K \approx \mathbb{RP}^2$.

We just have one generator (a) and one relation (a^2); the Fox-derivative of the relation is $\mathbf{d_a}(a^2) = \mathbf{d_a}(aa) \overset{\text{rule 2}}{=} \mathbf{d_a}(a) + a\mathbf{d_a}(a) = 1 + a$.

Plugging in $\#\text{gen.} = \#\text{rel.} = 1$ we obtain the chain complex

$$\begin{tikzcd} 0 \arrow[d] \\ \mathbb{Z}[\pi] \arrow[d, "\cdot(1+a)"] \\ \mathbb{Z}[\pi] \arrow[d, "\cdot(-1+a)"] \\ \mathbb{Z}[\pi] \arrow[d] \\ 0 \end{tikzcd} \tag{17}$$

For the kernel we get $\pi_2 K = \ker(\cdot(1+a)) \overset{(*)}{=} \mathbb{Z}(-1+a) \cong_{(\mathbb{Z}-Mod)} \mathbb{Z}$. The equalities are justified as follows: The inclusion "$\supseteq$" in $(*)$ is obvious because $(-1+a)(1+a) = -1 - a + a + a^2 = -1 + a^2 = -1 + 1 = 0$. On the other hand we know that the kernel is generated by one element when counting the ranks (see remark 2.17): $\mathrm{rk}_{\mathbb{Z}}(\pi_2 K) = (\#\text{rel.} - \#\text{gen.} + 1) \cdot \#\pi - 1 = (1 - 1 + 1) \cdot 2 - 1 = 1$.

One also checks that $-1 + a \leftarrow\!\shortmid 1$ gives the isomorphism with the free $\mathbb{Z}$-module $\mathbb{Z}$.

We should keep this example in mind as we will continue it in the next section. $\triangle$

Example 2.19 (Dihedral groups D_n)**.** Let $K = \langle s, d \mid s^2, d^n, (sd)^2 \rangle$, a presentation of the dihedral group of order $2n$, $K \cong_{(Grp)} D_n$.

Here we have the two generators s, d and three relations. The calculation of the Fox-derivatives is more involved:

$$\begin{aligned}
\mathbf{d_s}(s^2) &= 1 + s \\
\mathbf{d_d}(s^2) &= 0 \\
\mathbf{d_s}(d^n) &= 0 \\
\mathbf{d_d}(d^n) &= \mathbf{d_d}(dd^{n-1}) \\
&= \mathbf{d_d}(d) + d\mathbf{d_d}(d^{n-1}) \\
&= 1 + d(1 + d\mathbf{d_d}(d^{n-2})) \\
&= 1 + d(1 + d(1 + d(\ldots))) \\
&= 1 + d + d^2 + \ldots + d^{n-1} \\
\mathbf{d_s}((sd)^2) &= 1 + sd \\
\mathbf{d_d}((sd)^2) &= s + sds
\end{aligned}$$

Writing this in matrix form we get:

$$\begin{pmatrix} 1+s & 0 \\ 0 & 1+d+d^2+\ldots+d^{n-1} \\ 1+sd & s+sds \end{pmatrix}$$

Right multiplication with this matrix represents the second boundary map.

Here we have to be careful with the sides: For example in the case $n = 3$ we have $\pi = D_3 \cong \mathfrak{S}_3$ which is nonabelian, and thus $\mathbb{Z}[\pi]$ is a noncommutative ring. △

2.3 Whitehead's gamma functor: $\Gamma(-)$

In the motivation we defined the filler Γ_3^X for the certain exact sequence, and mentioned the result that it is isomorphic to $\Gamma(\pi_2 X)$. In this section we discuss the latter term, in particular we define *Whitehead's Gamma functor* (sometimes called *Whitehead's quadratic functor*).

In category theory we often look for the most efficient solution to an optimization problem, formally such objects satisfy an universal property. Here for any abelian group A we try to find a new group $\Gamma(A)$ which is universal with respect to quadratic maps, i.e. any quadratic map $A \to B$ between abelian groups should factor through $\Gamma(A)$.

Definition 2.20 (Quadratic map)**.** Let $A, B \in (Ab)$. A map $f : A \to B$ is called *quadratic* if $f(-a) = f(a)$ and if the function

$$\begin{aligned}
&A \times A \to B \\
&\quad (a, b) \mapsto f(a + b) - f(a) - f(b)
\end{aligned}$$

is bilinear.

The following proposition asserts the existence of such an object, by the usual argument it is unique up to unique isomorphism. The proof will give an explicit construction which we take as our definition.

Proposition 2.21 ([Whi50, Chapter II, 5.]). *For every abelian group A there is an abelian group $\Gamma(A)$ and an universal quadratic map $\gamma : A \to \Gamma(A)$, i.e. every quadratic map $f : A \to B$ uniquely factors through γ:*

$$\begin{tikzcd} A \arrow[rr, "f"] \arrow[dr, "\gamma"'] & & B \\ & \Gamma(A) \arrow[ur, dashed, "\exists!"'] & \end{tikzcd}$$

The construction is functorial, i.e. any $(\phi : A \to A') \in \mathrm{Mor}_{(Ab)}(A, A')$ induces a morphism $\Gamma(\phi) : \Gamma(A) \to \Gamma(A')$.

Proof. For each $a \in A$ define a generator $\gamma(a)$, we will look at words in these generators. Let $\sim$ be the relations

$$\gamma(-a) = \gamma(a) \tag{18}$$

$$\gamma(a+b+c)\gamma(b+c)^{-1}\gamma(c+a)^{1}\gamma(a+b)^{-1}\gamma(a)\gamma(b)\gamma(c) = e \tag{19}$$

for all $a, b, c \in A$. Define the gamma group as the quotient of the free group with generators $\gamma(a)$ modulo the relations $\sim$:

$$\Gamma(A) := *_{a\in A}\langle\gamma(a)\rangle / \sim$$

Observe that with $a = b = c = 0$ in relation 19 we get $\gamma(0) = e$. Now with $b = 0, a + c = d$ in relation 19 we have

$$\gamma(d)\gamma(c)^{-1}\gamma(d)^{-1}\gamma(c) = e$$

and thus $\Gamma(A)$ is abelian. Therefore we can write $\Gamma(A)$ and its relations additively from now on. The universal property of Γ follows from the universal property of the quotient (we mod out exactly the relation which describe quadratic maps).

For a homomorphism $(\phi : A \to A') \in \mathrm{Mor}_{(Ab)}(A, A')$ define

$$\gamma(a) \mapsto \gamma(\phi(a))$$

and this gives the homomorphism $\Gamma(\phi) : \Gamma(A) \to \Gamma(A')$. □

Whitehead continues to prove some formulas and properties of this functor. We just mention that $\Gamma(-)$ behaves well with respect to direct sums.

Theorem 2.22 ([Whi50, Chapter II, 7.])**.** *For two abelian groups K, L there is an isomorphism*

$$\Gamma(K \oplus L) \cong \Gamma(K) \oplus \Gamma(L) \oplus (K \otimes_{\mathbb{Z}} L).$$

We do not take a deeper dive into the general properties of this universal quadratic functor; rather we restrict our attention to the case of free abelian groups (as in the end we deal with $\Gamma(\pi_2 K)$ where $\pi_2 K$ is finitely generated free abelian). Let us assume $A = \mathbb{Z}^{\oplus I}$ from now on.

Proposition 2.23 ([Whi50, Chapter II, 5.(A)])**.** *Let A be free abelian with $(a_i)_{i \in I}$ a basis, assume that the index set I is ordered. Then the group $\Gamma(A)$ is free abelian and a basis is given by the set of the elements*

$$\{\gamma(a_i) \mid i \in I\} \cup \{\gamma(a_j + a_k) - \gamma(a_j) - \gamma(a_k) \mid j < k\}.$$

Example 2.24. For $\mathbb{Z}^{\oplus n}$ this generating set for $\Gamma(\mathbb{Z}^{\oplus n})$ contains

$$n + \#\{(j,k) \mid 1 \leq j < k \leq n\} = n + (n-1) + \ldots + 1 = \frac{n(n+1)}{2}$$

elements, i.e. $\Gamma(\mathbb{Z}^{\oplus n}) \cong \mathbb{Z}^{\oplus \frac{n(n+1)}{2}}$. This also follows by induction from theorem 2.22. △

In the free case this leads to a different (and often easier) description: For L finitely generated free abelian we could have defined $\Gamma(L)$ as the group of symmetric bilinear forms on L^*. Via the natural isomorphisms

$$\begin{aligned} \mathrm{Bil}_{\mathbb{Z}}(L^* \times L^*, \mathbb{Z}) &\cong \mathrm{Hom}(L^* \otimes_{\mathbb{Z}} L^*, \mathbb{Z}) = (L^* \otimes_{\mathbb{Z}} L^*)^* \\ &\cong L^{**} \otimes_{\mathbb{Z}} L^{**} \cong L \otimes_{\mathbb{Z}} L \end{aligned} \tag{20}$$

we can consider $\Gamma(L) \subseteq L \otimes_{\mathbb{Z}} L$: It is the subgroup generated by the symmetric tensors of the form $m \otimes m$ and $m \otimes n + n \otimes m$. This gives us a different description of a basis:

Proposition 2.25. *For L a free abelian group with basis (b_i) consider $\Gamma(L) \subseteq L \otimes_{\mathbb{Z}} L$. Then*

$$\{m \otimes m, m \otimes n + n \otimes m \mid m, n \in (b_i), m \neq n\}$$

is a basis for $\Gamma(L)$.

In the computer program the elements in $\Gamma(L)$ are represented as tensors, and we use the correspondence 20 to think of the tensors as bilinear maps (which in turn can be written as matrices).

