Computational Application of the Transfer Algorithm to Borromean Rings

A Senior Thesis Submitted to the Department of Mathematics In Partial Fulfillment of the Requirements for the Departmental Honors Baccalaureate

By Mervin Fansler

Millersville, Pennsylvania

June 2016

This Senior Thesis was completed in the Department of Mathematics, defended before and approved by the following members of the Thesis committee

---

Ronald Umble, Ph.D (Thesis Advisor)
Professor of Mathematics
Millersille University

Bruce Ikenaga, Ph.D
Associate Professor of Mathematics
Millersville University

Michael McCooey, Ph.D
Associate Professor of Mathematics and Department Chair
Franklin & Marshall College

James D. Stasheff, Ph.D, D. Phil
Professor of Mathematics Emeritus
University of North Carolina at Chapel Hill

Abstract

This thesis presents the application of a software implementation of an algorithm which induces an A_∞-coalgebra on the homology of an A_∞-coalgebra. A description of the algorithm and an exposition of the core algorithmic strategies employed in the implementation are given. After demonstrating the use of the software on a set of simple examples, results are presented from applying it to a chain complex on a cellular decomposition of the link complement of Borromean rings.

Acknowledgments

This project was conceived by Ron Umble and depended in large part on his frequent input and feedback throughout the process. I am extremely indebted to him for his ceaseless enthusiasm, not only in this project, but for more broadly shaping my appreciation of research. I would like to thank the thesis committee for their constructive criticism and insightful suggestions.

This research was in part supported by a Sage Days 65 grant.

Contents

Chapter 1

Introduction

Simple homology fails to capture linkage information in a chain complex or differential graded algebra. A classic approach to algebraically computing such linkage information is the use of Massey products, which has the power to distinguish n-fold linkages, such as are encountered in complements of n-component Brunnian links. An alternative approach has been proposed which uses a diagonal approximation defined on cellular chains to induce an A_∞-coalgebra on its homology [10]. It has been conjectured that features of the induced A_∞-coalgebra will similarly detect n-fold linkages in Brunnian links, as well as yield additional useful information about the underlying cellular structure [9].

The present work is the outcome of attempting to evaluate some of the initial cases involved in the conjecture. Due to the demanding calculations used in the transfer algorithm, a series of computational tools were developed with the hope to automate the application of the algorithm. Here we present the results of those tools as applied to the complement of the Borromean rings, a 3-component Brunnian link.

In this chapter, general background and basic definitions are introduced. Chapter two describes the procedure used to transfer the coproduct on chains to a coproducts on homology. Chapter three presents the computational implementations of algorithms used in the procedure. Chapter four illustrates the computations of the implementation using example data sets, including an unlink, a Hopf link, a coassociative differential graded coalgebra (DGC), and non-coassociative DGC. Chapter five presents the results of applying the transfer algorithm to a cellular decomposition of the complement of the Borromean rings.

Readers concerned primarily with the results and the algorithm used to obtain them can omit chapter three, and need only skim chapter four. Those focused on the computational underpinnings can skim chapter two in order to familiarize themselves with the symbols used and the arch of the algorithm, read chapter three thoroughly, and use chapter four to verify their understanding of the output.

1.1 Terms

Definition 1 *A* ***cellular decomposition*** *of a topological space X consists of a finite collection of k-cells, $k = 0, 1, 2, \ldots$, such that*

- *the boundary of a k-cell is a union of $(k-1)$-cells*
- *the non-empty intersection of k-cells is a k-cell*
- *the union of all cells is X.*

We often refer to 0-cells as vertices, 1-cells as edges, 2-cells as faces, and so forth.

Definition 2 *A* ***chain complex*** *(C_*, ∂) is graded set of $\mathbb{Z}_2$-vector spaces, $\{C_0, C_1, C_2, \ldots, C_k, \ldots\}$, with k-cells of a cellular decomposition as a basis for C_k, connected by a boundary operator $\partial : C_n \to C_{n-1}$, such that $\partial\partial = 0$. Elements of the C_i's are referred to as* ***chains****.*

Definition 3 *A chain x is a* ***cycle*** *if $\partial x = 0$. Given a chain complex C, the set of all cycles in C will be denoted by $Z(C) \subset C$.*

Definition 4 *Let x be a cycle. A* ***preboundary*** *of x is a chain y such that $\partial y = x$.*

Definition 5 *A cycle c is a* ***boundary*** *or* ***bounding cycle*** *if it has a preboundary; otherwise, it is a* ***non-bounding cycle****.*

Definition 6 *A k-cycle c is **simple** if a $(k-2)$-cell in ∂c is shared by exactly two $(k-1)$-cells in ∂c.*

Definition 7 *An **incidence matrix** is a matrix with rows and columns corresponding to cells, and whose $(i,j)^{th}$ entry is a non-negative integer equal to the number of times the cell i is present in the boundary of cell j.*

Definition 8 *A map $\Delta : X \to X \times X$ is a diagonal approximation if*

- *Δ is homotopic to the geometric diagonal $x \mapsto (x,x)$*
- *$\Delta(c)$ is a subcomplex of $c \times c$*
- *∂ is a coderivation of Δ, i.e., $\Delta\partial = (\partial \times \mathrm{Id} + \mathrm{Id} \times \partial)\Delta$.*

Definition 9 *Let (A, ∂_A) and (B, ∂_B) be chain complexes, and let $\xi \in \mathrm{Hom}(A,B)$. The **differential operator** $\nabla : Hom_k(A,B) \to Hom_{k-1}(A,B)$ is defined by $\nabla(\xi) = \partial_B \xi + \xi \partial_A$.*

Definition 10 *An A_∞-**colalgebra** is a chain complex (A, ∂) together with a family of multilinear operations $\{\Delta_n : A \to A^{\otimes n}\}_{n \geq 2}$ such that*

$$\nabla(\Delta_n) = \sum_{i+j=n+1, 0 \leq k \leq n-j} (1^{\otimes k} \otimes \Delta_j \otimes 1^{\otimes n-j-k})\Delta_i$$

for each n.

The relations of an A_∞-coalgebra can be represented by the associahedra $\{K_n\}_{n\geq 2}$, a sequence of $(n-2)$-dimensional polytopes [6]. The top dimension of each K_n polytope corresponds to the multilinear operator Δ_n; each codimension 1 cell corresponds to a component in $\nabla(\Delta_n)$.

Definition 11 *Let $(A, \partial_A, \Delta^2, \Delta^3, \ldots)$ and $(B, \partial_B, \Delta_2, \Delta_3, \ldots)$ be A_∞-colagebras and let TB denote the vector space of k-fold tensors, i.e., $TB = \mathbb{Z}_2 \oplus B \oplus B \otimes B \oplus \cdots$.*

*A map $G : A \to TB$ is an A_∞-**colagebra map** if there exist a family of maps $g^n : A \to B^{\otimes n}{}_{n\geq 1}$ such that $G = \sum_{n\geq 1} g^n$ and*

$$\nabla(g^n) = \sum_{i+j=n+1, 0\leq k\leq n-j} (1^{\otimes k} \otimes \Delta_j \otimes 1^{\otimes n-j-k}) g^i + \sum_{i_1+\cdots+i_k=n, 2\leq k\leq n} (g^{i_1} \otimes \cdots \otimes g^{i_k}) \Delta_k$$

for each n.

The relations of an A_∞-coalgebra map can be represented by the multiplihedra $\{J_n\}_{n\geq 1}$, a sequence of $(n-1)$-dimensional polytopes. The top dimension of each J_n polytope corresponds to the map g^n; each codimension 1 cell corresponds to a component in $\nabla(g^n)$.

Chapter 2

Transfer Algorithm

The transfer algorithm takes as input an A_∞-coalgebra $(C, \partial, \Delta_2, \Delta_3, \ldots)$ and a cycle-selecting map, $g : H \to Z(C)$, from $H = H_*(C, \partial)$ to representative non-bounding cycles in C. In the initialization step, a coproduct in homology, $\Delta^2 : H \to H \otimes H$ is induced, followed by determination of the A_∞-coalgebra map component $g^2 : H \to C \otimes C$.

Once these are obtained, an inductive step can be carried out, which uses the products $\Delta_2, \Delta_3, ..., \Delta_{n+1}; \Delta^2, \Delta^3, ..., \Delta^n$ with the map components $g^2, g^3, ..., g^n$ to induce Δ^{n+1} and g^{n+1}.

2.1 Formal Description

The following is due to Umble [10] and is a special case of a general transfer algorithm due to Sandeblidze and Umble [4].

The Transfer Algorithm

Let K_n denote the $(n-2)$-dimensional associahedron whose faces are indexed by down-rooted planar trees with n leaves. Let $K = \sqcup_{n\geq 2} K_n$ and let $C_*(K)$ denote the cellular chains of K, i.e., the $\mathbb{Z}_2$-vector space generated by the cells of K. Let θ^n denote the n-leaf corolla indexing the top dimensional cell of K_n. Let J_n denote the

$(n-1)$-dimensional multiplihedron whose $(n-2)$-dimensional faces are indexed by down-rooted 2-colored planar trees with n-leaves (see Figure 2). Let $J = \sqcup_{n\geq 1} J_n$, and let $C_*(J)$ denote the cellular chains of J. Let $\mathfrak{f}^n$ denote the n-leaf (red) corolla indexing the top dimensional cell of J_n (see Figure 1).

Initial data

- An A_∞-coalgebra $(C, \partial, \Delta_2, \Delta_3, \ldots)$ over $\mathbb{Z}_2$
- An injective (cycle-selecting) map $g : H \to Z$, where Z denotes the subspace of cycles in C.

Objectives

For each $n \geq 2$

- Define an operation $\Delta^n : H \to H^{\otimes n}$
- Construct a chain map $\alpha : C_*(K) \to Hom\left(H, H^{\otimes n}\right)$ defined on generators by $\alpha\left(\theta^n\right) := \Delta^n$
- Construct a map $G : H \Rightarrow C$ of A_∞-coalgebras, where

$$G = \left\{g^n \in Hom\left(H, C^{\otimes n}\right) \mid g^1 = g\right\}$$

Facts

- $\nabla \equiv 0$ on $Hom\left(H, H^{\otimes n}\right)$ implies $H^*\left(Hom\left(H, H^{\otimes n}\right)\right) = Hom\left(H, H^{\otimes n}\right)$
- The induced map $\tilde{g}_* : Hom\left(H, H^{\otimes n}\right) \to H^*\left(Hom\left(H, C^{\otimes n}\right)\right)$ given by $\tilde{g}_*(u) = \left[g^{\otimes n} u\right]$ is an isomorphism since g is a homology isomorphism
- The map $f : Z \to H$ defined by $f(z) = [z]$ is a homotopy inverse of g, i.e., $fg = \mathrm{Id}_H$ and $gf - \mathrm{Id}_Z = \nabla s$ for some $s \in Hom_1(C, C)$; thus $[gf] = [\mathrm{Id}_Z]$ and $[fg] = [\mathrm{Id}_H]$

- The induced map $\tilde{f}_* : H^*\left(Hom\left(H, C^{\otimes n}\right)\right) \to Hom\left(H, H^{\otimes n}\right)$ given by $\tilde{f}_*\left([u]\right) = f^{\otimes n}u$ is an isomorphism

- $\tilde{g}_*^{-1} = \tilde{f}_*$

Initialization:

1. Define β on $J_1 : \mathfrak{f}^1 = |$ $\mapsto g$

2. Use the composition $\Delta_2 g$ to extend β to one vertex of J_2 :

$$\theta^2\mathfrak{f}^1 =$$  $$\mapsto \Delta_2 g$$

3. Consider $\tilde{g} : Hom\left(H, H\right) \to Hom\left(H, C\right)$ and the ∇-cocycle $\Delta_2 g \in Hom_0\left(H, C \otimes C\right)$

   a. Define $\Delta^2 := \tilde{f}_*\left([\Delta_2 g]\right)$; then

   $$\tilde{g}_*\left(\Delta^2\right) = [\Delta_2 g]$$

   b. Define $\alpha\left(\theta^2\right) = \Delta^2$ and $\beta\left(\left(\mathfrak{f}^1 \otimes \mathfrak{f}^1\right)\theta^2\right) = \left(g \otimes g\right)\Delta^2$

   c. Note that $[\Delta_2 g - \left(g \otimes g\right)\Delta_2] = 0$

   d. If $\Delta_2 g = \left(g \otimes g\right)\Delta^2$, set $g^2 = 0$

   e. If $\Delta_2 g \neq \left(g \otimes g\right)\Delta^2$, choose $g^2 \in Hom_1\left(H, C \otimes C\right)$ such that

   $$\nabla g^2 = \Delta_2 g - \left(g \otimes g\right)\Delta^2$$

   f. In either case, define $\beta\left(\mathfrak{f}^2\right) = g^2$

 $\mapsto g^2$

Figure 1.