2.4 The action of the fundamental group: $-/\pi_1 K$

2.4.1 An action of $\pi_1 K$ on $\Gamma(\pi_2 K)$

Recall that there is an action of the fundamental group on the homotopy groups:

$$\pi_1(X) \times \pi_n(X) \to \pi_n(X).$$

Here is one possible construction, for a more in depth discussion of basic properties take a look at [Hat02, 4.1].

Definition 2.26 (Action of π_1 on π_n)**.** There is a homotopy equivalence $f\colon \mathbb{S}^n \to \mathbb{S}^n \vee [0,1]$ taking the basepoint $* \in \mathbb{S}^n$ to 1 (the end where the unit interval is not attached). For each path α from x_0 to x_1 in X we get a basepoint changing homomorphism

$$\begin{aligned} \pi_n(X, x_0) &\to \pi_n(X, x_1) \\ [g\colon \mathbb{S}^n \to X] &\mapsto [(g \vee \alpha) \circ f] \end{aligned}$$

If $\alpha \in \pi_1(X, x_0)$ is a loop this does not alter the basepoint and so we get an action on $\pi_n(X, x_0)$.

There is another point of view, assuming that the space X has an universal cover $\widetilde{X}$: Proposition 2.7 asserts that $\pi_n(X)$ and $\pi_n(\widetilde{X})$ are pratically the same for $n \geq 2$. We can identify $\pi_1(X)$ as the deck transformation group, i.e. every $g \in \pi_1(X)$ gives a map

$$\begin{tikzcd} \widetilde{X} \arrow[rr, "g"] \arrow[dr] & & \widetilde{X} \arrow[dl] \\ & X & \end{tikzcd}$$

The deck transformation just permutes the different lifts of the maps $[\mathbb{S}^n \to X] \in \pi_n(X)$. We do not have to worry about the change in base points as the universal cover is simply connected.

Observe that this action gives the higher homotopy groups the structure of a left $\mathbb{Z}[\pi_1(X)]$-module.

Algebraically for our homological description of the second homotopy group this is the multiplication in the group ring: For a chain $x \in C_2(\widetilde{K}) \subseteq \mathbb{Z}[\pi]^{\oplus n}$ in homology and $g \in \pi_1(K) = \pi$ the image is just $g \cdot x$.

If a group G acts on a module L we also get a diagonal action on the tensor product $L \otimes L$ which on the elementary tensors is defined as

$$\begin{aligned} G \times (L \otimes L) &\to (L \otimes L) \\ (g, m \otimes n) &\mapsto (gm) \otimes (gn) \end{aligned} \tag{21}$$

For L a free module we consider $\Gamma(L) \subseteq L \otimes_{\mathbb{Z}} L$. Since the image of a symmetric tensor under this diagonal action is still symmetric we can restrict the action in 21 to get

$$G \times \Gamma(L) \to \Gamma(L).$$

In our topological application we now have

$$\pi_1(K) \times \Gamma(\pi_2(K)) \to \Gamma(\pi_2(K))$$

induced by the π_1-action on π_2.

Remark 2.27. One has to be careful with the distinction between a left and a right action. In this section we will suppose that the action of π_1 is defined from the left, but this choice is arbitrary. Mutatis mutandis, all the results in this chapter hold as well for right actions.

2.4.2 Quotients by group actions

Now we can consider the quotient of a module M on which we have a left action of the group G. In this quotient all the orbits are collapsed, i.e. a module element is identified with all its images under the action.

Definition 2.28 (Co-invariants)**.** Let M be an abelian group and G an arbitrary group with a left action $G \times M \to M$. Define the group of *co-invariants*:

$$M/G := M/\langle m - gm \rangle$$

Here $\langle m - gm \rangle$ is the subgroup of M generated by all elements of the form $m - gm$ for $m \in M, g \in G$.

Remark 2.29. These are called the *co-invariants* because intuitively M/G is the largest quotient of M which is stable under the action of the group G. Dually the group of *invariants* is the largest submodule of M on which G acts trivially.

To get a better grasp of this consider the following alternate algebraic description of the quotient operation:

Theorem 2.30 ([Bro12, II, 2, (2.1)])**.** *Let M be an abelian group with a group G acting on M. Then there is an isomorphism*

$$M/G \cong \mathbb{Z} \otimes_{\mathbb{Z}[G]} M. \tag{22}$$

Here M is a left $\mathbb{Z}[G]$-module via the action of G given and $\mathbb{Z}$ is a right $\mathbb{Z}[G]$-module through the trivial action $g \mapsto 1$.

Proof. In $\mathbb{Z} \otimes_{\mathbb{Z}[G]} M$ there is the identity

$$1 \otimes gm = 1 \cdot g \otimes m = 1 \otimes m.$$

Thus the map

$$\begin{aligned} M &\to \mathbb{Z} \otimes_{\mathbb{Z}[G]} M \\ m &\mapsto 1 \otimes m \end{aligned}$$

factors through the quotient to give a well-defined map

$$\begin{aligned} M/G &\to \mathbb{Z} \otimes_{\mathbb{Z}[G]} M \\ \overline{m} &\mapsto 1 \otimes m \end{aligned}$$

(Convention: $\overline{m}$ denotes the image of m in the quotient).

In the other direction it is easy to check that

$$\begin{aligned} \mathbb{Z} \times M &\to M/G \\ (a, m) &\mapsto a\overline{m} \end{aligned}$$

is bilinear, thus by the universal property of the tensor product there is a map

$$\begin{aligned} \mathbb{Z} \otimes_{\mathbb{Z}[G]} M &\to M/G \\ a \otimes m &\mapsto a\overline{m} \end{aligned}$$

These maps are inverses of one another. □

Remark 2.31 (Naturality). Observe that this identification with a tensor product is natural:

The isomorphism in 2.30, let us call it ψ, is functorial in M, i.e. for any map $(f : M \to M') \in \mathrm{Mor}_{(Ab)}(M, M')$ the following diagram commutes

$$\begin{tikzcd} M/G \arrow[r, "\psi_M"] \arrow[d, "\bar{f}"] & \mathbb{Z} \otimes_{\mathbb{Z}[G]} M \arrow[d, "\mathrm{id} \otimes f"] \\ M'/G \arrow[r, "\psi_{M'}"] & \mathbb{Z} \otimes_{\mathbb{Z}[G]} M' \end{tikzcd} \tag{23}$$

In other words: Theorem 2.30 gives a natural isomorphism of functors

$$-/G \cong \mathbb{Z} \otimes_{\mathbb{Z}[G]} -.$$

Using properties of the tensor product we conclude for the quotient operator $-/G$:

- The tensor product with a module is right-exact, i.e. it preserves cokernels. From 2.30 we know that the quotient operation $-/G$ is naturally identified with a tensor product, and consequently this functor is right-exact as well.

- For $F \cong \mathbb{Z}[G]^{\oplus n}$ a free $\mathbb{Z}[G]$-module we know

$$F/G \cong \mathbb{Z} \otimes_{\mathbb{Z}[G]} (\mathbb{Z}[G]^{\oplus n}) \cong (\mathbb{Z} \otimes_{\mathbb{Z}[G]} \mathbb{Z}[G])^{\oplus n} \cong \mathbb{Z}^{\oplus n}$$

  and thus the quotient will be a free $\mathbb{Z}$-module. Careful: We cannot apply this in our setting with $M = \Gamma(\pi_2 K)$, because even though M is always a free $\mathbb{Z}$-module it is not always free regarded as $\mathbb{Z}[\pi_1 K]$-module. Whether being free still holds after taking the quotient is the big question that we try to answer!

Applying theorem 2.30 with $G = \pi_1 K$ and $M = \Gamma(\pi_2 K)$ to our situation we get

$$\Gamma(\pi_2 K)/\pi_1 K \cong \mathbb{Z} \otimes_{\mathbb{Z}[\pi_1 K]} \Gamma(\pi_2 K)$$

Example 2.32 (2.18 continued)**.** The second homotopy group is equal to $\mathbb{Z}(-1+a)$. Now we get for the gamma module

$$\begin{aligned}\Gamma(\mathbb{Z}(-1+a)) \cong \Gamma(\mathbb{Z}) \cong \mathbb{Z} &\cong \mathbb{Z} \otimes_{\mathbb{Z}} \mathbb{Z} \\ 1 &\mapsto 1 \otimes 1\end{aligned}$$

Under these isomorphism the generator $1 \otimes 1 \in \mathbb{Z} \otimes_{\mathbb{Z}} \mathbb{Z}$ corresponds to $(-1+a) \otimes (-1+a) \in \mathbb{Z}(-1+a) \otimes_{\mathbb{Z}} \mathbb{Z}(-1+a)$.

Let us look at the action of the fundamental group. It is enough to determine the product of the generator of $\Gamma(M)$ and the generator $a \in \pi_1(K)$:

$$\begin{aligned}a \cdot ((-1+a) \otimes (-1+a)) &= (a(-1+a)) \otimes (a(-1+a)) \\ &= (-a+a^2) \otimes (-a+a^2) \\ &= (1-a) \otimes (1-a) \\ &= (-1)(-1+a) \otimes (-1)(-1+a) \\ &= (-1+a) \otimes (-1+a)\end{aligned}$$

So here the action of $\pi_1(K)$ on $\Gamma(\pi_2(K))$ is trivial! Thus taking the co-invariants does not change anything as everything is stable under the group action, $\Gamma(\pi_2(K))/\pi_1(K) \cong \Gamma(\pi_2(K)) \cong \mathbb{Z}$. △

Computationally such a quotient of a free R-module M can be calculated by describing the cokernel of a linear map. To each quotient M/G we associate a surjective module homomorphism:

$$\begin{aligned}R^{\oplus[\mathrm{rk}(M)\cdot(\#\text{generators of G})]} &\xrightarrow{\Lambda} M \\ r_{m_i,g_j} &\mapsto m_i - g_j m_i\end{aligned} \tag{24}$$

The cokernel of this map is exactly the module of G-co-invariants:

$$\operatorname{Coker}\Lambda = M/\operatorname{Im}(\Lambda) = M/\langle m_i - g_j m_i\rangle = M/G.$$

The domain and codomain of Λ are free modules, after choosing bases the map can be represented by a matrix with entries in R. Over rings which are nice enough there are algorithms for calculating the cokernel of a matrix:

Proposition 2.33 (Smith normal form, [HH70, 7.10, 7.15])**.** *Let $A \in \mathrm{Mat}(m \times n, R)$ be a nonzero matrix over a principal ideal domain R. Then there exist invertible matrices $S \in \mathrm{Mat}(m \times m, R), T \in \mathrm{Mat}(n \times n, R)$ such that the product is of the form*

$$SAT = \begin{pmatrix} \alpha_1 & 0 & 0 & & \dots & & 0 \\ 0 & \alpha_2 & 0 & & \dots & & 0 \\ 0 & 0 & \ddots & & & & 0 \\ \vdots & & & \alpha_r & & & \vdots \\ & & & & 0 & & \\ & & & & & \ddots & \\ 0 & & & \dots & & & 0 \end{pmatrix} \tag{25}$$

and the diagonal entries α_i satisfy $\alpha_i | \alpha_{i+1}$. We call this the Smith normal form *of the matrix A.*

The elements α_i are unique up to multiplication with units in R and are called elementary divisors *or* invariant factors*.*

Remark 2.34. There are efficient algorithms that calculate the Smith form of a matrix. If R is an Euclidean domain, i.e. there is an Euclidean function $d \colon R \setminus \{0\} \to \mathbb{N}_0$, it is even easier. Take a look at [HH70] for more details. Of course $\mathbb{Z}$ is a PID, in fact an Euclidean ring.

In the classification of finitely generated abelian groups in section 2.5 the Smith form will be useful again because the α_i turn out to describe the structure of the cokernel of left multiplication with A.

2.5 Classification of finitely generated abelian groups: $\mathrm{Tors}(-)$

Now we know that $\Gamma(\pi_2 K)$ is a finitely generated free abelian group. The images of the generators under the projection $\Gamma(\pi_2 K) \to \Gamma(\pi_2 K)/\pi$ generate the quotient, and thus this is still a finitely generated abelian group.

Those groups are completely classified:

Theorem 2.35 (Fundamental theorem of finitely generated abelian groups, [HH70, 10.3])**.** *Every finitely generated abelian group G is isomorphic to a direct sum of primary cyclic groups and infinite cyclic groups, i.e.*

$$G \cong \mathbb{Z}^{\oplus r} \oplus \bigoplus_{i=1}^{t} \mathbb{Z}/({p_i}^{n_i}) \tag{26}$$

where the p_i are prime numbers which are (up to rearranging the indices) uniquely determined by G.

Using theorem 2.35 we get an isomorphism

$$\Gamma(\pi_2 K)/\pi_1 K \cong \mathbb{Z}^{\oplus r} \oplus \underbrace{\mathbb{Z}/(q_1) \oplus \ldots \mathbb{Z}/(q_t)}_{\text{torsion part}} \tag{27}$$

and ask the question: Is this torsion part $\neq 0$?

Example 2.36 (2.18, 2.32 continued for the last time)**.** For $K = \langle a \mid a^2 \rangle$ the situation is easy: We have

$$\Gamma(\pi_2 K)/\pi_1 K \cong \mathbb{Z}$$

and this is free abelian and thus especially torsion-free. $\triangle$

This holds more generally for all presentations of this form of cyclic groups:

Example 2.37 (Cyclic groups)**.** Let $K = \langle a \mid a^n \rangle \cong_{(Grp)} \mathbb{Z}/(n)$. The topological space associated to it is a $\mathbb{S}^1$ (considered as a subset $\mathbb{S}^1 \subset \mathbb{C}$) into which we glue a two-disk $\mathbb{D}^2$ via the attaching map $z \mapsto z^n$. Observe that different from the case $n = 2$ this is not a manifold.

The Fox-derivative of the single relation can be calculated similar to example 2.19:

$$\mathbf{d_a}(a^n) = 1 + a + a^2 + \ldots + a^{n-1}$$

To enhance our intuition we describe the universal cover of this complex in more detail (take a look at figure 4): The n lifts of the unique 1-cell fit together at their endpoints to give a single $\mathbb{S}^1$ in the cover. Into this we glue n distinct $\mathbb{D}^2$ along their boundary, their interiors stay disjoint.

The cellular chain complex is (#gen. = #rel. = 1):

$$\begin{array}{c} 0 \\ \downarrow \\ \mathbb{Z}[\pi] \\ \downarrow{\scriptstyle \cdot(1+a+a^2+\ldots+a^{n-1})} \\ \mathbb{Z}[\pi] \\ \downarrow{\scriptstyle \cdot(-1+a)} \\ \mathbb{Z}[\pi] \\ \downarrow \\ 0 \end{array} \tag{28}$$

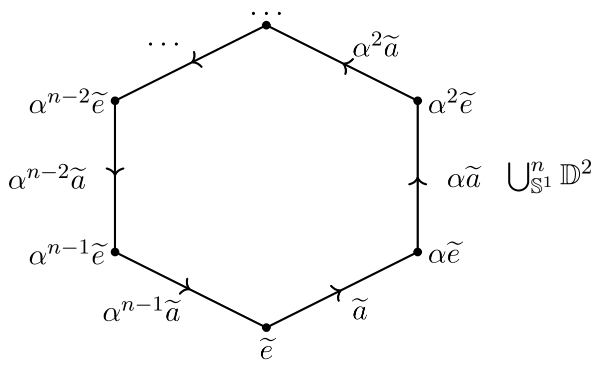

Figure 4: Universal cover of $K = \langle a \mid a^n \rangle$

The kernel as $\mathbb{Z}[\pi]$-module is generated by elements of the form $(1 - a), (1 - a^2), (1 - a^3), \dots, (1 - a^{n-1})$. One of the inclusion is again obvious, for the other we employ the trick from remark 2.17 and conclude $\mathrm{rk}_{\mathbb{Z}}(\pi_2 K) = (\#\text{rel.} - \#\text{gen.} + 1) \cdot \#\pi - 1 = (1 - 1 + 1) \cdot n - 1 = n - 1$.

Actually $(1 - a), (1 - a^2), (1 - a^3), \dots, (1 - a^{n-1})$ is already a basis for the kernel as a $\mathbb{Z}$-module. These elements give a basis of the so called *augmentation ideal*:

Definition 2.38. For a group ring $R[G]$ remember the *augmentation map*

$$\begin{aligned} \epsilon\colon R[G] &\to R \\ \sum r_i g_i &\mapsto \sum r_i \end{aligned}$$

(by definition of the group ring all the sums are finite). We call the two-sided ideal $I := \ker(\epsilon)$ the *augmentation ideal.*

Now we can refer to [HK88, Theorem 2.1] which asserts that for $L = I$ the quotient $\Gamma(L)/G$ is torsion-free. $\triangle$

One can probably already guess that the calculations needed are a lot more complicated if the group presentations are complex. Doing them by hand takes a lot of time and is prone to error. This should justify the effort to write a computer program to automate this task.

3 Computer implementation

3.1 The software which was used

The individual parts of the algorithm are implemented in the free and open source mathematics software system *SageMath* [Dev16]. This was the program of choice because some of the routines needed are already available in form of packages; whatever is missing can easily be added through a Python-based language.

The program is divided in subroutines corresponding to the subsections in section 2. Here we will discuss the most important functions in these modules `pi_2.sage`, `gamma.sage` and `torsion.sage`. During this discussion we will see what exactly the program computes and how this fits together with the topological application we are interested in.

3.2 Individual parts of the program

The input of the program is a group presentation, the combinatorial description of a 2-complex discussed in the previous section. It is enough to specify the number of generators and a list of words which are the relations of the group. We will see how the user can enter a group presentation into *Sage* in example 3.5.

Example 3.1 (Klein four-group)**.** To illustrate all the calculations we will follow the group presentation

$$K = \langle a, b \mid a^2, b^2, aba^{-1}b^{-1} \rangle$$

through the individual steps. This is a presentation of the *Klein four-group*, $K \cong_{(Grp)} \mathbb{Z}/(2) \times \mathbb{Z}/(2)$. Here the number of generators is 2 and the list of relations is $(a^2, b^2, aba^{-1}b^{-1})$. $\triangle$

The program makes some assumptions about the input: The group presentation has to be of a finitely presented finite group, otherwise the algorithm will never terminate.

Remark 3.2. This is because in a later step we choose an isomorphism with a subgroup of a permutation group, and there is a limit on the number of group elements the program is able to handle. If the input is too large the program will fail with an error message in this step.

If the group is (by theory) known to be finite the algorithms are guaranteed to terminate (if there is sufficient memory available), but the time needed for the calculation cannot be bounded a priori. Some problems involving group presentations are even undecidable, e.g. the word problem: Given a group presentation and a word in the generators there is no algorithm which is always able to decide if this word is the trivial element of the group [Boo58].

Thus it is the user's responsibility to only call the program with a valid input!

3.2.1 `pi_2.sage`

The first module contains all the functions for calculating the second homotopy group of a 2-complex associated to a group presentation.