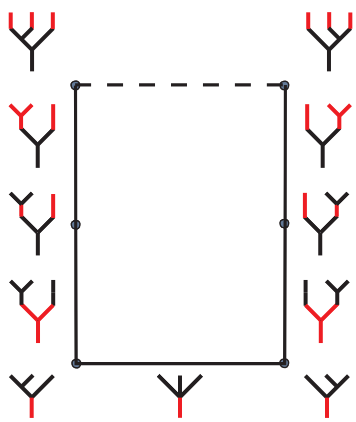
4. Use Δ^2 and compositions $(\Delta^2 \otimes \mathbf{1})\,\Delta^2$ and $(\mathbf{1} \otimes \Delta_2)\,\Delta^2$ to define α on

- K_2 :  $\mapsto \Delta^2$
- ∂K_3 :  $\mapsto (\Delta^2 \otimes \mathbf{1})\,\Delta^2$ and  $\mapsto (\mathbf{1} \otimes \Delta^2)\,\Delta^2$
- Since $\nabla \equiv 0$ on $Hom\,(H, H^{\otimes 3})$, the Transfer Theorem implies

$$\left(\Delta^2 \otimes \mathbf{1}\right)\Delta^2 = \left(\mathbf{1} \otimes \Delta^2\right)\Delta^2$$

5. Use the compositions involving Δ^2, Δ_2, g and g^2 to extend β to

- all of $C_*\,(J_2)$:  $\mapsto (g \otimes g)\,\Delta^2$
- $\partial J_3 \smallsetminus \operatorname{int} K_3 \times 1 = \partial\,(K_3 \times 1)$:



$\mapsto$

$g^{\otimes 3}\,(\Delta^2 \otimes \mathbf{1})\,\Delta^2$	$g^{\otimes 3}\,(\mathbf{1} \otimes \Delta^2)\,\Delta^2$
$(g^2 \otimes g)\,\Delta^2$	$(g \otimes g^2)\,\Delta^2$
$(\Delta_2 g \otimes g)\,\Delta^2$	$(g \otimes \Delta_2 g)\,\Delta^2$
$(\Delta_2 \otimes \mathbf{1})\,g^2$	$(\mathbf{1} \otimes \Delta_2)\,g^2$

$(\Delta_2 \otimes \mathbf{1})\,\Delta_2 g \qquad \Delta_3 g \qquad (\mathbf{1} \otimes \Delta_2)\,\Delta_2 g$

Figure 2.

Induction hypothesis.

Given $n \geq 3$, assume there exist chain maps

- $\alpha : C_*(K_i) \to Hom(H, H^{\otimes i})$ such that $\alpha(\theta^i) = \Delta^i$ for $2 \leq i < n$
- $\alpha : C_*(\partial K_n) \to Hom(H, H^{\otimes n})$
- $\beta : C_*(J_i) \to Hom(H, C^{\otimes i})$ such that $\beta(\mathfrak{f}^i) = g^i$ for $1 \leq i < n$
- $\beta : C_*(\partial J_n \smallsetminus \operatorname{int} K_n \times 1) \to Hom(H, C^{\otimes n})$

Induction objectives:

- Extend α to $\theta^n \in C_{n-2}(K_n)$
- Extend β to $(\mathfrak{f}^1)^{\otimes n}\theta^n \leftrightarrow K_n \times 1 \in C_{n-2}(J_n)$
- Extend β to $\mathfrak{f}^n \in C_{n-1}(J_n)$

Induction:

1. Define $\varphi_n := \beta(\partial J_n \smallsetminus \operatorname{int} K_n \times 1)$
2. $\nabla\varphi_n = \nabla\beta(\partial J_n \smallsetminus \operatorname{int} K_n \times 1) = \beta\partial(K_n \times 1) = 0$
3. $[\varphi_n] \in H^*(Hom(H, C^{\otimes n}))$
4. Define $\Delta^n := \tilde{f}_*([\varphi_n])$
5. $\tilde{g}_*(\Delta^n) = [\varphi_n]$ implies $[g^{\otimes n}\Delta^n - \varphi_n] = 0$
6. If $g^{\otimes n}\Delta^n = \varphi_n$, define $g^n = 0$
7. If $g^{\otimes n}\Delta^n \neq \varphi_n$, choose $g^n \in Hom_{n-1}(H, C^{\otimes n})$ such that $\nabla g^n = g^{\otimes n}\Delta^n - \varphi_n$
8. Define $\alpha(\theta^n) := \Delta^n$, $\beta(\mathfrak{f}^n) := g^n$, and $\beta\left((\mathfrak{f}^1)^{\otimes n}\theta^n\right) := g^{\otimes n}\Delta^n$
9. Then $\alpha\partial(\theta^n) = \nabla\alpha(\theta^n) = 0$

This completes the induction.

Chapter 3

Computational Implementation

All computational tools for this project were developed using the Python programming language. Throughout this section, however, we will attempt to describe the algorithms in generic terms, with explanations of any idiosyncrasies that are particular to the language choice.

The specific version of all referenced code is available at `https://github.com/mfansler/CellularChainParser/tree/v0.1.0`.

3.1 Core Data Structures

3.1.1 Cell

A simple cell of a chain complex was stored as a string identifier. For example, the vertex v_0 would be stored as the string literal `"v_{0}"`. The cells of a tensor product of chain complexes was stored as a tuple of strings. For example, the three dimensional cell $a \otimes ab \otimes v \in C_1 \otimes C_2 \otimes C_0$ would have the representation `("a", "ab", "v")`.

3.1.2 Chain

Chains were represented in two interconvertible forms, an *expanded* form and a *factored* form. In the expanded form, a chain is simply a list of cells. In factored form, a chain is represented as a list of tuples, where the tuple components are lists of simple cells. The following table illustrates some example representations:

Chain	**Expanded Form**	**Factored Form**
v	`["v"]`	`[(["v"], )]`
$m_1 + m_2 + m_3$	`["m_{1}", "m_{2}", "m_{3}"]`	`[(["m_{1}"], ["m_{2}"], ["m_{3}"])]`
$a \otimes a + a \otimes b + b \otimes a + b \otimes b$	`[("a", "a"), ("a", "b"), ("b", "a"), ("b", "b")]`	`[(["a", "b"], ["a", "b"])]`

3.1.3 Maps

Maps are represented using Python's `dict` object, which acts as a simple key-value store data structure where the keys represent the domain and the values the codomain.

3.1.4 Coalgebra

The `Coalgebra` class implemented in **Coalgebra.py** consists of three components:

1. a graded set of simple cells represented as a `dict` with integer dimension as keys and lists of simple cells as values;
2. a boundary map represented as a map from simple cells to chains; and
3. a coproduct map represented as a map from simple cells to chains of tuple cells.

Technically, this data structure represents a differential graded coalgebra.

3.2 Utilities

3.2.1 File Input and Parsing

A chain complex, together with an associated coproduct (diagonal) definition is stored as a `.tex` file. The data in the file is given in LaTeX syntax. To convert the raw file to a `Coalgebra` data structure, an LALR parser which recognizes a subset of LaTeX was developed using the Python library **PLY (Python Lex-Yacc)** (see **CellChainLex.py** and **CellChainParse.py**). A list of lexicographic tokens and a formal grammar are provided in Appendix A.

3.2.2 Validating Diagonal Definition

To verify that the provided coproduct in an input file is a coderivation with respect to the chain complex, the script **validateCoproduct.py** was developed. For each cell in the graded set of the `Coalgebra`, two chain maps were computed:

1. $\Delta_2 \partial$

2. $(\partial \otimes 1 + 1 \otimes \partial)\Delta_2$

and then tallies of the cells appearing in resulting chains were compared. If tallies were identical for all cells then the coproduct was confirmed as a coderivation; were any tallies inconsistent, the cell and the discrepancy was explicitly identified in the script output.

3.2.3 Homology Groups

Multiple means of obtaining homology groups for a given differential graded coalgebra were provided:

- a direct interface to CHomP within the transfer script;

- a script to output incidence matrices for CHomP; and

- a script to output code for SageMath `ChainComplex` object.

In all cases, the correctness of the homology groups ultimately relies on the correctness of CHomP, since SageMath itself uses CHomP to compute homology.

Interface to CHomP

The generic transfer script, **transferPart.py**, lines 79-148, begins from a parsed `Coalgebra` object, generates an incidence matrix, outputs the matrix to a temporary file, and then launches a CHomP instance. The output of the CHomP homology computation is captured and parsed into a set of generators for each homology group.

The **chain2matrix.py** script provides an option to directly output a CHomP-compatible incidence matrix to a specified output file.

SageMath `ChainComplex` Conversion

An alternate option available in the **chain2matrix.py** script is to output SageMath code to create an instance of a `ChainComplex` object corresponding to the chain complex specified in an input file. The output code is of the form:

```
d1 = matrix(Integers(2), <# dim 0 cells>, <# dim 1 cells>, {(i, j):  1, ...}, sparse=True)
d2 = matrix(Integers(2), <# dim 1 cells>, <# dim 1 cells>, {(i, j):  1, ...}, sparse=True)
...
d<n> = matrix(Integers(2), <# dim (n-1) cells>, <# dim n cells>, {(i, j):  1, ...}, sparse=True)
ChainComplex({ 1:  d1, 2:  d2, ..., n:  dn }, degree=-1)
```

Once represented as a SageMath object, the class method `homology()` can be used to obtain homology groups, as well as generators.

Explicit Specification of Homology Groups

The generic transfer script provides a commandline option to specify homology groups and the generators that will be used for the cycle-selecting map. The flag `-hgroups` or `-hg` followed by an input file formatted as a Python `dict` object with homology groups as the keys and cycles as the values, is used to provide a custom homology group.

It should be noted that this code does not test for the compatibility and correctness of the homology groups so specified.

3.3 Key Functions

3.3.1 Cycle to Homology Group Map

The function $f : Z(C) \to H$ is implemented following Algorithm 1. In our implementation, the list of cycle combinations is constructed from $g : H \to Z(C)$ by generating all combinations of elements from the codomain of g. The empty cycle is included in *cycleCombos* to cover the case where z itself is a bounding cycle.

Algorithm 1 $f : Z(C) \to H$ function

---

function $f(z)$
  if $\partial z \neq 0 (\mod 2)$ **then**
    return trivial group
  $cycleCombos \leftarrow$ [list of all combinations of cycles in distinct homology groups]
  for each $cycleCombo \in cycleCombos$ **do**
    $y \leftarrow cycleCombo + z (\mod 2)$
    if $\exists_x \ni y = M \cdot x$, where M is the chain complex incidence matrix **then**
      return homologyGroup($cycleCombo$)
  otherwise, RaiseException()

To test whether a given sum $z + cycleCombo$ is a bounding cycle, the incidence matrix of the chain complex is row reduced and the row reduction operations are stored as a matrix. The vector x is multiplied by the row reducing matrix and the result is checked for any non-zero components in the rows with index exceeding the rank of the incidence matrix. Products are linear combinations of boundary cycles iff no such components are present.