Remark 3.3 (Credit). In here we rely on the *Sage*-packages

- `sage.groups.free_group` written by Miguel Angel Marco Buzunariz and Volker Braun
- `sage.groups.finitely_presented` written by Miguel Angel Marco Buzunariz
- `sage.algebras.group_algebra.GroupAlgebra` written by David Loeffler, Martin Raum, John Palmieri.

Furthermore these *Sage*-modules provide an interface to the computer algebra system GAP (Groups, Algorithms, Programming) [GAP16] where most of the algorithms are implemented.

Recall that the second boundary map

$$\mathbb{Z}[\pi]^{\oplus \#\text{relations}} \xrightarrow{\widetilde{d_2}} \mathbb{Z}[\pi]^{\oplus \#\text{generators}}$$

is described by the Fox-derivatives of the relations. Now we want to give an explicit matrix representation of $\widetilde{d_2}$. We saw that $\widetilde{d_2}$ is $\mathbb{Z}[\pi]$-linear, so it is enough to know the images of the lifted relations $\widetilde{r}$ for $r \in R$. Writing these images as the rows of a matrix $D \in \mathrm{Mat}(\#\text{rel.} \times \#\text{gen.}, \mathbb{Z}[\pi])$ we have a representation of the map with which we can do further calculations. For a vector $v \in C_2(K) = \mathbb{Z}[\pi]^{\oplus \#\text{rel.}}$ its image under $\widetilde{d_2}$ is given by the matrix product $v \cdot D$. Observe that this will give a left π-equivariant map because for $g \in \pi$ we have

$$(gv) \cdot D = g(v \cdot D).$$

Thus we have to multiply from the left to get the images of the translates $g\widetilde{r}$, and the π_1-action also is from the left.

For abelian groups π the choice between matrix multiplication with D from the right or with the transpose D^t from the left does not change the end result we are interested in. But for nonabelian π the multiplication in the group ring $\mathbb{Z}[\pi]$ is noncommutative, and here the choice matters: Using the map $v \mapsto v \cdot D$ the multiplications with the matrix entries in the group ring are done from the right, taking $v \mapsto D^t \cdot v$ the multiplications are done from the other side.

Once and for all choose an ordering $(g_1, g_2, \ldots, g_n)$ of the generators and $(r_1, r_2, \ldots, r_m)$ of the relations in the group presentation. In *Sage* these are stored as Python tuples and thus are ordered. For practical reasons we will just keep the order in which those are entered by the user. The entries of the matrix D are given by the Fox-derivatives of the relation with respect to the generators. As a convention let the first index in a matrix denote the row, the second the column. Then the entries are given as

$$D_{ij} = \partial_{\mathbf{g_j}}(r_i). \tag{29}$$

The input G (our group presentation) is of *Sage*-type
<class 'sage.groups.finitely_presented.FinitelyPresentedGroup_with_category'>.

Take a look at this code fragment adapted from the package sage.groups.free_group which calculates the Fox-derivatives.

```
def fox(word, gen, im_gens=None, ring=None):
l = list(word.Tietze())
if im_gens is None:
    # ...
else:
    # ...
    symb = list(im_gens)
    symb += reversed([R((a.trailing_support())^(-1)) for
        a in im_gens]) # We had to change this line
i = gen.Tietze()[0]   # So 'gen' is the 'i'-th
                      # generator of the free group.
a = R.zero()
coef = R.one()
# Here the actual computation takes place
while len(l) > 0:
    b = l.pop(0)   # Take the first letter of the word
    if b == i:     # It is the generator 'gen'
        a += coef * R.one()
        coef *= symb[b-1]
    elif b == -i: # It is the inverse of the generator
        a -= coef * symb[b]
        coef *= symb[b]
    elif b > 0:    # The other cases
        coef *= symb[b-1]
    else:
        coef *= symb[b]
return a
```

The while-loop just applies the two defining rules in 2.12. The variable a (initialized with $0 \in R$) holds the intermediate result, the list l the part of the word that still needs processing. If the first letter b in l is the generator or the inverse of the generator with regard to which we calculate the derivative, we just add one respectively minus this generator to a. Then we continue with the rest of the word according to the second rule in 2.12. If b is neither gen nor its inverse the derivative just jumps over it due to

$$\begin{aligned}\mathbf{d_g}(hw) &= \underbrace{\mathbf{d_g}(h)}_{=0} + h\mathbf{d_g}(w)\\ &= h\mathbf{d_g}(w)\end{aligned}$$

for a generator g, a letter $h \neq g, g^{-1}$ and any word w. The latter two if-clauses handle this case.

The parameter im_gens is used later: It computes the image of the operator $\mathbf{d_a}$ in the group ring $\mathbb{Z}[\pi]$ (remember that this is called $\partial_{\mathbf{a}}$).

From this step on we do not use the group presentation directly anymore because (as mentioned above) it is not easy to decide whether two different words represent the same element. As a countermeasure the program finds an isomorphism from G to a subgroup of some permutation group $\mathfrak{S}_k$. The

Sage-function `.as_permutation_group()` provides an interface to an algorithm implemented in GAP (*coset enumeration*, see [Neu82]) which given a group presentation of a finite group, returns such a subgroup of $\mathfrak{S}_k$ with generators corresponding to $g_1, \ldots, g_n$. In the source code an upper bound for the number of cosets has to be provided, the calculation is cancelled in case the limit is reached. If need be the user could change this limit.

Next we proceed to the group ring in *Sage* by calling the constructor `GroupAlgebra(G.as_permutation_group(), ZZ)`. When we give the generators of the group ring to the `fox`-function in the argument `im_gens` it returns the image $\partial_{\mathbf{a}}$ of the derivation operators in this group ring.

Now that we are able to calculate Fox-derivatives the next goal is to determine the second homology of the complex, which amounts to finding a basis for the kernel of the second boundary map. Because internally *Sage* already has an order of the group elements it automatically provides a $\mathbb{Z}$-module isomorphism

$$\begin{aligned} \chi\colon \mathbb{Z}[\pi] &\xrightarrow{\cong} \mathbb{Z}^{\oplus\#\pi} \\ g_i &\mapsto e_i \end{aligned} \tag{30}$$

i.e. a $\mathbb{Z}$-basis for $\mathbb{Z}[\pi]$. Thus from now on we can represent group algebra elements as vectors with $\mathbb{Z}$-entries, and the endomorphism which is multiplication with an group algebra element $h \in \mathbb{Z}[\pi]$ as a matrix $R_h \in \mathrm{Mat}(\#\pi \times \#\pi, \mathbb{Z})$.

$$\begin{tikzcd} \mathbb{Z}[\pi] \arrow[r, "\cdot h"] \arrow[d, "\chi"] & \mathbb{Z}[\pi] \arrow[d, "\chi"] \\ \mathbb{Z}^{\oplus\#\pi} \arrow[r, "R_h"] & \mathbb{Z}^{\oplus\#\pi} \end{tikzcd} \tag{31}$$

Remark 3.4. The diagram 31 shows this for right multiplication, of course this works for left multiplication $h\cdot$ as well resulting in a matrix L_h. For π nonabelian there are $h \in \mathbb{Z}[\pi]$ with $L_h \neq R_h$!

This allows us to convert a matrix with entries in $\mathbb{Z}[\pi]$ to an (albeit larger) matrix with $\mathbb{Z}$-coefficients that still acts by right-multiplication. Concretely we use the `.to_matrix(side='right')` function on a group algebra element and map this function onto all the matrix entries. We actually have to use a transpose here because of the conventions in *Sage*: `h.to_matrix(side='right')` returns a matrix `R`, and **left** multiplication with `R` represents **right** multiplication with `h` on the group algebra. Since we later take the matrix product with D from the right, mapping `.to_matrix(side='right').transpose()` to the matrix returned by `second_boundary_matrix(G)` (and converting the resulting block matrix into a regular matrix) gives the correct integer representation of D.

Next we calculate the kernel of this integer matrix by calling the *Sage* method `.left_kernel()`. This gives a basis for the free module which is the second homology/homotopy group.

The function `pi_2(grouppresentation)` combines all this and given a group presentation just returns the second homotopy group in the form of a free $\mathbb{Z}$-module with a basis.

Example 3.5 (3.1 continued: boundary map and π_2)**.** We will unravel the steps `pi_2` performs on the input $K = \langle a, b \mid a^2, b^2, aba^{-1}b^{-1} \rangle$.

Remark 3.6 (on how the *Sage*-examples are presented)*.* All the examples are held in a way that hopefully one does not need prior knowledge of *Sage* to understand them.

We will use the command-line interface of *Sage*. After starting the program with the command `sage` the user is confronted with a *REPL*, a read-evaluate-print-loop. In the first step the user can enter commands, define variables and call functions or methods on them. Then the program evaluates the statement and prints the result to the console. Finally it loops to the beginning and the user can enter a new expression.

In this document the same *Sage*-session will often be split up in smaller parts to allow some explanatory text in between. The line numbers are a good indication for the continuation of one session, but sometimes we will implicitly assume that some of the variables used have been defined in an earlier block. From the context and the preceding examples it will be clear which values are meant.

We show how one would define the group presentation and work with it in such an interactive *Sage* shell session.

```
# Create a free group with two generators
sage: F = FreeGroup(2)
sage: words = [F([1])^2, F([2])^2, F([1])*F([2])*F([1])
    ^-1*F([2])^-1)]
# Now 'words' is the list of relations
sage: words
[x0^2, x1^2, x0*x1*x0^-1*x1^-1]
# Take the quotient F modulo the relations
sage: G = F.quotient(words)
sage: G
Finitely presented group < x0, x1 | x0^2, x1^2, x0*x1*x0
    ^-1*x1^-1 >
```

Next the program uses the algorithm in `fox` to calculate the Fox-derivatives $\mathbf{d}_{\mathbf{x_i}}$ of the relations `x0^2, x1^2, x0*x1*x0^-1*x1^-1` with regard to the generators `x0, x1`. Those will be written into a matrix according to the convention in equation 29.

```
sage: DD = second_boundary_matrix(G)
sage: DD
[            B[1] + B[x0]                                0]
[                       0                      B[1] + B[x1]]
[ B[1] - B[x0*x1*x0^-1] B[x0] - B[x0*x1*x0^-1*x1^-1]]
```

When reading this one has to ignore the "`B`" as this is just the way *Sage* displays the elements in a group algebra. We could verify the output by calculating the derivations by hand:

$$\begin{aligned} D_{11} = \partial_{\mathbf{x_0}}(x_0^2) &= \partial_{\mathbf{x_0}}(x_0 x_0) \\ &= \partial_{\mathbf{x_0}}(x_0) + x_0 \partial_{\mathbf{x_0}}(x_0) \\ &= 1 + x_0 \\ D_{12} = \partial_{\mathbf{x_1}}(x_0^2) &= \partial_{\mathbf{x_1}}(x_0 x_0) \\ &= \partial_{\mathbf{x_1}}(x_0) + x_0 \partial_{\mathbf{x_1}}(x_0) \\ &= 0 \\ D_{21} = \partial_{\mathbf{x_0}}(x_1^2) &= \dots \end{aligned}$$

Now we pass to an isomorphic subgroup of $\mathfrak{S}_k$:

```
sage: G.as_permutation_group()
Permutation Group with generators [(1,2)(3,4),(1,3)(2,4)]
sage: second_boundary_matrix_with_permutation(G)
[ () + (1,2)(3,4)                    0]
[                0    () + (1,3)(2,4)]
[ () - (1,3)(2,4)  -() + (1,2)(3,4)]
```

Sage represents the elements in $\mathfrak{S}_k$ as a string that defines a permutation using disjoint cycle notation. `()` corresponds to the identity permutation. The output tells us the isomorphism calculated by GAP is given on the generators as

$$\begin{aligned} \langle x_0, x_1 \mid x_0^2, x_1^2, x_0 x_1 x_0^{-1} x_1^{-1} \rangle &\xrightarrow{\cong} \langle (1,2)(3,4), (1,3)(2,4) \rangle \subset \mathfrak{S}_4 \\ x_0 &\mapsto (1,2)(3,4) \\ x_1 &\mapsto (1,3)(2,4) \end{aligned}$$

Now let us create the group ring of this permutation subgroup:

```
# Construct the group ring
sage: Zperm = GroupAlgebra(G.as_permutation_group(), ZZ)
sage: Zperm
Group algebra of group "Permutation Group with generators
    [(1,2)(3,4), (1,3)(2,4)]" over base ring Integer
   Ring
```

Sage chose an ordering of the group ring's basis vectors and uses this to convert group ring elements to vectors (looking at them as elements in a free module) and matrices (looking at multiplication with them as an endomorphism).

```
# This displays the ordering of the basis elements
# and is the basis we will take for the group ring
# as free Z - module
sage: list(Zperm.basis())
[(), (1,2)(3,4), (1,3)(2,4), (1,4)(2,3)]
# Assign names to the images of the group's
# generators in the group ring
sage: a = Zperm(G.as_permutation_group().gens()[0])
sage: b = Zperm(G.as_permutation_group().gens()[1])
# Some examples illustrating the conversion
# to integer vectors
sage: a.to_vector()
(0, 1, 0, 0)
sage: b.to_vector()
(0, 0, 1, 0)
sage: (a+b).to_vector()
(0, 1, 1, 0)
sage: (a*b).to_vector()
(0, 0, 0, 1)
# And conversion to integer matrices
sage: a.to_matrix(side='left')
[0 1 0 0]
[1 0 0 0]
[0 0 0 1]
[0 0 1 0]
```

In the matrix-conversion function `.to_matrix()` of a group algebra element h we choose the side of multiplication. Mapping this conversion on all the entries of the boundary matrix we get a map between free $\mathbb{Z}$-modules and just have to calculate its kernel with the *Sage*-routine `.left_kernel()`:

```
sage: DDint = second_boundary_matrix_with_integers(G)
sage: DDint
[ 1  1  0  0| 0  0  0  0]
[ 1  1  0  0| 0  0  0  0]
[ 0  0  1  1| 0  0  0  0]
[ 0  0  1  1| 0  0  0  0]
[-----------+-----------]
[ 0  0  0  0| 1  0  1  0]
[ 0  0  0  0| 0  1  0  1]
[ 0  0  0  0| 1  0  1  0]
[ 0  0  0  0| 0  1  0  1]
[-----------+-----------]
[ 1  0 -1  0|-1  1  0  0]
[ 0  1  0 -1| 1 -1  0  0]
[-1  0  1  0| 0  0 -1  1]
[ 0 -1  0  1| 0  0  1 -1]
sage: DDint.left_kernel()
Free module of degree 12 and rank 7 over Integer Ring
Echelon basis matrix:
[ 1  0  0 -1  0  0  0  0  0  0  1  1]
[ 0  1  0 -1  0  0  0  0  0  0  1  1]
[ 0  0  1 -1  0  0  0  0  0  0  0  0]
[ 0  0  0  0  1  0  0 -1  0 -1  0 -1]
[ 0  0  0  0  0  1  0 -1  0  0  0  0]
[ 0  0  0  0  0  0  1 -1  0 -1  0 -1]
[ 0  0  0  0  0  0  0  0  1  1  1  1]
```

Here the rows of the Echelon basis matrix give a basis for the second homology as free $\mathbb{Z}$-module and by our earlier discussion this is also a description of $\pi_2(K)$.

The convenient function `pi_2(grouppresentation)` combines this: When called with the group presentation `G` it automatically performs all the steps above and returns without further ado the second homotopy group.

```
sage: pi_2(G)
Free module of degree 12 and rank 7 over Integer Ring
Echelon basis matrix:
[ 1  0  0 -1  0  0  0  0  0  0  1  1]
[ 0  1  0 -1  0  0  0  0  0  0  1  1]
# ... (see above)
[ 0  0  0  0  0  0  0  0  1  1  1  1]
```

This concludes the first step in the calculation of this example. △

3.2.2 `gamma.sage`

Next up we describe the representation of $\Gamma(M)$ for M a free $\mathbb{Z}$-module in this computer program. This part also contains the functions needed to calculate the image $g \cdot m, g \in \pi, m \in \Gamma(M)$ for the action of the fundamental group.

Remark 3.7 (Credit). Here we use the following *Sage*-packages:

- `sage.tensor.modules.`
  `finite_rank_free_module.FiniteRankFreeModule`

- `sage.tensor.modules.tensor_free_module.TensorFreeModule`

both written by Eric Gourgoulhon and Michal Bejger

Recall the output format of the function `pi_2(grouppresentation)`: It returns a basis for a free $\mathbb{Z}$-module which is a description of the second homology group (in turn isomorphic to the second homotopy group). Only the $\mathbb{Z}$-module structure is explicit, but we know that actually we are dealing with a $\mathbb{Z}[\pi]$-module. The additional structure is not forgotten and we will see how it can be recovered.

To get the group ring description of the kernel we just read the vectors $v \in \mathbb{Z}^{\oplus \#\pi \cdot \#\text{rel.}}$ as vectors in $(\mathbb{Z}^{\oplus \#\pi})^{\oplus \#\text{rel.}}$. The smaller components of length $\#\pi$ give the coefficients of the group algebra element in the basis we chose when using the isomorphism χ in equation 30.

Example 3.8 (Regaining the $\mathbb{Z}[\pi]$-structure)**.** We can interpret the output describing $\pi_2(G)$ as a $\mathbb{Z}[\pi]$-module again.

For visualization insert the separators "|":

```
sage: H_2 = pi_2(G)
sage: H_2
[ 1  0  0 -1 | 0  0  0  0 | 0  0  1  1]
[ 0  1  0 -1 | 0  0  0  0 | 0  0  1  1]
[ 0  0  1 -1 | 0  0  0  0 | 0  0  0  0]
[ 0  0  0  0 | 1  0  0 -1 | 0 -1  0 -1]
[ 0  0  0  0 | 0  1  0 -1 | 0  0  0  0]
[ 0  0  0  0 | 0  0  1 -1 | 0 -1  0 -1]
[ 0  0  0  0 | 0  0  0  0 | 1  1  1  1]
```

Remember the basis for the group ring we fixed before:

```
sage: list(Zperm.basis())
[(), (1,2)(3,4), (1,3)(2,4), (1,4)(2,3)]
```

Now for example the vector `[1, 0, 0, -1]` corresponds to the group algebra element `() - (1,4)(2,3)`. Using this on each of the small components of the rows we get back the $\mathbb{Z}[\pi]$-module structure:

```
sage: pi_2_with_permutation(G)
[         () - (1,4)(2,3)|                       0|          (1,3)(2,4) + (1,4)(2,3)]
[(1,2)(3,4) - (1,4)(2,3)|                       0|          (1,3)(2,4) + (1,4)(2,3)]
[(1,3)(2,4) - (1,4)(2,3)|                       0|                                 0]
[                       0|         () - (1,4)(2,3)|        -(1,2)(3,4) - (1,4)(2,3)]
[                       0|(1,2)(3,4) - (1,4)(2,3)|                                 0]
[                       0|(1,3)(2,4) - (1,4)(2,3)|        -(1,2)(3,4) - (1,4)(2,3)]
[                       0|                       0|() + (1,2)(3,4) + (1,3)(2,4) + (1,4)(2,3)]
```

△

Algebraically the action of the fundamental group on second homology is just left multiplication of the vector with the group element in the algebra. Now for $v \in \mathbb{Z}^{\oplus \#\pi \cdot \#\text{rel.}}$ we look at $h \in \pi$ as the matrix L_h (see remark 3.4), this representation is with regard to the $\mathbb{Z}$-basis of $\mathbb{Z}[\pi]$ we fixed before. Again we have to be careful with the side of the action as π does not have to be abelian. To ensure compatibility with the choice that the boundary matrix acts from the right we have to take the left action $g \mapsto h \cdot g$ on the group algebra.