3.3.2 Preboundary Functions

Throughout the transfer process, we frequently encounter the following problems:

Problem 1 *Given a cycle $x \in C^{\otimes n}$ of degree k, find a chain $y \in C^{\otimes n}$ of degree $k+1$, such that $\partial(y) = x$.*

Problem 2 *Given a ∇-cycle $\xi \in Hom_k(A, B)$, find a chain $\gamma \in Hom_{k+1}(A, B)$, such that $\nabla(\gamma) = \xi$.*

In general, a solution can be obtained, when one exists, by using Algorithm 2, which is essentially identical to that presented in [2].

Algorithm 2 General Preboundary Algorithm

---

function $preboundary(x)$
    $k \leftarrow \text{degree}(x)$
    $M \leftarrow$ empty matrix
    $faces \leftarrow$ [list of all degree $(k+1)$ faces]
    for each $face \in faces$ **do**
        append column to M with incidences of degree k cells for $\partial(face)$
    $v \leftarrow \text{Solve}(M \cdot v = x)$
    return [list of $(k+1)$ faces corresponding to solution v]

However, as Kaczynski *et al.* point out [2, p. 140], "[a]lthough the algebraic methods are very general and theoretically correct, in practice they may lead to unacceptably expensive computations."[1] Nevertheless, a general linear approach appeared to be unavoidable in certain cases, and so the strategy indicated in Algorithm 3 was developed for solving linear systems in $\mathbb{Z}_2$ with sparse matrices, following techniques described in [1].

Algorithm 3 Strategy for Solving a Sparse Linear System over $\mathbb{Z}_2$

---

procedure $sparseLinearSolve$(sparse matrix M, column vector y)
    Reorder columns of M by sparsity (low$\rightarrow$high), then by sum of nonzero indices
    Augment M with y
    while M is not in row echelon form (excluding augmented column) **do**
        Apply Gaussian elimination, selecting sparsest row as pivot
        if linearly dependent column is encountered **then**
            set that column to all 0
    Perform Backsubstitution

Such linear methods prove unmanageable as we scale to larger cell complexes and progress further along the inductive steps of the transfer algorithm. As an example, computing the coassociator Δ_3 on chains would require an incidence matrix for the elements in $Hom_1(C, C \otimes C \otimes C)$. For the Borromean rings cellular decomposition used in Chapter 5, which has 11 vertices, 32 edges, 24 faces, and 5 spaces, the num-

[1] In fact, initial attempts at developing the present tools led us to confront such "expensive computations" directly. Despite the sparse nature of the matrices encountered, solving systems by these methods quickly becomes intractable.

ber of elements in $Hom_1(C, C \otimes C \otimes C)$ exceeds 4 million. Row reducing such an incidence matrix is beyond the resources available to us in undertaking this project, and therefore we are forced to resort to other methods.

Algorithm 4 describes a best-first algorithm for searching the space of possible solutions to Problem 2.

Algorithm 4 Best-First Map Integration

```
function bestFirstMapIntegrate(chain map ξ)
    frontier ← set()                                    ▷ initialize empty set
    for each simple component ξ_i in ξ do
        compute every possible map γ, such that ξ_i ∈ ∇(γ)
        frontier.push(γ)                                ▷ add γ to the frontier
    for each γ ∈ frontier do
        Compute ξ' ← ξ − ∇(γ)
        if ∃_γ ∋ ξ' == 0 then
            return γ
    Sort frontier by size(ξ'), smallest first
    while frontier ≠ ∅ do
        temp ← frontier.pop()
        result ← bestFirstMapIntegrate(ξ − ∇(temp))
        if result ≠ FAILFLAG then
            return result + temp
    return FAILFLAG
```

3.3.3 Factorization

In the induction section of the transfer algorithm, steps 3 and 4 require computing the homology groups of $Hom(H, C^{\otimes n})$. Again, while a general linear approach is theoretical robust it can be practically unmanageable, especially in complexes with many elements. Since $\partial H = 0$, the computation essentially reduces to finding $H_*(C^{\otimes n})$. By the Künneth Theorem, we have

$$H_*(C^{\otimes n}, \mathbb{Z}_2) \cong H_*(C, \mathbb{Z}_2)^{\otimes n} \tag{3.1}$$

which tells us that all non-bounding cycles of $Z(C^{\otimes n})$ will be of the form $Z(C)^{\otimes n}$. This allows us to equate computing $[\varphi_n]$ to finding cycles in φ_n of the form $Z(C)^{\otimes n}$, at which point Δ^n can be computed using $f^{\otimes n} : Z(C)^{\otimes n} \to H^{\otimes n}$. This strategy thus gives rise to Problem 3.

Problem 3 (Factorization) *Given a cycle $c \in Z(C^{\otimes n})$, find all subcycles of c of the form $Z(C)^{\otimes n}$.*

Algorithm 5 attempts to solve this problem.

Algorithm 5 Factor Cycles of the Form $Z(C)^{\otimes n}$

```
procedure factorizeCycle(chain xs)
    results ← xs                                          ▷ xs in expanded form
    for each tensor component position i = 1, 2, 3, ...n do
        Partition all chains in results based on equality, ignoring position i
        partitions ← results; results ← []
        for each partition ∈ partitions do
            for each c ∈ partition do
                if ∂c == 0 then
                    move c to results
            cycleLength ← 2
            while size(partition) ≥ cycleLength do
                remaining ← partition
                partition ← []
                while remaining do
                    c ← remaining.pop()
                    for each combination of length cycleLength of c with cells in remaining do
                        if combination is a simple cycle then
                            merge combination into cycle
                            move cycle to results
                    if no cycle formed then
                        move c to partition
                cycleLength + +
            move partition to results
    return results
```

3.4 Initialization

3.4.1 Computing Δ^2

Δ^2 is computed as follows

$\Delta_2 g \leftarrow \{\}$

for each $h \mapsto z \in g$ **do**

$\Delta_2 g(h) \leftarrow \text{factorize}(\Delta(z))$

$\Delta^2 \leftarrow \{\}$

for each $h \mapsto z_1 \otimes z_2 \in \Delta_2 g$ **do**

$\Delta^2(h) \leftarrow \text{f}(z_1) \otimes \text{f}(z_2)$

It should be noted that while the coderivative property of Δ_2 ensures that cycles are mapped to cycles, it only guarantees that $\forall_{z \in Z(C)}, \Delta_2(z) \in Z(C \otimes C)$. In order for the function f to be applied to the output of $\Delta_2 g$, it must be transformed from the form $Z(C \otimes C)$ to $Z(C) \otimes Z(C)$, which is what the `factorize_cycles` method does to a limited extent.

3.4.2 Computing g^2

g^2 is computed as follows

$(g \otimes g)\Delta^2 \leftarrow \{\}$

for each $h \mapsto h_1 \otimes h_2 \in \Delta^2$ **do**

$(g \otimes g)\Delta^2(h) \leftarrow g(h_1) \otimes g(h_2)$

$\varphi_0 \leftarrow \{\}$

for each $h \in H$ **do**

$\varphi_0(h) \leftarrow \text{factorize}(\Delta_2 g(h) + (g \otimes g)\Delta^2(h) \mod 2)$

$g^2 \leftarrow \{\}$

for each $h \mapsto c \otimes c \in \varphi_0$ **do**

$g^2(h) \leftarrow \text{preboundary}(c \otimes c)$

3.4.3 Verification

Once Δ^2 and g^2 are computed, their consistency is verified by checking that both

$$(g \otimes g)\Delta^2 + \Delta_2 g + \nabla(g^2) = 0 \tag{3.2}$$

and

$$(1 \otimes \Delta^2)\Delta^2 + (\Delta^2 \otimes 1)\Delta^2 = 0 \tag{3.3}$$

are satisfied.

3.4.4 Computing Δ_3

If Δ_2 is non-coassociative, Δ_3 must be computed prior to proceeding with the induction of Δ^3. Here Algorithm 4 is applied to the coassociative difference $(1 \otimes \Delta_2 + \Delta_2 \otimes 1)\Delta_2$.

3.5 Induction Steps

3.5.1 Computing φ_1

For the first induction step, φ_1 is computed as the sum of the components:

$$(g^2 \otimes g)\Delta^2 + (g \otimes g^2)\Delta^2 + (\Delta_2 \otimes 1)g^2 + (1 \otimes \Delta_2)g^2 + \Delta_3 g \tag{3.4}$$

which are shown as corresponding to faces of J_3 in Figure 2. These compositions are computed in a manner identical to that indicated for computing the composition $(g \otimes g)\Delta^2$ from Section 3.4.2. As a consistency check we verify that $\nabla(\varphi_1) = 0$ is satisfied.

3.5.2 Computing Δ^3

Δ^3 is computed in the same manner as Δ^2

for each $h \mapsto z3 \in \varphi_1$ **do**

  $\varphi_1(h) \leftarrow \text{factorize}(z3)$

$\Delta^3 \leftarrow \{\}$

for each $h \mapsto z_1 \otimes z_2 \otimes z_3 \in \varphi_2$ **do**

  $\Delta^3(h) \leftarrow \text{f}(z_1) \otimes \text{f}(z_2) \otimes \text{f}(z_3)$

3.5.3 Computing g^3

Likewise, g^3 is computed in a manner identical to g^2, by applying the `preboundary` method to the difference of φ_1 and $(g \otimes g \otimes g)\Delta^3$.

The consistency of g^3 and Δ^3 is verified by checking the relation

$$\varphi_1 + (g \otimes g \otimes g)\Delta^3 + \nabla(g^3) = 0 \tag{3.5}$$

and the pentagonal K_3-boundary relation

$$(\Delta^3 \otimes 1 + 1 \otimes \Delta^3)\Delta^2 + (\Delta^2 \otimes 1 \otimes 1 + 1 \otimes \Delta^2 \otimes 1 + 1 \otimes 1 \otimes \Delta^2)\Delta^3 = 0 \tag{3.6}$$

Chapter 4

Example Computations

In this chapter we present the application of the transfer algorithm software to a handful of cell complexes and differential graded coalgebras. A summary of the program output is provided, rather than the raw output. Raw output can be obtained by running the provided command relative to the root directory of the software repository.

4.1 Two-Component Unlink

The following construction is from Umble [8]. Let UN be the link complement (of open tubular neighborhoods U_i) of two unlinked unknots in the lower hemisphere of S^3. Its boundary can be realized as the wedge product $t_1 \vee t_2 \vee S^2$ of pinched spheres t_i in the lower hemisphere of S^3 and the equatorial 2-sphere $S^2 \subset S^3$ with the upper hemispherical 3-ball p attached along S^2 and $q =$ (lower hemispherical 3-ball) $\smallsetminus (U_1 \cup U_2)$ attached along S^2.

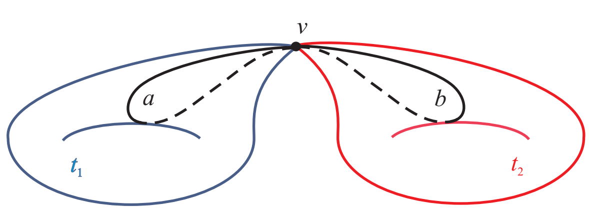


Figure 4-1: $\partial(UN) \subset S^3$

4.1.1 Input

Cellular chains with $\mathbb{Z}_2$ coefficients

$$C_0\left(UN\right)=\{v\}$$
$$C_1\left(UN\right)=\{a,b\}$$
$$C_2\left(UN\right)=\{s,t_1,t_2\}$$
$$C_3\left(UN\right)=\{p,q\}$$

Boundary Map

$$\partial p=s$$
$$\partial q=s+t_1+t_2$$

Coalgebra Structure on Cellular Chains

$$\Delta v=v\otimes v$$
$$\Delta a=v\otimes a+a\otimes v$$
$$\Delta b=v\otimes b+b\otimes v$$
$$\Delta s=v\otimes s+s\otimes v$$
$$\Delta t_1=v\otimes t_1+t_1\otimes v$$
$$\Delta t_2=v\otimes t_2+t_2\otimes v$$
$$\Delta p=v\otimes p+p\otimes v$$
$$\Delta q=v\otimes q+q\otimes v$$

4.1.2 Cellular Homology

$$\bar{H}_0\left(UN\right)=\mathbb{Z}_2\left[v\right]$$
$$\bar{H}_1\left(UN\right)=\mathbb{Z}_2\left[a\right]\oplus\mathbb{Z}_2\left[b\right]$$
$$\bar{H}_2\left(UN\right)=\mathbb{Z}_2\left[t_1\right]$$
$$\bar{H}_i\left(UN\right)=0,\ i\geq 3$$

4.1.3 Program Results

Command

```
python transferPart.py data/unlinked.tex
```

Output Summary

$$(\Delta_2 \otimes 1 + 1 \otimes \Delta_2)\,\Delta_2 = 0$$

$$g = v\partial_{h0_0} + a\partial_{h1_0} + b\partial_{h1_1} + t_1\partial_{h2_0}$$

$$\Delta_2 g = (v \otimes v)\partial_{h0_0} + (v \otimes a + a \otimes v)\partial h1_0$$
$$+(v \otimes b + b \otimes v)\partial h1_1 + (v \otimes t_1 + t_1 \otimes v)\partial h2_0$$

$$\Delta^2 = (h0_0 \otimes h0_0)\partial_{h0_0} + (h1_0 \otimes h0_0 + h0_0 \otimes h1_0)\partial_{h1_0}$$
$$+(h1_1 \otimes h0_0 + h0_0 \otimes h1_1)\partial_{h1_1} + (h0_0 \otimes h2_0 + h2_0 \otimes h0_0)\partial_{h2_0}$$

$$(g \otimes g)\Delta^2 + \Delta_2 g = 0$$
$$g^2 = 0$$

$$\Delta_i = 0,\ i \geq 3$$
$$\varphi_1 = 0$$
$$\Delta^3 = 0$$
$$g^3 = 0$$

4.1.4 Discussion

In the case of the two component unlink, the program results yield a primitive Δ^2 and $\Delta^i = 0, \forall i \geq 3$.

4.2 Hopf Link

The following construction is from Umble [8]. Let LN be the link complement (of open tubular neighborhoods U_i) of the Hopf link in the lower hemisphere of S^3. Its boundary can be realized as tori $t_1' \cup t_2'$ sharing loops a and b in the lower hemisphere of S^3 (as shown in 4-2) wedged with the equatorial 2-sphere $S^2 \subset S^3$ with the upper hemispherical 3-ball p attached along S^2 and $q' = (U_1 \cup U_2)' \cap$ (lower hemispherical 3-ball) attached along S^2.

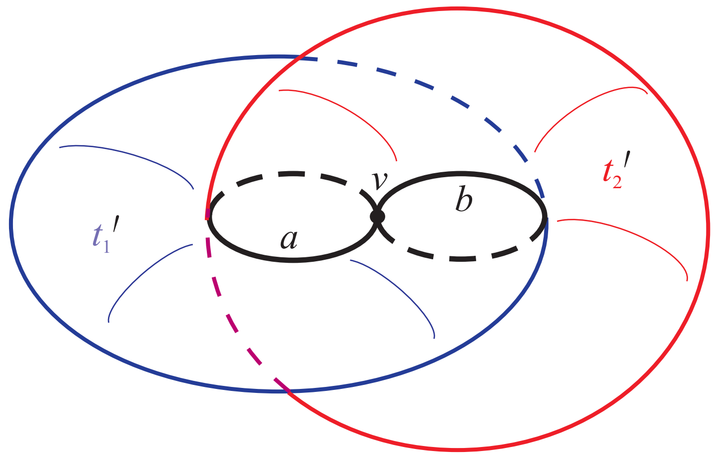


Figure 4-2: $\partial(LN) \subset S^3$

4.2.1 Input

Cellular chains with $\mathbb{Z}_2$ coefficients

$$C_0\,(LN) = \{v\}$$

$$C_1\,(LN) = \{a, b\}$$

$$C_2\,(LN) = \{s, t_1, t_2\}$$

$$C_3\,(LN) = \{p, q\}$$

Boundary Map

$$\partial p = s$$

$$\partial q = s + t_1 + t_2$$

Coalgebra Structure on Cellular Chains

$$\Delta v = v \otimes v$$

$$\Delta a = v \otimes a + a \otimes v$$

$$\Delta b = v \otimes b + b \otimes v$$

$$\Delta s = v \otimes s + s \otimes v$$

$$\Delta t_1 = v \otimes t_1 + a \otimes b + b \otimes a + t_1 \otimes v$$

$$\Delta t_2 = v \otimes t_2 + a \otimes b + b \otimes a + t_2 \otimes v$$

$$\Delta p = v \otimes p + p \otimes v$$

$$\Delta q = v \otimes q + q \otimes v$$

4.2.2 Cellular Homology

$\bar{H}_0\left(LN\right) = \mathbb{Z}_2\left[v\right]$

$\bar{H}_1\left(LN\right) = \mathbb{Z}_2\left[a\right] \oplus \mathbb{Z}_2\left[b\right]$

$\bar{H}_2\left(LN\right) = \mathbb{Z}_2\left[t_1\right]$

$\bar{H}_i\left(LN\right) = 0,\ i \geq 3$

Note that the cellular homology for the Hopf link is identical to the two component unlink in the previous section.

4.2.3 Program Results

Command

```
python transferPart.py data/linked.tex
```

Output Summary

$$\left(\Delta_2 \otimes 1 + 1 \otimes \Delta_2\right)\Delta_2 = 0$$

$$g = v\partial_{h0_0} + a\partial_{h1_0} + b\partial_{h1_1} + t_1\partial_{h2_0}$$

$$\Delta_2 g = (v \otimes v)\partial_{h0_0} + (v \otimes a + a \otimes v)\partial h1_0 + (v \otimes b + b \otimes v)\partial h1_1$$
$$+(a \otimes b + b \otimes a + v \otimes t_1 + t_1 \otimes v)\partial h2_0$$

$$\Delta^2 = (h0_0 \otimes h0_0)\partial_{h0_0} + (h1_0 \otimes h0_0 + h0_0 \otimes h1_0)\partial_{h1_0} + (h1_1 \otimes h0_0 + h0_0 \otimes h1_1)\partial_{h1_1}$$
$$+(h1_0 \otimes h1_1 + h1_1 \otimes h1_0 + h0_0 \otimes h2_0 + h2_0 \otimes h0_0)\partial_{h2_0}$$

$$(g \otimes g)\Delta^2 + \Delta_2 g = 0$$

$$g^2 = 0$$

$$\Delta_i = 0,\ i \geq 3$$

$$\varphi_1 = 0$$

$$\Delta^3 = 0$$

$$g^3 = 0$$

4.2.4 Discussion

In the case of the Hopf link, the program results yield a non-primitive Δ^2 and $\Delta^i = 0, \forall i \geq 3$.

4.3 A Coassociative Differential Graded Coalgebra

This example was constructed by Umble and Sandeblidze [5].

4.3.1 Input

Cellular chains with $\mathbb{Z}_2$ coefficients

$$C_0(X) = \{v\}$$
$$C_1(X) = \{a, b\}$$
$$C_2(X) = \{aa, ab\}$$

Boundary Map

$$\partial aa = b$$

Coalgebra Structure on Cellular Chains

$$\Delta v = v \otimes v$$
$$\Delta a = v \otimes a + a \otimes v$$
$$\Delta b = v \otimes b + b \otimes v$$
$$\Delta aa = v \otimes aa + a \otimes a + aa \otimes v$$
$$\Delta ab = v \otimes ab + a \otimes b + ab \otimes v$$

4.3.2 Cellular Homology

$$\bar{H}_0(X) = \mathbb{Z}_2[v]$$
$$\bar{H}_1(X) = \mathbb{Z}_2[a]$$
$$\bar{H}_2(X) = \mathbb{Z}_2[ab]$$
$$\bar{H}_i(X) = 0, \; i \geq 3$$

4.3.3 Program Results

Command

```
python transferPart.py -hg data/hom_dgc.py data/dgc.tex
```

Output Summary

$$(\Delta_2 \otimes 1 + 1 \otimes \Delta_2)\,\Delta_2 = 0$$

$$g = v\partial_{h0_0} + a\partial_{h1_0} + ab\partial_{h2_0}$$

$$\Delta_2 g = (v \otimes v)\partial_{h0_0} + (v \otimes a + a \otimes v)\partial h1_0$$
$$+(v \otimes ab + a \otimes b + ab \otimes v)\partial h2_0$$

$$\Delta^2 = (h0_0 \otimes h0_0)\partial_{h0_0} + (h1_0 \otimes h0_0 + h0_0 \otimes h1_0)\partial_{h1_0} + (h0_0 \otimes h2_0 + h2_0 \otimes h0_0)\partial_{h2_0}$$

$$(g \otimes g)\Delta^2 + \Delta_2 g = (a \otimes b)\partial_{h2_0}$$

$$g^2 = (a \otimes aa)\partial_{h2_0}$$

$$\nabla\left(g^2\right) = (a \otimes b)\partial_{h2_0}$$

$$(g \otimes g)\Delta^2 + \Delta_2 g + \nabla\left(g^2\right) = 0$$

$$\Delta_i = 0,\ i \geq 3$$

$$\varphi_1 = (a \otimes a \otimes a)\partial_{h2_0}$$

$$\Delta^3 = (h1_0 \otimes h1_0 \otimes h1_0)\partial_{h2_0}$$

$$(g \otimes g \otimes g)\Delta^3 + \varphi_1 = 0$$

$$g^3 = 0$$

$$\varphi_2 = 0$$

$$\Delta^4 = 0$$

4.3.4 Discussion

The program results yield a primitive Δ^2, however, unlike the previous examples, the homotopy g^2 does not vanish.

4.4 A Non-Coassociative Differential Graded Coalgebra

This example, due to Umble [8], is a variant on the previous example with the difference that a non-coassociative Δ_2 is defined. This triggers the program to compute a Δ_3 and Δ_4 on chains, which serves to test the correctness of the portion of the code used to compute the preboundary of a chain map. Homology is computed using the direct interface to CHomP.

4.4.1 Input

Cellular chains with $\mathbb{Z}_2$ coefficients

$$C_0(X) = \{v\}$$
$$C_1(X) = \{a, b\}$$
$$C_2(X) = \{aa, ab, c\}$$
$$C_3(X) = \{aba\}$$

Boundary Map

$$\partial aa = b$$
$$\partial aba = c$$

Coalgebra Structure on Cellular Chains

$$\Delta v = v \otimes v$$
$$\Delta a = v \otimes a + a \otimes v$$
$$\Delta b = v \otimes b + b \otimes v$$
$$\Delta aa = v \otimes aa + a \otimes a + aa \otimes v$$
$$\Delta ab = v \otimes ab + a \otimes b + ab \otimes v$$
$$\Delta c = v \otimes c + c \otimes v$$
$$\Delta aba = v \otimes aba + ab \otimes a + aba \otimes v$$

4.4.2 Cellular Homology

$$\bar{H}_0(X) = \mathbb{Z}_2[v]$$
$$\bar{H}_1(X) = \mathbb{Z}_2[a]$$
$$\bar{H}_2(X) = \mathbb{Z}_2[ab]$$
$$\bar{H}_i(X) = 0, \ i \geq 3$$