The function `action(h, vect)` splits the vector `vect` into the components corresponding to a single group algebra element, then applies the matrix `h.to_matrix(side='left')` to these smaller vectors. The multiplication with the matrix `h.to_matrix(side='left')` $\in \mathrm{Mat}(\#\pi \times \#\pi, \mathbb{Z})$ has to be done from the left (this is the convention in *Sage* mentioned before), and the linear transformation carried out by left-multiplication with this matrix is the left action of the element $h \in \mathbb{Z}[\pi]$ on the group algebra.

Now that tensors come into play we convert $\pi_2 K$ to an object of *Sage*-type `FiniteRankFreeModule`. This is done by calling the constructor with the $\mathbb{Z}$-rank of `pi_2(K)` we got from the earlier calculations. Objects of type `FiniteRankFreeModule` do not automatically have a basis associated with them. We can create a default basis with the `.basis()` command, and in our interpretation this basis will be in one-to-one correspondence with the submodule basis of the kernel returned by the function `pi_2()`.

To deal with the fact that we need to represent the group action with respect to different bases we introduce a relative version of the action-function called action_in_coordinates(g, vect, module). This begins by doing the same as action(g, vect) on the group ring element and the vector, but then returns the coordinates of the image with respect to the default basis of module.

Example 3.9 (Converting bases and using .coordinates(v)). Consider the submodule that is the second homotopy group of K.

```
sage: H_2.basis()
[
(1, 0, 0, -1, 0, 0, 0,  0, 0,  0, 1,  1),
(0, 1, 0, -1, 0, 0, 0,  0, 0,  0, 1,  1),
(0, 0, 1, -1, 0, 0, 0,  0, 0,  0, 0,  0),
(0, 0, 0,  0, 1, 0, 0, -1, 0, -1, 0, -1),
(0, 0, 0,  0, 0, 1, 0, -1, 0,  0, 0,  0),
(0, 0, 0,  0, 0, 0, 1, -1, 0, -1, 0, -1),
(0, 0, 0,  0, 0, 0, 0,  0, 1,  1, 1,  1)
]
```

H_2's basis is the ordered list above. Given a vector with entries in the integers which is contained in the submodule H_2, we can get its coordinates with respect to the submodule basis of H_2:

```
sage: H_2.coordinates([1,1,0,-2,0,0,0,0,0,0,2,2])
[1, 1, 0, 0, 0, 0, 0]
```

Read this as: The vector [1, 1, 0, -2, 0, 0, 0, 0, 0, 0, 2, 2] is the sum of the first and the second basis vector of H_2. △

Next up are four functions to create elements in the tensor product and their images under the group action. In the following m and n are elements of the module

$$\texttt{module} \cong \mathbb{Z}^{\oplus k}$$

given in the coordinates of the canonical basis

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, e_k = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

M is of type FiniteRankFreeModule with $\mathrm{rk}_{\mathbb{Z}}(M) = k$, its default basis is in one-to-one correspondence with the basis of module (but neither of those needs to be the canonical basis $e_1, \ldots, e_k$).

- diagonal_tensor(m, module, M) returns the diagonal element $m \otimes m$ in the tensor product $M \otimes M$.
- image_of_diagonal_tensor(g, m, module, M) returns the image of a diagonal element under the action of g, i.e. $(gm) \otimes (gm) \in M \otimes M$.
- other_tensor(m, n, module, M) returns the tensor $m \otimes n + n \otimes m \in M \otimes M$

- `image_of_other_tensor(g, m, n, module, M)` returns the image of such a tensor under the action, i.e. $(gm) \otimes (gn) + (gn) \otimes (gm)$.

Observe that all the return values of these functions lie in $\Gamma(M)$ as they are all symmetric tensors.

In subsection 3.2.3 we will see how the elements in the tensor product are represented internally. We will need this there as we will have to find linear combinations of the basis symmetric tensors that yield the tensors returned by the four functions above. Let us conclude the current subsection with an example:

Example 3.10 (Using the tensor product)**.** We generate some example tensors just to see how everything works.

```
# Take v to be the sum of the
# first two basis vectors of H_2
sage: v = H_2.basis()[0] + H_2.basis()[1]
sage: v
(1, 1, 0, -2, 0, 0, 0, 0, 0, 0, 2, 2)
sage: H_2.coordinates(v)
[1, 1, 0, 0, 0, 0, 0]
# Generate a FiniteRankFreeModule
sage: M = FiniteRankFreeModule(ZZ, H_2.rank())
sage: M
Rank-7 free module over the Integer Ring
# Fix a default basis for M,
# everything in the tensor product
# will be given with respect to this basis
sage: f = M.basis('f')
sage: f
Basis (f_0,f_1,f_2,f_3,f_4,f_5,f_6) on the Rank-7 free
    module over the Integer Ring
# From the convention above:
# f_0 corresponds to H_2.basis()[0]
# f_1 corresponds to H_2.basis()[1] ...
sage: M([0,0,0,1,0,0,0]).display(f)
f_3
# '*' is the operator to create
# elements in the tensor product
# 'Free module of type-(2,0) tensors on the Rank-7 free
    module over the Integer Ring'
# Example: tensor product of the first
# and the second basis vector of M
sage: (M([1,0,0,0,0,0,0])*M([0,1,0,0,0,0,0])).display()
f_0*f_1
# diagonal_tensor() called with argument v
# returns the tensor v*v=(f_0 + f_1)*(f_0 + f_1)
sage: diagonal_tensor(v, H_2, M).display()
f_0*f_0 + f_0*f_1 + f_1*f_0 + f_1*f_1

# Be careful not to mix up the bases!
# This does not give v*v because we are confusing
# the basis of H_2 with the basis of the new module M
sage: (M(v)*M(v)).display()
f_0*f_0 + f_0*f_1 - 2 f_0*f_3 + f_1*f_0 + f_1*f_1 - 2 f_1
    *f_3 - 2 f_3*f_0 - 2 f_3*f_1 + 4 f_3*f_3
```

Similarly all the functions involving the action of a group ring element operate. △

3.2.3 `torsion.sage`

Now that we collected all the ingredients the last part is throwing everything together. The module `torsion.sage` involves three main steps:

- `tensors_identified_with_zero()` gives a generating set of the module we collapse, i.e. a list of generators for $\langle m - gm \mid m \in \Gamma(M), g \in \pi \rangle$.
- `matrix_of_surjection()` uses this list to construct the surjection 24 whose cokernel we want to classify. For this it takes the tensors from the first step, writes them as a linear combination of basis vectors of Γ and arranges the coefficients as the columns of a matrix.
- In the last step it classifies the matrix with some Smith-algorithm.

Part 1 - tensors_identified_with_zero(grouppresentation):
For generating $\langle m - gm \mid m \in \Gamma(M), g \in \pi \rangle$ it is obviously enough to restrict to a generating set of $\Gamma(M)$. One can obtain one of these by taking a basis of M and then looking at all elements of the form $m \otimes m$ for m in the basis, and $m \otimes n + n \otimes m$ for pairs (m, n) of basis vectors. As the pairs (m, n) and (n, m) yield the same in the second case we just have to take one of those. Less obvious might be the fact that it is also enough to restrict to a generating set of the group π. This is because we can write

$$m - (gh)m = (m - gm) + g(m - hm).$$

and we took the elements in brackets beforehand.

`tensors_identified_with_zero_use_just_generators()`
iterates through two loops: The first calculating in each step $m \otimes m - g(m \otimes m)$ for m in a basis of M and g generator of π, the second yielding $m \otimes n + n \otimes m - g(m \otimes n + n \otimes m)$. For efficiency we call `.to_matrix()` just once for each generator of the group and pass the matrix representations of $g \in \pi$ through all the function calls. Here it is very convenient that functions in Python are not strictly typed: One could write a function that can accept both group algebra elements and matrices, and acts according to the type of input given.

Example 3.11. Take a look at an excerpt from the list of tensors:

```
sage: tens=tensors_identified_with_zero_use_just_generators(G)
# For displaying the result we map the
# .display() function onto the list
sage: map(lambda x: x.display(), tens)
[f_0*f_0 - f_1*f_1 + f_1*f_2 + f_2*f_1 - f_2*f_2,
 f_0*f_0 - f_1*f_1 + f_1*f_2 + f_1*f_6 + f_2*f_1 - f_2*f_2 - f_2
    *f_6 + f_6*f_1 - f_6*f_2 - f_6*f_6,
```

```
 -f_0*f_0 + f_0*f_2 + f_1*f_1 + f_2*f_0 - f_2*f_2,
 f_1*f_6 + f_6*f_1 - f_6*f_6,
 0,
 -f_0*f_0 + f_0*f_1 + f_1*f_0 - f_1*f_1 + f_2*f_2,
 f_3*f_3 - f_4*f_4 + f_4*f_5 + f_4*f_6 + f_5*f_4 - f_5*f_5 - f_5
     *f_6 + f_6*f_4 - f_6*f_5 - f_6*f_6,
 f_3*f_3 - f_4*f_4 + f_4*f_5 + f_5*f_4 - f_5*f_5,
 -f_3*f_3 + f_3*f_5 + f_4*f_4 + f_5*f_3 - f_5*f_5,
 0,
# ... (the entire list contains 56 entries)
 2 f_5*f_6 + 2 f_6*f_5 + 2 f_6*f_6,
 -f_3*f_6 + f_4*f_6 + f_5*f_6 - f_6*f_3 + f_6*f_4 + f_6*f_5]
```

△

Part 2 - matrix_of_surjection(): We think of the tensors as bilinear maps from now on, and we can represent these bilinear maps via their Gramian matrices. In *Sage* the method `.components()[:]` gives back this matrix representation:

Example 3.12 (Converting tensors to matrices)**.** Take for example the first tensor from the list above:

```
sage: t=tens[0]
# This is how we printed the tensors until now:
sage: t.display()
f_0*f_0 - f_1*f_1 + f_1*f_2 + f_2*f_1 - f_2*f_2
# .components() gives the matrix of the bilinear form.
# '[:]' is Python syntax for displaying the entire range.
sage: t.components()[:]
[ 1  0  0  0  0  0  0]
[ 0 -1  1  0  0  0  0]
[ 0  1 -1  0  0  0  0]
[ 0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0]
```

△

All the tensors in the list returned by `tensors_identified_with_zero(grouppresentation)` are symmetric tensors, i.e. they lie in the submodule $\Gamma(\pi_2 K) \subseteq \pi_2 K \otimes \pi_2 K$. Obviously these symmetric tensors correspond to the symmetric bilinear forms, which in turn are represented by symmetric matrices.

To describe the surjection we now fix a basis for the module

$$\Gamma(\pi_2 K) \cong \{A \in \mathrm{Mat}(\mathrm{rk}(\pi_2 K) \times \mathrm{rk}(\pi_2 K), \mathbb{Z}) \mid A \text{ symmetric}\}.$$

Let $E_{kl} := (\delta_{ik}\delta_{lj})_{ij}$ be the matrix with a single 1 in the kth row and lth column. Then one possible basis for the space of symmetric matrices is

$$\{E_{kk} \mid 1 \leq k \leq \mathrm{rk}(\pi_2 K)\} \cup \{E_{kl} + E_{lk} \mid 1 \leq k < l \leq \mathrm{rk}(\pi_2 K)\}. \tag{32}$$

Example 3.13. Here $\mathrm{rk}(\pi_2 K) = 7$, so we have $\mathrm{rk}(\Gamma(\pi_2 K)) = \frac{7 \cdot 8}{2} = 28$. The basis matrices in equation 32 look like this:

```
sage: basis_for_symmetric_matrices(7)
[
[1 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 1 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 1 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0], [0 0 0 0 0 0 0], [0 0 0 0 0 0 0],
# ... (more matrices E_kk with a single '1' on the diagonal)
[0 0 0 0 0 0 0]  [0 1 0 0 0 0 0]  [0 0 1 0 0 0 0]
[0 0 0 0 0 0 0]  [1 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [1 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 1], [0 0 0 0 0 0 0], [0 0 0 0 0 0 0],
# ... (more matrices of the form E_kl + E_lk)
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 0]
[0 0 0 0 0 1 0]  [0 0 0 0 0 0 1]  [0 0 0 0 0 0 0]
[0 0 0 0 1 0 0]  [0 0 0 0 0 0 0]  [0 0 0 0 0 0 1]
[0 0 0 0 0 0 0], [0 0 0 0 1 0 0], [0 0 0 0 0 1 0]
]
```

△

For the next step we write the symmetric matrices as (long) lists in *Sage*, i.e. we use the isomorphism $\mathrm{Mat}(k \times k, R) \cong R^{k \cdot k}$. From these vectors we can obtain the submodule generated by the symmetric matrices.

Example 3.14. `subspace_of_symmetric_matrices(n)` calls the *Sage* function `span(basis_vectors, ZZ)` which returns the $\mathbb{Z}$-module generated by the elements in `basis_vectors` (these are the matrices in 32, written as lists). As these are linearly independent we get a free module.

Remark 3.15 (Warning on the order of basis vectors). When generating a submodule from vectors in *Sage* the order of the basis vectors might change, i.e. the basis of the new module might not be in the same order as the generating set given as the function argument. But that does not trouble us: Permuting the vectors in a basis is just a module automorphism and does not change its structure.

```
sage: Sym=subspace_of_symmetric_matrices(7)
sage: Sym
Free module of degree 49 and rank 28 over Integer Ring
Echelon basis matrix:
28 x 49 dense matrix over Integer Ring
# For a large matrix the user has to call the print command explicitly
sage: print Sym.matrix().str()
[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
# ... (the entire list contains 28 basis vectors)
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
```

△

Now we are ready to discuss the `matrix_of_surjection(grouppresentation)`-function:

It takes the list of tensors identified with zero and the subspace of symmetric $(\mathrm{rk}(\pi_2 K) \times \mathrm{rk}(\pi_2 K))$-matrices and writes each tensor as linear combination of the matrices. The coefficients needed in the linear combination for each tensor are written as the columns of a new matrix A, and the latter is returned. The matrix A describes a linear map from a free $\mathbb{Z}$-module to $\Gamma(\pi_2(K))$ which maps each basis vector to one of the elements in `tensors_identified_with_zero(grouppresentation)`.

In the cokernel of this map the name `tensors_identified_with_zero` says it all: Each of the tensors in the list is zero in the quotient $\mathrm{Coker}(A) = \Gamma(\pi_2(K))/\operatorname{Im}(A)$.

Part 3 - Classification of the matrix:

The last step is the classification of a finitely generated abelian group, here the cokernel of the matrix A returned by the function `matrix_of_surjection(grouppresentation)`.

The elementary divisors of the matrix (defined as the diagonal entries in the Smith normal form, see proposition 2.33) describe the cokernel completely. If we are just interested in the existence or absence of torsion (i.e. we do not need to calculate an actual torsion element) we can forget the base change information from the Smith algorithm and thus save time and memory.

There are a number of implementations of algorithms already available, in the program we just use one of them. Here is a short list with some of the methods we could use:

- The *Sage*-function `A.smith_form()` returns a 3-tuple (S, U, V) of matrices such that $S = U \cdot A \cdot V$. Here S is a diagonal matrix in Smith normal form.

- The *Sage*-function `A.elementary_divisors()` returns a list of the invariants of the cokernel of left multiplication with this matrix. These are exactly the diagonal entries of the matrix S we get from a call to `A.smith_form()`.

- *Sage* can convert the dense matrix (i.e. every entry of the matrix is written in some cell of computer memory) to a sparse matrix (i.e. only the nonzero entries with their positions are saved). For this just call `A.sparse_matrix()`.
  The functions `.smith_form()` and `.elementary_divisors()` work like above. Applied to a matrix with a lot of zero-entries the latter algorithm can be more efficient on the sparse type.

- We can use the implementation in GAP: The `libgap` package provides an interface with all the capabilities of GAP. There the function `SmithNormalFormIntegerMat` is available.
  The program currently uses this implementation because experimentation has shown that the algorithm in GAP performs best.

Example 3.16. Let us conclude this section by wrapping up the example: At first use the function to generate the matrix. The result is very large, so we just show the first two columns.

```
sage: AA=matrix_of_surjection_use_just_generators(G)
sage: AA
28 x 56 dense matrix over Rational Field
(use the '.str()' method to see the entries)
# Do not get confused by the 'Rational Field',
# all the entries of the matrix are integers.
sage: AA.columns()[0]
(1, 0, 0, 0, 0, 0, 0, -1, 1, 0, 0, 0, 0, -1, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
sage: AA.columns()[1]
(1, 0, 0, 0, 0, 0, 0, -1, 1, 0, 0, 0, 1, -1, 0, 0, 0, -1,
    0, 0, 0, 0, 0, 0, 0, 0, 0, -1)
```

Now let us classify the matrix. In this example we just use the default implementation in *Sage*.

```
sage: (S, U, V) = AA.smith_form()
# 'S' is the diagonal matrix in Smith form,
# 'U' and 'V' are the associated base changes.
# We are only interested in the diagonal
# of 'S' as all the other entries are zero.
sage: diagonal(S)
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0]
# Observe that this gives the same as the
# function .elementary_divisors()
sage: AA.elementary_divisors()
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0]
sage: diagonal(S)==AA.elementary_divisors()
True
```

group presentation K	$\cong_{(Grp)}$	$\mathrm{rk}_{\mathbb{Z}}(\pi_2 K)$	$\Gamma(\pi_2 K)/\pi_1 K \cong_{(Ab)}$
$\langle a \mid a^2 \rangle$	$\mathbb{Z}/(2)$	1	$\mathbb{Z}$
$\langle a \mid a^3 \rangle$	$\mathbb{Z}/(3)$	2	$\mathbb{Z}$
$\langle a \mid a^4 \rangle$	$\mathbb{Z}/(4)$	3	$\mathbb{Z}^{\oplus 2}$
$\langle a \mid a^5 \rangle$	$\mathbb{Z}/(5)$	4	$\mathbb{Z}^{\oplus 2}$
$\langle a \mid a^6 \rangle$	$\mathbb{Z}/(6)$	5	$\mathbb{Z}^{\oplus 3}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 1: Presentations of cyclic groups

Now we can read off the end result: As the invariant factors are all equal to 0 or 1 the quotient does not contain torsion. From the output we actually get a complete description:

$$\begin{aligned} K &= \langle a, b \mid a^2, b^2, aba^{-1}b^{-1} \rangle \\ \Gamma(\pi_2 K)/\pi_1 K &\cong_{(Ab)} \left(\bigoplus_{i=1}^{18} \underbrace{\mathbb{Z}/(1)}_{=0} \right) \oplus \left(\bigoplus_{i=1}^{10} \underbrace{\mathbb{Z}/(0)}_{=\mathbb{Z}} \right) \\ &\cong \mathbb{Z}^{\oplus 10} \end{aligned}$$

Thus the quotient is free and $\mathrm{Tors}(\Gamma(\pi_2 K)/\pi_1 K) = 0$. $\triangle$

4 Results and conclusion

In this last section we give the program's results for some concrete group presentations. Finally we reflect on the methods used and put this work into context.