4.4.3 Program Results

Command

```
python transferPart.py data/dgc2.tex
```

Output Summary

$$g = v\partial_{h0_0} + a\partial_{h1_0} + ab\partial_{h2_0}$$

$$\Delta_2 g = (v \otimes v)\partial_{h0_0} + (v \otimes a + a \otimes v)\partial h1_0$$
$$+(v \otimes ab + a \otimes b + ab \otimes v)\partial h2_0$$

$$\Delta^2 = (h0_0 \otimes h0_0)\partial_{h0_0} + (h1_0 \otimes h0_0 + h0_0 \otimes h1_0)\partial_{h1_0} + (h0_0 \otimes h2_0 + h2_0 \otimes h0_0)\partial_{h2_0}$$

$$(g \otimes g)\Delta^2 + \Delta_2 g = (a \otimes b)\partial_{h2_0}$$

$$g^2 = (a \otimes aa)\partial_{h2_0}$$

$$\nabla\left(g^2\right) = (a \otimes b)\partial_{h2_0}$$

$$(g \otimes g)\Delta^2 + \Delta_2 g + \nabla\left(g^2\right) = 0$$

$$\left(\Delta_2 \otimes 1 + 1 \otimes \Delta_2\right)\Delta_2 = (a \otimes b \otimes a)\partial_{aba}$$

$$\Delta_3 = (a \otimes aa \otimes a)\partial_{aba}$$

$$\left(\Delta_2 \otimes 1 + 1 \otimes \Delta_2\right)\Delta_2 + \nabla\left(\Delta_3\right) = 0$$

$$\varphi_1 = (a \otimes a \otimes a)\partial_{h2_0}$$

$$\Delta^3 = (h1_0 \otimes h1_0 \otimes h1_0)\partial_{h2_0}$$

$$(g \otimes g \otimes g)\Delta^3 + \varphi_1 = 0$$

$$g^3 = 0$$

$$(1 \otimes 1 \otimes \Delta_2)\Delta_3 + (1 \otimes \Delta_2 \otimes 1)\Delta_3 + (\Delta_2 \otimes 1 \otimes 1)\Delta_3$$
$$+(\Delta_3 \otimes 1)\Delta_2 + (1 \otimes \Delta_3)\Delta_2 = (a \otimes a \otimes a \otimes a)\partial_{aba}$$

$$\Delta_4 = (a \otimes a \otimes a \otimes a)\partial_c$$

$$(1 \otimes 1 \otimes \Delta_2)\Delta_3 + (1 \otimes \Delta_2 \otimes 1)\Delta_3 + (\Delta_2 \otimes 1 \otimes 1)\Delta_3$$
$$+(\Delta_3 \otimes 1)\Delta_2 + (1 \otimes \Delta_3)\Delta_2 + \nabla\left(\Delta_4\right) = 0$$

$$\varphi_2 = 0$$

$$\Delta^4 = 0$$

4.4.4 Discussion

The program correctly identifies higher coproducts on chains consistent with the K_n-boundary relations.

Chapter 5

Transfer Algorithm Applied to Borromean Rings

We conclude the work with an application of the transfer algorithm to the link complement of the Borromean rings.

5.1 Chain Complex

The following construction is from Umble [7]. Let BR denote an open tubular neighborhood of the Borromean rings embedded in the "lower" hemisphere of S^3, and let $X = S^3 \smallsetminus BR$. Diagrams of the cellular decomposition are provided in Appendix B.

5.2 Input

The chain complex definition is provided in the file **data\borromean.tex**.

5.2.1 Cellular chains with $\mathbb{Z}_2$ coefficients

$$C_0(X) = \{1, 2, \ldots, 11\}$$
$$C_1(X) = \{m_1, \ldots, m_{14}, c_1, \ldots, c_{18}\}$$
$$C_2(X) = \{a_1, \ldots, a_4, e_1, e_2, s_1, \ldots, s_{10}, t_1, \ldots, t_8\}$$
$$C_3(X) = \{D, q_1, \ldots, q_4\}$$

5.2.2 Boundary Map

$$\partial D = a_1 + a_2 + a_3 + a_4$$
$$\partial q_1 = a_1 + e_1 + s_3 + s_7 + s_9 + t_1 + t_5$$
$$\partial q_2 = a_2 + e_2 + s_4 + s_8 + s_{10} + t_3 + t_7$$
$$\partial q_3 = a_3 + e_1 + s_1 + s_5 + s_{11} + t_2 + t_6$$
$$\partial q_4 = a_4 + e_2 + s_2 + s_6 + s_{12} + t_4 + t_8$$

$$\partial a_1 = m_{14} + c_{17}$$
$$\partial a_2 = m_{14} + c_{18}$$
$$\partial a_3 = m_1 + c_{17}$$
$$\partial a_4 = m_1 + c_{18}$$
$$\partial t_1 = m_{10} + m_{11} + c_1 + c_3 + c_4$$
$$\partial t_2 = m_4 + m_5 + c_1 + c_3 + c_4$$
$$\partial t_3 = m_{10} + m_{11} + c_7 + c_8 + c_{11}$$
$$\partial t_4 = m_4 + m_5 + c_7 + c_8 + c_{11}$$
$$\partial t_5 = m_8 + m_9 + c_2 + c_5 + c_6$$
$$\partial t_6 = m_6 + m_7 + c_2 + c_5 + c_6$$
$$\partial t_7 = m_8 + m_9 + c_9 + c_{10} + c_{12}$$
$$\partial t_8 = m_6 + m_7 + c_9 + c_{10} + c_{12}$$

$$\partial e_1 = c_1 + c_2 + c_{13} + c_{15} + c_{17}$$
$$\partial e_2 = c_{11} + c_{12} + c_{14} + c_{16} + c_{18}$$
$$\partial s_1 = m_1 + m_2 + m_3 + m_4 + m_6 + c_3 + c_5$$
$$\partial s_2 = m_1 + m_2 + m_3 + m_4 + m_6 + c_7 + c_9$$
$$\partial s_3 = m_9 + m_{12} + c_5 + c_{15}$$
$$\partial s_4 = m_9 + m_{12} + c_9 + c_{16}$$
$$\partial s_5 = m_3 + m_7 + c_6 + c_{13}$$
$$\partial s_6 = m_3 + m_7 + c_{10} + c_{14}$$
$$\partial s_7 = m_{10} + m_{12} + m_{13} + c_4 + c_6$$
$$\partial s_8 = m_{10} + m_{12} + m_{13} + c_8 + c_{10}$$
$$\partial s_9 = m_8 + m_{11} + m_{13} + m_{14} + c_3 + c_{13}$$
$$\partial s_{10} = m_8 + m_{11} + m_{13} + m_{14} + c_7 + c_{14}$$
$$\partial s_{11} = m_2 + m_5 + c_4 + c_{15}$$
$$\partial s_{12} = m_2 + m_5 + c_8 + c_{16}$$

$$
\begin{array}{lllll}
\partial m_1 = 1+11 & \partial m_8 = 9+10 & \partial c_1 = 1+5 & \partial c_8 = 3+4 & \partial c_{15} = 5+6 \\
\partial m_2 = 3+6 & \partial m_9 = 6+7 & \partial c_2 = 6+10 & \partial c_9 = 7+8 & \partial c_{16} = 5+6 \\
\partial m_3 = 8+11 & \partial m_{10} = 4+5 & \partial c_3 = 2+3 & \partial c_{10} = 8+9 & \partial c_{17} = 1+11 \\
\partial m_4 = 1+2 & \partial m_{11} = 1+2 & \partial c_4 = 3+4 & \partial c_{11} = 1+5 & \partial c_{18} = 1+11 \\
\partial m_5 = 4+5 & \partial m_{12} = 5+8 & \partial c_5 = 7+8 & \partial c_{12} = 6+10 & \\
\partial m_6 = 6+7 & \partial m_{13} = 3+9 & \partial c_6 = 8+9 & \partial c_{13} = 10+11 & \\
\partial m_7 = 9+10 & \partial m_{14} = 1+11 & \partial c_7 = 2+3 & \partial c_{14} = 10+11 &
\end{array}
$$

5.2.3 Coalgebra Structure on Cellular Chains

Order the vertices $v_1 < v_2 < \cdots < v_{11}$ and orient the edges accordingly. Define a diagonal $\Delta_2 : C_*(X) \to C_*(X) \otimes C_*(X)$ as follows:

$$\Delta v_i = v_i \otimes v_i$$

$$\Delta c_i = \textit{(minimal vertex of } c_i\textit{)} \otimes c_i + c_i \otimes \textit{(maximal vertex of } c_i\textit{)}$$

$$\Delta m_i = \textit{(minimal vertex of } m_i\textit{)} \otimes m_i + m_i \otimes \textit{(maximal vertex of } m_i\textit{)}$$

$$\Delta a_i = v_1 \otimes a_i + a_i \otimes v_{11}$$

$$\Delta e_1 = v_1 \otimes e_1 + e_1 \otimes v_{11} + c_1 \otimes (c_{15} + c_2 + c_{13}) + c_{15} \otimes (c_2 + c_{13}) + c_2 \otimes c_{13}$$

$$\Delta e_2 = v_1 \otimes e_2 + e_2 \otimes v_{11} + c_{11} \otimes (c_{16} + c_{12} + c_{14}) + c_{16} \otimes (c_{12} + c_{14}) + c_{12} \otimes c_{14}$$

$$
\begin{aligned}
\Delta s_1 &= v_1 \otimes s_1 + s_1 \otimes v_{11} + m_4 \otimes (c_3 + m_2 + m_6 + c_5 + m_3) + c_3 \otimes (m_2 + m_6 + c_5 + m_3) \\
&\quad + m_2 \otimes (m_6 + c_5 + m_3) + m_6 \otimes (c_5 + m_3) + c_5 \otimes m_3 \\
\Delta s_2 &= v_1 \otimes s_1 + s_1 \otimes v_{11} + m_4 \otimes (c_7 + m_2 + m_6 + c_9 + m_3) + c_7 \otimes (m_2 + m_6 + c_9 + m_3) \\
&\quad + m_2 \otimes (m_6 + c_9 + m_3) + m_6 \otimes (c_9 + m_3) + c_9 \otimes m_3 \\
\Delta s_3 &= v_5 \otimes s_3 + s_3 \otimes v_8 + c_{15} \otimes (m_9 + c_5) + m_9 \otimes c_5 \\
\Delta s_4 &= v_5 \otimes s_4 + s_4 \otimes v_8 + c_{16} \otimes (m_9 + c_9) + m_9 \otimes c_9 \\
\Delta s_5 &= v_8 \otimes s_5 + s_5 \otimes v_{11} + c_6 \otimes (m_7 + c_{13}) + m_7 \otimes c_{13} \\
\Delta s_6 &= v_8 \otimes s_6 + s_6 \otimes v_{11} + c_{10} \otimes (m_7 + c_{14}) + m_7 \otimes c_{14} \\
\Delta s_7 &= v_3 \otimes s_7 + s_7 \otimes v_9 + c_4 \otimes (m_{10} + m_{12} + c_6) + m_{10} \otimes (m_{12} + c_6) + m_{12} \otimes c_6 \\
\Delta s_8 &= v_3 \otimes s_8 + s_8 \otimes v_9 + c_8 \otimes (m_{10} + m_{12} + c_{10}) + m_{10} \otimes (m_{12} + c_{10}) + m_{12} \otimes c_{10} \\
\Delta s_9 &= v_1 \otimes s_9 + s_9 \otimes v_{11} + m_{11} \otimes (c_3 + m_{13} + m_8 + c_{13}) + c_3 \otimes (m_{13} + m_8 + c_{13}) \\
&\quad m_{13} \otimes (m_8 + c_{13}) + m_8 \otimes c_{13} \\
\Delta s_{10} &= v_1 \otimes s_{10} + s_{10} \otimes v_{11} + m_{11} \otimes (c_7 + m_{13} + m_8 + c_{14}) + c_7 \otimes (m_{13} + m_8 + c_{14}) \\
&\quad m_{13} \otimes (m_8 + c_{14}) + m_8 \otimes c_{14} \\
\Delta s_{11} &= v_3 \otimes s_{11} + s_{11} \otimes v_6 + c_4 \otimes (m_5 + c_{15}) + m_5 \otimes c_{15} \\
\Delta s_{12} &= v_3 \otimes s_{12} + s_{12} \otimes v_6 + c_8 \otimes (m_5 + c_{16}) + m_5 \otimes c_{16}
\end{aligned}
$$

$$
\begin{aligned}
\Delta t_1 &= v_1 \otimes t_1 + t_1 \otimes v_5 + m_{11} \otimes (c_3 + c_4 + m_{10}) + c_3 \otimes (c_4 + m_{10}) + c_4 \otimes m_{10} \\
\Delta t_2 &= v_1 \otimes t_2 + t_2 \otimes v_5 + m_4 \otimes (c_3 + c_4 + m_5) + c_3 \otimes (c_4 + m_5) + c_4 \otimes m_5 \\
\Delta t_3 &= v_1 \otimes t_3 + t_3 \otimes v_5 + m_{11} \otimes (c_7 + c_8 + m_{10}) + c_7 \otimes (c_8 + m_{10}) + c_8 \otimes m_{10} \\
\Delta t_4 &= v_1 \otimes t_4 + t_4 \otimes v_5 + m_4 \otimes (c_7 + c_8 + m_5) + c_7 \otimes (c_8 + m_5) + c_8 \otimes m_5 \\
\Delta t_5 &= v_6 \otimes t_5 + t_5 \otimes v_{10} + m_9 \otimes (c_5 + c_6 + m_8) + c_5 \otimes (c_6 + m_8) + c_6 \otimes m_8 \\
\Delta t_6 &= v_6 \otimes t_6 + t_6 \otimes v_{10} + m_6 \otimes (c_5 + c_6 + m_7) + c_5 \otimes (c_6 + m_7) + c_6 \otimes m_7 \\
\Delta t_7 &= v_6 \otimes t_7 + t_7 \otimes v_{10} + m_9 \otimes (c_9 + c_{10} + m_8) + c_9 \otimes (c_{10} + m_8) + c_{10} \otimes m_8 \\
\Delta t_8 &= v_6 \otimes t_8 + t_8 \otimes v_{10} + m_6 \otimes (c_9 + c_{10} + m_7) + c_9 \otimes (c_{10} + m_7) + c_{10} \otimes m_7 \\
\Delta q_1 &= v_1 \otimes q_1 + q_1 \otimes v_{11} + t_1 \otimes (m_{12} + c_6 + m_8 + c_{13}) + (c_1 + c_{15}) \otimes t_5 + t_5 \otimes c_{13} \\
&\quad + c_1 \otimes s_3 + s_3 \otimes (c_6 + m_8 + c_{13}) + (m_{11} + c_3) \otimes s_7 + s_7 \otimes (m_8 + c_{13}) \\
\Delta q_2 &= v_1 \otimes q_2 + q_2 \otimes v_{11} + t_3 \otimes (m_{12} + c_{10} + m_8 + c_{14}) + (c_{11} + c_{16}) \otimes t_7 + t_7 \otimes c_{14} \\
&\quad + c_{11} \otimes s_4 + s_4 \otimes (c_{10} + m_8 + c_{14}) + (m_{11} + c_7) \otimes s_8 + s_8 \otimes (m_8 + c_{14}) \\
\Delta q_3 &= v_1 \otimes q_3 + q_3 \otimes v_{11} + t_2 \otimes (m_6 + c_5 + m_3 + c_{15}) + (c_1 + c_{15}) \otimes t_6 + t_6 \otimes c_{13} \\
&\quad + (m_4 + c_3) \otimes s_{11} + s_{11} \otimes (m_6 + c_5 + m_3) + (c_1 + c_{15} + m_6 + c_5) \otimes s_5 \\
\Delta q_4 &= v_1 \otimes q_4 + q_4 \otimes v_{11} + t_{42} \otimes (m_6 + c_9 + m_3 + c_{16}) + (c_{11} + c_{16}) \otimes t_8 + t_8 \otimes c_{14} \\
&\quad + (m_4 + c_7) \otimes s_{12} + s_{12} \otimes (m_6 + c_9 + m_3) + (c_{11} + c_{16} + m_6 + c_9) \otimes s_6 \\
\Delta D &= v_1 \otimes D + D \otimes v_{11}.
\end{aligned}
$$

The **validateCoproduct.py** script was used to verify that Δ_2 so defined is a coderivation with respect to the chain complex X.

5.3 Cellular Homology

The following results were obtained using SageMath/CHomP:

$$
\begin{aligned}
&\bar{H}_0(X) = \mathbb{Z}_2[v_1] \\
&\bar{H}_1(X) = \mathbb{Z}_2[m_4 + m_{11}] \oplus \mathbb{Z}_2[c_3 + c_7] \oplus \mathbb{Z}_2[m_6 + m_9] \\
&\bar{H}_2(X) = \mathbb{Z}_2[t_1 + t_2 + t_3 + t_4] \oplus \mathbb{Z}_2[t_5 + t_6 + t_7 + t_8] \\
&\bar{H}_i(X) = 0,\ i \geq 3
\end{aligned}
$$

5.4 Program Results

5.4.1 Command

```
python transferBR.py -hg data/hom_borromean.py data/borromean.tex
```

5.4.2 Output Summary

The coproduct on chains is not coassociative: $(\Delta_2 \otimes 1 + 1 \otimes \Delta_2)\,\Delta_2 \neq 0$. A full expression of the coassociative difference is omitted, but can easily be obtained from program output.

$$
\begin{aligned}
g = {} & (v_1)\partial_{h0_0} \\
& +(m_{11} + m_4)\partial_{h1_0} + (c_3 + c_7)\partial_{h1_1} + (m_6 + m_9)\partial_{h1_2} \\
& +(t_5 + t_6 + t_7 + t_8)\partial_{h2_1} + (t_1 + t_2 + t_3 + t_4)\partial_{h2_0}
\end{aligned}
$$

$$
\begin{aligned}
\Delta_2 g = {} & ((v_1) \otimes (v_1))\partial_{h0_0} \\
& +((m_{11} + m_4) \otimes (v_2) + (v_1) \otimes (m_4 + m_{11}))\partial_{h1_0} \\
& +((c_7 + c_3) \otimes (v_3) + (v_2) \otimes (c_3 + c_7))\partial_{h1_1} \\
& +((v_6) \otimes (m_6 + m_9) + (m_9 + m_6) \otimes (v_7))\partial_{h1_2} \\
& +((c_7 + c_3) \otimes (m_5 + m_{10}) + (t_3 + t_2 + t_4 + t_1) \otimes (v_5) \\
& \quad +(v_1) \otimes (t_3 + t_4 + t_1 + t_2) + (m_4 + m_{11}) \otimes (c_8 + c_4) \\
& \quad +(m_4 + m_{11}) \otimes (c_3 + c_7) + (c_4 + c_8) \otimes (m_5 + m_{10}))\partial_{h2_0} \\
& +((c_9 + c_5) \otimes (m_8 + m_7) + (t_8 + t_5 + t_6 + t_7) \otimes (v_{10}) \\
& \quad +(m_9 + m_6) \otimes (c_6 + c_{10}) + (m_6 + m_9) \otimes (c_5 + c_9) \\
& \quad +(v_6) \otimes (t_7 + t_5 + t_8 + t_6) + (c_6 + c_{10}) \otimes (m_8 + m_7))\partial_{h2_1}
\end{aligned}
$$

$$
\begin{aligned}
\Delta^2 = {} & (h0_0 \otimes h0_0)\partial_{h0_0} \\
& +(h1_0 \otimes h0_0 + h0_0 \otimes h1_0)\partial_{h1_0} \\
& +(h1_1 \otimes h0_0 + h0_0 \otimes h1_1)\partial_{h1_1} \\
& +(h0_0 \otimes h1_2 + h1_2 \otimes h0_0)\partial_{h1_2} \\
& +(h0_0 \otimes h2_0 + h2_0 \otimes h0_0)\partial_{h2_0} \\
& +(h2_1 \otimes h0_0 + h0_0 \otimes h2_1)\partial_{h2_1}
\end{aligned}
$$

$$
\begin{aligned}
(g \otimes g)\Delta^2 + \Delta_2 g = {} & ((m_4 + m_{11}) \otimes (v_1 + v_2))\partial_{h1_0} \\
& +((c_7 + c_3) \otimes (v_1 + v_3) + (v_1 + v_2) \otimes (c_3 + c_7))\partial_{h1_1} \\
& +((v_1 + v_6) \otimes (m_6 + m_9) + (m_6 + m_9) \otimes (v_1 + v_7))\partial_{h1_2} \\
& +((c_7 + c_3 + c_4 + c_8) \otimes (m_5 + m_{10}) + (t_3 + t_2 + t_4 + t_1) \otimes (v_1 + v_5) \\
& \quad +(m_4 + m_{11}) \otimes (c_3 + c_7 + c_8 + c_4))\partial_{h2_0} \\
& +((v_1 + v_6) \otimes (t_7 + t_5 + t_8 + t_6) + (c_5 + c_9 + c_6 + c_{10}) \otimes (m_7 + m_8) \\
& \quad +(t_8 + t_5 + t_6 + t_7) \otimes (v_1 + v_{10}) + (m_6 + m_9) \otimes (c_5 + c_9 + c_6 + c_{10}))\partial_{h2_1}
\end{aligned}
$$

$$
\begin{aligned}
g^2 = {} & (((m_4 + m_{11}) \otimes (m_4))\partial_{h1_0} \\
& +((c_7 + c_3) \otimes (m_4 + c_3) + (m_4) \otimes (c_3 + c_7))\partial_{h1_1} \\
& +((m_4 + c_3 + m_2) \otimes (m_6 + m_9) + (m_6 + m_9) \otimes (m_4 + c_3 + m_2 + m_6))\partial_{h1_2} \\
& +((s_5 + s_6 + s_7 + s_8 + s_9 + s_{10}) \otimes (m_5 + m_{10}) \\
& \quad +(t_3 + t_2 + t_4 + t_1) \otimes (m_1 + m_3 + m_{12}) \\
& \quad +(m_4 + m_{11}) \otimes (s_5 + s_6 + s_7 + s_8 + s_9 + s_{10}))\partial_{h2_0} \\
& +(m_4 + c_3 + m_2) \otimes (t_5 + t_7 + t_8 + t_6) \\
& \quad +(t_8 + t_5 + t_6 + t_7) \otimes (m_4 + c_3 + m_{13} + m_7) \\
& \quad +(s_1 + s_2 + s_5 + s_6 + s_9 + s_{10}) \otimes (m_7 + m_8) \\
& \quad +(m_6 + m_9) \otimes (s_1 + s_2 + s_5 + s_6 + s_9 + s_{10}))\partial_{h2_1}
\end{aligned}
$$

$$(g \otimes g)\Delta^2 + \Delta_2 g + \nabla(g^2) = 0$$

$$\Delta_3(e_1) = (c_1 + c_{15}) \otimes c_2 \otimes c_{13} + c_1 \otimes c_{15} \otimes (c_2 + c_{13})$$

$$\Delta_3(e_2) = (c_{11} + c_16) \otimes c_{12} \otimes c_{14} + c_{11} \otimes c_{16} \otimes (c_{12} + c_{14})$$