Surprisingly we only found one group presentations K yet which has torsion in $\Gamma(\pi_2 K)/\pi_1 K$: It is the complex $\langle a, b, c \mid a^2, b^2, c^4, [a, b], [a, c], [b, c] \rangle$ associated to the group $\mathbb{Z}/(2) \times \mathbb{Z}/(2) \times \mathbb{Z}/(4)$.

4.1 Some example outputs

4.1.1 Finite cyclic groups

Remember what we proved in example 2.37:

For the presentation $K = \langle a \mid a^n \rangle$ of a cyclic group, $\mathrm{Tors}\,\Gamma(\pi_2 K)/\pi_1 K$ is trivial. The program's results are consistent with this, at least for the cases tested (cyclic groups of order ≤ 20). See the subsection 4.2 for a comment on the performance of the program. In table 1 we show some of the results for this type of group presentation.

group presentation K	$\cong_{(Grp)}$	$\mathrm{rk}_{\mathbb{Z}}(\pi_2 K)$	$\Gamma(\pi_2 K)/\pi_1 K \cong_{(Ab)}$
$\langle a, b \mid a^2, b^2, [a,b]\rangle$	$(\mathbb{Z}/(2))^2$	7	$\mathbb{Z}^{\oplus 10}$
$\langle a, b \mid a^2, b^3, [a,b]\rangle$	$\mathbb{Z}/(2) \times \mathbb{Z}/(3)$	11	$\mathbb{Z}^{\oplus 12}$
$\langle a, b \mid a^2, b^4, [a,b]\rangle$	$\mathbb{Z}/(2) \times \mathbb{Z}/(4)$	15	$\mathbb{Z}^{\oplus 18}$
$\langle a, b \mid a^2, b^5, [a,b]\rangle$	$\mathbb{Z}/(2) \times \mathbb{Z}/(5)$	19	$\mathbb{Z}^{\oplus 20}$
$\langle a, b \mid a^3, b^3, [a,b]\rangle$	$(\mathbb{Z}/(3))^2$	17	$\mathbb{Z}^{\oplus 17}$
$\langle a, b \mid a^3, b^4, [a,b]\rangle$	$\mathbb{Z}/(3) \times \mathbb{Z}/(4)$	23	$\mathbb{Z}^{\oplus 24}$
$\langle a, b \mid a^4, b^4, [a,b]\rangle$	$(\mathbb{Z}/(4))^2$	31	$\mathbb{Z}^{\oplus 34}$
$\langle a, b, c \mid a^2, b^2, c^2, [a,b], [a,c], [b,c]\rangle$	$(\mathbb{Z}/(2))^3$	31	$\mathbb{Z}^{\oplus 76}$
$\langle a, b, c \mid a^2, b^2, c^3, [a,b], [a,c], [b,c]\rangle$	$(\mathbb{Z}/(2))^2 \times \mathbb{Z}/(3)$	47	$\mathbb{Z}^{\oplus 100}$
$\langle a, b, c \mid a^2, b^2, c^4, [a,b], [a,c], [b,c]\rangle$	$(\mathbb{Z}/(2))^2 \times \mathbb{Z}/(4)$	63	$\mathbb{Z}^{\oplus 140} \oplus (\mathbb{Z}/(2))^{\oplus 2}$

Table 2: Presentations of products of cyclic groups

group presentation K	$\cong_{(Grp)}$	$\mathrm{rk}_{\mathbb{Z}}(\pi_2 K)$	$\Gamma(\pi_2 K)/\pi_1 K \cong_{(Ab)}$
$\langle s, d \mid s^2, d^3, (sd)^2\rangle$	D_3	11	$\mathbb{Z}^{\oplus 14}$
$\langle s, d \mid s^2, d^4, (sd)^2\rangle$	D_4	15	$\mathbb{Z}^{\oplus 20}$
$\langle s, d \mid s^2, d^5, (sd)^2\rangle$	D_5	19	$\mathbb{Z}^{\oplus 24}$
$\langle s, d \mid s^2, d^6, (sd)^2\rangle$	D_6	23	$\mathbb{Z}^{\oplus 30}$
$\langle s, d \mid s^2, d^7, (sd)^2\rangle$	D_7	27	$\mathbb{Z}^{\oplus 34}$
$\langle s, d \mid s^2, d^8, (sd)^2\rangle$	D_8	31	$\mathbb{Z}^{\oplus 40}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 3: Presentations of dihedral groups

4.1.2 Products of finite cyclic groups

Here we look at groups of the form $\mathbb{Z}/(n_1) \times \mathbb{Z}/(n_2) \times \ldots \times \mathbb{Z}/(n_k)$. In the presentations the commutators $[g,h] := ghg^{-1}h^{-1}$ of the generators are used to write the relations. Because of the high number of relations the matrices involved in the calculation are very large. Since the memory in our computer is limited we can only test some small examples. The results are summarized in table 2.

4.1.3 Dihedral groups

The dihedral groups D_n of order $2n$ are examples of nonabelian groups. Here we use the presentation $D_n \cong_{(Grp)} \langle s, d \mid s^2, d^n, (sd)^2\rangle$, the generator s corresponds to the reflection, d to the rotation by $\frac{2\pi}{n}$. The results can be seen in table 3, we tested for $n \leq 8$ and did not find torsion.

4.2 Performance

One could speed up the program a lot by implementing the CPU-intensive parts in a more efficient language, for example in C. Furthermore, the program does not use any form of multithreading yet. Some of the algorithms, e.g. the calculation of the tensors identified with zero, could be done in parallel because they do not depend on any other results.

The matrices involved in the calculation grow in size considerably as the input group presentations are more complex. A problem is the quadratic growth in the rank of $\Gamma(M)$. In the last step we have to deal with a matrix of dimensions $\mathrm{rk}_{\mathbb{Z}}(\Gamma(\pi_2 K)) \times (\#(\text{gen. of } \pi) \cdot \mathrm{rk}_{\mathbb{Z}}(\Gamma(\pi_2 K)))$. To operate on this matrix *Sage* has to allocate a lot of memory.

Example 4.1. This gets out of hand even for simple presentations like $K = \langle a, b, c \mid a^2, b^2, c^4, [a, b], [a, c], [b, c]\rangle \cong_{(Grp)} \mathbb{Z}/(2) \times \mathbb{Z}/(2) \times \mathbb{Z}/(4)$. Here we have $\mathrm{rk}_{\mathbb{Z}}(\pi_2 K) = 63$ and this implies $\mathrm{rk}_{\mathbb{Z}}(\Gamma(\pi_2 K)) = \frac{63 \cdot 64}{2} = 2016$ for the Γ-construction. Then the program has to calculate all the entries of a 2016×6048-matrix and handle it further afterwards (e.g. apply the Smith algorithm). $\triangle$

4.3 Conclusion

In the beginning most of my time was spent on planning the rough outline of the algorithm and finding the *Sage*-packages suitable for the job. This involved getting all the library functions to work and interpreting the output of *Sage* correctly. But after the initial challenges were overcome I had time to focus on the actual mathematics behind the topic. In the end I both got a better understanding of the topology contained in this question as well as had to think about formulating the sometimes abstract constructions concretely in the computer language.

This thesis directly leads to a lot of questions where no answer is known yet, here is a small list just to get started:

- What are more examples of group presentations K where there **is** torsion in $\Gamma(\pi_2 K)/\pi_1 K$?
- Can we prove for more types of presentations that there never is torsion (like we did in example 2.37)?
- How could the calculation of $\mathrm{Tors}(\Gamma(\pi_2 K)/\pi_1 K)$ be done more easily/efficiently? (Implementation in another language?)
- Which other tedious calculations in the classification of manifolds could be automated?

References

[Boo58] William W Boone. The word problem. *Proceedings of the National Academy of Sciences*, 44(10):1061–1065, 1958.

[Bro12] K.S. Brown. *Cohomology of Groups*. Graduate Texts in Mathematics. Springer New York, 2012.

[Dev16] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 7.1)*, 2016. `http://www.sagemath.org`.

[Fen83] R. Fenn. *Techniques of Geometric Topology*. British and American Playwrights, 1750-1920. Cambridge University Press, 1983.

[Fox53] Ralph H Fox. Free differential calculus. i: Derivation in the free group ring. *Annals of Mathematics*, pages 547–560, 1953.

[GAP16] The GAP Group. *GAP – Groups, Algorithms, and Programming (Version 4.7.9)*, 2016. `http://www.gap-system.org`.

[Hat02] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[HH70] Brian Hartley and Trevor O Hawkes. *Rings, modules and linear algebra: a further course in algebra describing the structure of Abelian groups and canonical forms of matrices through the study of rings and modules*. Chapman & Hall/CRC, 1970.

[HK88] Ian Hambleton and Matthias Kreck. On the classification of topological 4-manifolds with finite fundamental group. *Mathematische Annalen*, 280(1):85–104, 1988.

[Neu82] J. Neubüser. An elementary introduction to coset table methods in computational group theory. *Groups-St Andrews 1981*, 71:1, 1982.

[Sti12] John Stillwell. *Classical topology and combinatorial group theory*, volume 72. Springer Science & Business Media, 2012.

[Whi50] John HC Whitehead. A certain exact sequence. *Annals of Mathematics*, pages 51–110, 1950.