$$
\begin{aligned}
\Delta_3(s_1) &= c_3 \otimes m_2 \otimes m_3 + m_6 \otimes c_5 \otimes m_3 + m_2 \otimes m_6 \otimes c_5 \\
&+ c_3 \otimes m_6 \otimes c_5 + m_2 \otimes m_6 \otimes m_3 + c_3 \otimes m_6 \otimes m_3 \\
&+ m_2 \otimes c_5 \otimes m_3 + m_4 \otimes c_3 \otimes c_5 + m_4 \otimes c_3 \otimes m_2 \\
&+ m_4 \otimes m_6 \otimes c_5 + m_4 \otimes c_3 \otimes m_3 + m_4 \otimes m_6 \otimes m_3 \\
&+ m_4 \otimes c_5 \otimes m_3 + m_4 \otimes m_2 \otimes m_6 + c_3 \otimes m_2 \otimes m_6 \\
&+ c_3 \otimes m_2 \otimes c_5 + m_4 \otimes c_3 \otimes m_6 + c_3 \otimes c_5 \otimes m_3 \\
&+ m_4 \otimes m_2 \otimes m_3 + m_4 \otimes m_2 \otimes c_5 \\
\Delta_3(s_2) &= c_7 \otimes m_2 \otimes c_9 + c_7 \otimes m_6 \otimes m_3 + c_7 \otimes c_9 \otimes m_3 \\
&+ m_6 \otimes c_9 \otimes m_3 + m_4 \otimes m_2 \otimes c_9 + m_4 \otimes c_9 \otimes m_3 \\
&+ m_2 \otimes m_6 \otimes m_3 + m_2 \otimes c_9 \otimes m_3 + m_4 \otimes c_7 \otimes m_2 \\
&+ c_7 \otimes m_6 \otimes c_9 + m_4 \otimes m_2 \otimes m_3 + m_4 \otimes m_2 \otimes m_6 \\
&+ m_4 \otimes m_6 \otimes m_3 + m_4 \otimes c_7 \otimes m_3 + m_4 \otimes m_6 \otimes c_9 \\
&+ c_7 \otimes m_2 \otimes m_6 + c_7 \otimes m_2 \otimes m_3 + m_4 \otimes c_7 \otimes c_9 \\
&+ m_4 \otimes c_7 \otimes m_6 + m_2 \otimes m_6 \otimes c_9 \\
\Delta_3(s_3) &= c_{15} \otimes m_9 \otimes c_5 \\
\Delta_3(s_4) &= c_{16} \otimes m_9 \otimes c_9 \\
\Delta_3(s_5) &= c_6 \otimes m_7 \otimes c_{13} \\
\Delta_3(s_6) &= c_{10} \otimes m_7 \otimes c_{14} \\
\Delta_3(s_7) &= c_4 \otimes m_{12} \otimes c_6 + c_4 \otimes m_{10} \otimes m_{12} + m_{10} \otimes m_{12} \otimes c_6 + c_4 \otimes m_{10} \otimes c_6 \\
\Delta_3(s_8) &= c_8 \otimes m_{10} \otimes m_{12} + m_{10} \otimes m_{12} \otimes c_{10} + c_8 \otimes m_{10} \otimes c_{10} + c_8 \otimes m_{12} \otimes c_{10} \\
\Delta_3(s_9) &= c_3 \otimes m_8 \otimes c_{13} + m_{11} \otimes m_{13} \otimes m_8 + m_{11} \otimes m_{13} \otimes c_{13} + c_3 \otimes m_{13} \otimes c_{13} \\
&+ m_{13} \otimes m_8 \otimes c_{13} + m_{11} \otimes m_8 \otimes c_{13} + m_{11} \otimes c_3 \otimes m_8 + c_3 \otimes m_{13} \otimes m_8 \\
&+ m_{11} \otimes c_3 \otimes m_{13} + m_{11} \otimes c_3 \otimes c_{13} \\
\Delta_3(s_{10}) &= m_{13} \otimes m_8 \otimes c_{14} + c_7 \otimes m_8 \otimes c_{14} + m_{11} \otimes m_{13} \otimes m_8 + m_{11} \otimes c_7 \otimes m_8 \\
&+ m_{11} \otimes c_7 \otimes m_{13} + c_7 \otimes m_{13} \otimes c_{14} + m_{11} \otimes m_8 \otimes c_{14} + c_7 \otimes m_{13} \otimes m_8 \\
&+ m_{11} \otimes c_7 \otimes c_{14} + m_{11} \otimes m_{13} \otimes c_{14} \\
\Delta_3(s_{11}) &= c_4 \otimes m_5 \otimes c_{15} \\
\Delta_3(s_{12}) &= c_8 \otimes m_5 \otimes c_{16}
\end{aligned}
$$

$$\Delta_3(t_1) = m_{11} \otimes c_3 \otimes c_4 + m_{11} \otimes c_4 \otimes m_{10} + c_3 \otimes c_4 \otimes m_{10} + m_{11} \otimes c_3 \otimes m_{10}$$
$$\Delta_3(t_2) = m_4 \otimes c_3 \otimes c_4 + c_3 \otimes c_4 \otimes m_5 + m_4 \otimes c_3 \otimes m_5 + m_4 \otimes c_4 \otimes m_5$$
$$\Delta_3(t_3) = m_{11} \otimes c_7 \otimes c_8 + m_{11} \otimes c_7 \otimes m_{10} + c_7 \otimes c_8 \otimes m_{10} + m_{11} \otimes c_8 \otimes m_{10}$$
$$\Delta_3(t_4) = m_4 \otimes c_7 \otimes c_8 + c_7 \otimes c_8 \otimes m_5 + m_4 \otimes c_8 \otimes m_5 + m_4 \otimes c_7 \otimes m_5$$
$$\Delta_3(t_5) = m_9 \otimes c_5 \otimes m_8 + m_9 \otimes c_5 \otimes c_6 + m_9 \otimes c_6 \otimes m_8 + c_5 \otimes c_6 \otimes m_8$$
$$\Delta_3(t_6) = c_5 \otimes c_6 \otimes m_7 + m_6 \otimes c_5 \otimes m_7 + m_6 \otimes c_6 \otimes m_7 + m_6 \otimes c_5 \otimes c_6$$
$$\Delta_3(t_7) = c_9 \otimes c_{10} \otimes m_8 + m_9 \otimes c_9 \otimes m_8 + m_9 \otimes c_9 \otimes c_{10} + m_9 \otimes c_{10} \otimes m_8$$
$$\Delta_3(t_8) = m_6 \otimes c_9 \otimes m_7 + m_6 \otimes c_9 \otimes c_{10} + c_9 \otimes c_{10} \otimes m_7 + m_6 \otimes c_{10} \otimes m_7$$

$$
\begin{aligned}
\Delta_3(q_1) &= c_1 \otimes s_3 \otimes m_8 + c_3 \otimes s_7 \otimes c_{13} + c_1 \otimes c_{15} \otimes t_5 + c_1 \otimes s_3 \otimes c_{13} \\
&+ s_3 \otimes c_6 \otimes c_{13} + c_1 \otimes t_5 \otimes c_{13} + c_3 \otimes s_7 \otimes m_8 + t_1 \otimes m_8 \otimes c_{13} \\
&+ c_1 \otimes s_3 \otimes c_6 + s_3 \otimes c_6 \otimes m_8 + t_1 \otimes c_6 \otimes m_8 + m_{11} \otimes c_3 \otimes s_7 \\
&+ t_1 \otimes m_{12} \otimes m_8 + c_{15} \otimes t_5 \otimes c_{13} + s_3 \otimes m_8 \otimes c_{13} + t_1 \otimes m_{12} \otimes c_{13} \\
&+ t_1 \otimes c_6 \otimes c_{13} + s_7 \otimes m_8 \otimes c_{13} + m_{11} \otimes s_7 \otimes m_8 + m_{11} \otimes s_7 \otimes c_{13} \\
&+ t_1 \otimes m_{12} \otimes c_6 \\
\Delta_3(q_2) &= m_{11} \otimes c_7 \otimes s_8 + t_3 \otimes m_{12} \otimes c_{14} + t_3 \otimes m_8 \otimes c_{14} + c_7 \otimes s_8 \otimes m_8 \\
&+ s_4 \otimes m_8 \otimes c_{14} + t_3 \otimes m_{12} \otimes m_8 + c_{11} \otimes c_{16} \otimes t_7 + s_8 \otimes m_8 \otimes c_{14} \\
&+ c_{11} \otimes s_4 \otimes c_{10} + t_3 \otimes m_{12} \otimes c_{10} + t_3 \otimes c_{10} \otimes m_8 + c_{11} \otimes t_7 \otimes c_{14} \\
&+ c_7 \otimes s_8 \otimes c_{14} + s_4 \otimes c_{10} \otimes c_{14} + s_4 \otimes c_{10} \otimes m_8 + c_{11} \otimes s_4 \otimes m_8 \\
&+ c_{16} \otimes t_7 \otimes c_{14} + m_{11} \otimes s_8 \otimes m_8 + t_3 \otimes c_{10} \otimes c_{14} + m_{11} \otimes s_8 \otimes c_{14} \\
&+ c_{11} \otimes s_4 \otimes c_{14} \\
\Delta_3(q_3) &= t_2 \otimes m_6 \otimes c_5 + c_{15} \otimes t_6 \otimes c_{13} + c_3 \otimes s_{11} \otimes m_3 + c_1 \otimes c_5 \otimes s_5 \\
&+ c_1 \otimes m_6 \otimes s_5 + t_2 \otimes c_5 \otimes m_3 + m_6 \otimes c_5 \otimes s_5 + t_2 \otimes c_{15} \otimes m_6 \\
&+ c_3 \otimes s_{11} \otimes c_5 + c_1 \otimes t_6 \otimes c_{13} + t_2 \otimes c_{15} \otimes m_3 + c_3 \otimes s_{11} \otimes m_6 \\
&+ m_4 \otimes c_3 \otimes s_{11} + m_4 \otimes s_{11} \otimes c_5 + m_4 \otimes s_{11} \otimes m_6 + s_{11} \otimes m_6 \otimes m_3 \\
&+ t_2 \otimes c_{15} \otimes c_5 + c_{15} \otimes m_6 \otimes s_5 + t_2 \otimes m_6 \otimes m_3 + c_1 \otimes c_{15} \otimes s_5 \\
&+ m_4 \otimes s_{11} \otimes m_3 + s_{11} \otimes m_6 \otimes c_5 + c_{15} \otimes c_5 \otimes s_5 + c_1 \otimes c_{15} \otimes t_6 \\
&+ s_{11} \otimes c_5 \otimes m_3 \\
\Delta_3(q_4) &= c_{11} \otimes t_8 \otimes c_{14} + t_4 \otimes c_{16} \otimes m_3 + c_7 \otimes s_{12} \otimes m_6 + t_4 \otimes c_{16} \otimes m_6 \\
&+ c_{16} \otimes m_6 \otimes s_6 + c_{16} \otimes c_9 \otimes s_6 + c_{11} \otimes c_{16} \otimes t_8 + s_{12} \otimes m_6 \otimes c_9 \\
&+ m_4 \otimes s_{12} \otimes m_6 + m_4 \otimes c_7 \otimes s_{12} + c_{11} \otimes c_9 \otimes s_6 + c_{11} \otimes m_6 \otimes s_6 \\
&+ t_4 \otimes m_6 \otimes c_9 + m_6 \otimes c_9 \otimes s_6 + s_{12} \otimes c_9 \otimes m_3 + m_4 \otimes s_{12} \otimes m_3 \\
&+ t_4 \otimes c_9 \otimes m_3 + m_4 \otimes s_{12} \otimes c_9 + c_7 \otimes s_{12} \otimes m_3 + t_4 \otimes m_6 \otimes m_3 \\
&+ c_{11} \otimes c_{16} \otimes s_6 + t_4 \otimes c_{16} \otimes c_9 + s_{12} \otimes m_6 \otimes m_3 + c_{16} \otimes t_8 \otimes c_{14} \\
&+ c_7 \otimes s_{12} \otimes c_9
\end{aligned}
$$

$$(\Delta_2 \otimes 1 + 1 \otimes \Delta_2)\,\Delta_2 + \nabla\,(\Delta_3) = 0$$

$$\varphi_1 \neq 0$$

At this point automated computations were insufficient to determine a Δ^3 that captured all non-bounding cycles such that $(g \otimes g \otimes g)\Delta^3 + \varphi_1$ consisted of only bounding cycles.

5.5 Determination of g^3 and Δ^3

Since the program failed to captured all non-bounding cycles, φ_1 was manually examined for possible decomposition into bounding and non-bounding cycles. Essentially, the strategy was to collect together chains whose summation could ultimately be expressed in the form $Z(C) \otimes Z(C) \otimes Z(C)$. Since working over $\mathbb{Z}_2$, an important tactic was to add the same cell or chain twice in order to fill out what otherwise might appear as an incomplete cycle. Once in this form, the cycle could be identified as either part of Δ^3 or g^3, depending on whether the cycles in the tuple components were all non-bounding cycles or not.

The following Δ^3 and g^3 were determined and then verified to satisfy the relation

$$(g \otimes g \otimes g)\Delta^3 + \varphi_1 + \nabla\left(g^3\right) = 0 \tag{5.1}$$

5.5.1 Δ^3

$$\begin{aligned}\Delta^3 = &(h1_0 \otimes h1_1 \otimes h1_2 + h1_1 \otimes h1_0 \otimes h1_2 + h1_2 \otimes h1_1 \otimes h1_0 + h1_2 \otimes h1_0 \otimes h1_1)\partial_{h2_1} \\ &+(h1_0 \otimes h1_1 \otimes h1_2 + h1_0 \otimes h1_2 \otimes h1_1 + h1_2 \otimes h1_1 \otimes h1_0 + h1_1 \otimes h1_2 \otimes h1_0)\partial_{h2_0}\end{aligned}$$

5.5.2 g^3

$$\begin{aligned}
g^3(h0_0) &= 0 \\
g^3(h1_0) &= (m_{11} + m_4) \otimes (m_4) \otimes (m_4) \\
g^3(h1_1) &= (m_4) \otimes (c_3 + c_7) \otimes (m_4 + c_3) + (c_3 + c_7) \otimes (c_3) \otimes (m_4 + c_3) \\
&\quad +(c_3 + c_7) \otimes (m_4) \otimes (m_4) \\
g^3(h1_2) &= (m_4 + c_3 + m_2) \otimes (m_6 + m_9) \otimes (m_4 + c_3 + m_2 + m_6) \\
&\quad +(m_4) \otimes (c_3 + m_2) \otimes (m_6 + m_9) + (c_3) \otimes (m_2) \otimes (m_6 + m_9) \\
&\quad +(m_6 + m_9) \otimes (m_6) \otimes (m_6) + (m_6 + m_9) \otimes (m_2 + m_6) \otimes (m_4 + c_3 + m_2) \\
&\quad +(m_6 + m_9) \otimes (c_3) \otimes (m_4 + c_3) + (m_6 + m_9) \otimes (m_4) \otimes (m_4)
\end{aligned}$$

$$
\begin{aligned}
g^3(h2_0) = {} & (t_4+t_2+t_3+t_1)\otimes(m_1+m_3+m_{12})\otimes(m_1) \\
& +(t_4+t_2+t_3+t_1)\otimes(m_{12})\otimes(m_3) \\
& +(s_5+s_6+s_7+s_8+s_9+s_{10})\otimes(m_5+m_{10})\otimes(m_1+m_3+m_{12}) \\
& +(m_4+m_{11})\otimes(s_5+s_6+s_9+s_{10})\otimes(m_1) \\
& +(s_7+s_8)\otimes(m_{10}+m_{12}+c_6)\otimes(m_5+m_{10}) \\
& +(m_4+c_3)\otimes(s_7+s_8)\otimes(m_5+m_{10})+(m_4+m_{11})\otimes(m_4)\otimes(s_9+s_{10}) \\
& +(m_4+m_{11})\otimes(c_3+c_4+m_{10}+m_{12})\otimes(s_5+s_6) \\
& +(m_4+m_{11})\otimes(c_3)\otimes(s_7+s_8) \\
& +(s_5+s_6+s_9+s_{10})\otimes(c_8+m_{13}+m_8+c_{14})\otimes(m_5+m_{10}) \\
& +(m_4+c_3+m_{13}+c_6)\otimes(s_5+s_6)\otimes(m_5+m_{10}) \\
& +(m_4+m_{11})\otimes(s_7+s_8)\otimes(m_1+c_{13}+m_7) \\
& +(m_4+m_{11})\otimes(m_4+m_{11})\otimes(s_9+s_{10})+(m_4+m_{11})\otimes(t_5+t_6)\otimes(c_7+c_3) \\
& +(m_4+m_{11})\otimes(m_8+m_7)\otimes(s_9+s_{10})+(t_5+t_6)\otimes(c_3+c_7)\otimes(m_4+m_{11}) \\
& +(m_8+m_7)\otimes(s_9+s_{10})\otimes(m_4+m_{11})+(m_8+m_7)\otimes(c_{14}+c_{13})\otimes(t_1+t_2) \\
& +(m_4+m_{11})\otimes(s_8)\otimes(c_{13}+c_{14})+(m_4+m_{11})\otimes(c_4+c_8)\otimes(s_6) \\
& +(s_8)\otimes(c_6+c_{10})\otimes(m_5+m_{10})+(c_6+c_{10})\otimes(s_8)\otimes(m_5+m_{10}) \\
& +(m_4+m_{11})\otimes(s_9+s_{10})\otimes(m_5+m_{10})+(m_4+m_{11})\otimes(c_3+c_7)\otimes(s_6+s_8) \\
& +(m_4+m_{11})\otimes(c_7+c_3)\otimes(t_5+t_6) \\
& +(s_5+s_6+s_9+s_{10})\otimes(m_6+m_9)\otimes(m_4+m_{11}) \\
& +(c_6+c_{10})\otimes(t_5+t_6)\otimes(m_4+m_{11})+(c_{10}+c_6)\otimes(m_7+m_8)\otimes(t_1+t_2)
\end{aligned}
$$

$$
\begin{aligned}
g^3(h2_1) = {} & (m_2+c_3+m_4)\otimes(t_7+t_8+t_6+t_5)\otimes(m_4+c_3+m_{13}+m_7) \\
& +(c_3+m_4)\otimes(m_2+c_3)\otimes(t_5+t_6+t_8+t_7) \\
& +(c_3)\otimes(c_3)\otimes(t_5+t_6+t_8+t_7) \\
& +(t_6+t_5+t_8+t_7)\otimes(m_4+c_3+m_{13}+m_7)\otimes(m_4+c_3+m_{13}+m_7) \\
& +(t_5+t_7+t_8+t_6)\otimes(m_4+c_3+m_{13})\otimes(m_7) \\
& +(t_5+t_8+t_7+t_6)\otimes(m_4+c_3)\otimes(m_{13})+(t_5+t_8+t_7+t_6)\otimes(m_4)\otimes(c_3) \\
& +(m_4+m_{11})\otimes(c_3+c_7)\otimes(t_5+t_6)+(m_6+m_9)\otimes(t_5+t_6)\otimes(c_{13}+c_{14}) \\
& +(m_6+m_9)\otimes(m_6+m_9)\otimes(s_9+s_{10})+(t_5+m_6)\otimes(c_5+c_9)\otimes(m_7+m_8) \\
& +(m_7+m_8)\otimes(s_1+s_2+s_9+s_{10})\otimes(m_7+m_8) \\
& +(m_4+c_3+m_2)\otimes(m_6+m_9)\otimes(s_1+s_2+s_5+s_6+s_9+s_{10}) \\
& +(m_6+m_9)\otimes(s_1+s_2+s_5+s_6+s_9+s_{10})\otimes(m_4+c_3+m_{13}+c_6+m_3) \\
& +(m_9+m_6)\otimes(m_2+c_3+m_2+m_6)\otimes(s_1+s_2+s_9+s_{10}) \\
& +(s_1+s_2+s_5+s_6+s_9+s_{10})\otimes(m_7+m_8)\otimes(m_4+c_3+m_{13}+m_7) \\
& +(m_4+c_3+m_2+m_6+c_5)\otimes(s_5+s_6)\otimes(m_7+m_8) \\
& +(m_6+m_9)\otimes(c_9)\otimes(s_5+s_6) \\
& +(s_1+s_2+s_5+s_6+s_9+s_{10})\otimes(m_3+c_{10})\otimes(m_7+m_8) \\
& +(c_6+c_{10})\otimes(s_6)\otimes(m_8+m_7)+(m_6+m_9)\otimes(m_4)\otimes(s_1+s_2+s_9+s_{10}) \\
& +(m_6+m_9)\otimes(m_2)\otimes(s_1+s_2+s_9+s_{10}) \\
& +(m_6+m_9)\otimes(c_3+c_7)\otimes(s_2+s_{10}+a_1+a_3) \\
& +(c_3+c_7)\otimes(s_2+s_{10}+a_1+a_3)\otimes(m_7+m_8) \\
& +(c_3+c_7)\otimes(m_4+m_{11})\otimes(t_5+t_6)+(m_6)\otimes(c_5+c_9)\otimes(m_7+m_8) \\
& +(m_6+m_9)\otimes(c_6+c_{10})\otimes(s_5) \\
& +(m_6+m_9)\otimes(s_2+s_6+s_{10}+a_1+a_3)\otimes(c_{13}+c_{14}) \\
& +(s_1+s_5+s_9+a_1+a_3)\otimes(c_{13}+c_{14})\otimes(m_7+m_8) \\
& +(t_6)\otimes(c_5+c_9)\otimes(m_7+m_8)+(t_5+t_6)\otimes(c_6+c_{10})\otimes(m_7+m_8) \\
& +(m_7+m_8)\otimes(s_5+s_6)\otimes(m_7+m_8)+(m_6+m_9)\otimes(t_5+t_6)\otimes(c_3+c_7) \\
& +(m_6+m_9)\otimes(m_7+m_8)\otimes(s_9+s_{10})
\end{aligned}
$$

5.6 Discussion

The program correctly computes Δ^2 and g^2, and derives a Δ_3 and Δ_4 on chains (the latter is not included here). However, the current software implementation proposes a Δ^3 that fails to capture all the non-bounding cycles present in φ_1. Due to this, the software explicitly fails when deriving g^3 since the `chain_integrate` method relies on the input being linearly dependent on a basis spanning the space of all bounding cycles in $C \otimes C \otimes C$.

One possible reason for failure of the software in this respect is due to the naive approaches used to solve Problem 3. The current algorithm only attempts to construct cycles that are immediately present, but does not use any tactics for adding in terms that may have canceled due to working in $\mathbb{Z}_2$. When manually determining g^3, frequent use was made of adding in terms in a chain twice (i.e, adding zero) in order to complete an intuitively obvious boundary cycle. How such intuitive decisions could be codified remains an open question.

Another possible limitation could be due to a potential for ambiguity in how a chain in expanded form could be factored into cycles, even if all cells are present (i.e., no canceling due to modulus). The extent to which this is a consideration is unclear.

These calculations demonstrate that an A_∞-coalgebra on homology for the link complement of the Borromean rings has a primitive Δ^2 and a non-vanishing Δ^3. Further calculations with the software show that $\varphi_2 \neq 0$, however, additional work is required to determine Δ^4 and g^4.

Chapter 6

Conclusions

Through a combination of classical linear algebraic techniques, search-based algorithms, and manually calculations we successfully applied the Transfer Algorithm to a cellular decomposition of the link complement of the Borromean rings, up to Δ^3. The result of this was consistent with the conjecture concerning A_∞-coalgebras induced in homology using the Transfer Algorithm, namely, that a complex with no pairwise linkage will yield a trivial Δ^2 and a complex with 3-fold linkage will yield a nontrivial Δ^3.

A natural continuation of this work is to further extend the induction for Δ^4, g^4, and so forth. Calculations at present show that φ_2 is non-vanishing, so either Δ^4 or g^4 must necessarily be nontrivial. This could either entail a fair amount of manual work or some improved method for determining Δ^4, e.g., developing a more robust factorization method.

Perhaps even more of interest is to apply the algorithm to a 4-component Brunnian link. Preliminary calculations using a cellular decomposition and diagonal due to Niemersheim and Umble [3, 9], successfully yield a primitive Δ^2 and a non-vanishing g^2, however, the greedy algorithm for computing a Δ_3 appears to get bogged down in local minima (a known weakness of such algorithms), resulting in an unpredictably long run time. An important question to answer in this respect is whether an admissible heuristic exists for estimating the distance to a solution for the preboundary problem in this context. If this can be answered in the positive, then it would be worthwhile evaluating the effectiveness of the A* search algorithm to this problem.

Appendix A

Formal Parser Specification

Table A.1: Lexicographic Tokens

Token Name	**Regular Expression String**
PARTIAL	`r'\\partial'`
DELTA	`r'\\Delta'`
GROUP	`r'C_{([0-9]+)}\([a-zA-Z0-9]*\)'`
IDENTIFIER	`r'[a-zA-Z0-9]+(_{[^}]+})?(\^{[^}]+})?'`
PLUS	`r'\+'`
EQUALS	`r'='`
OTIMES	`r'\\otimes'`
LBRACE	`r'\\{'`
RBRACE	`r'\\}'`
LPAREN	`r'\('`
RPAREN	`r'\)'`
COMMA	`r','`

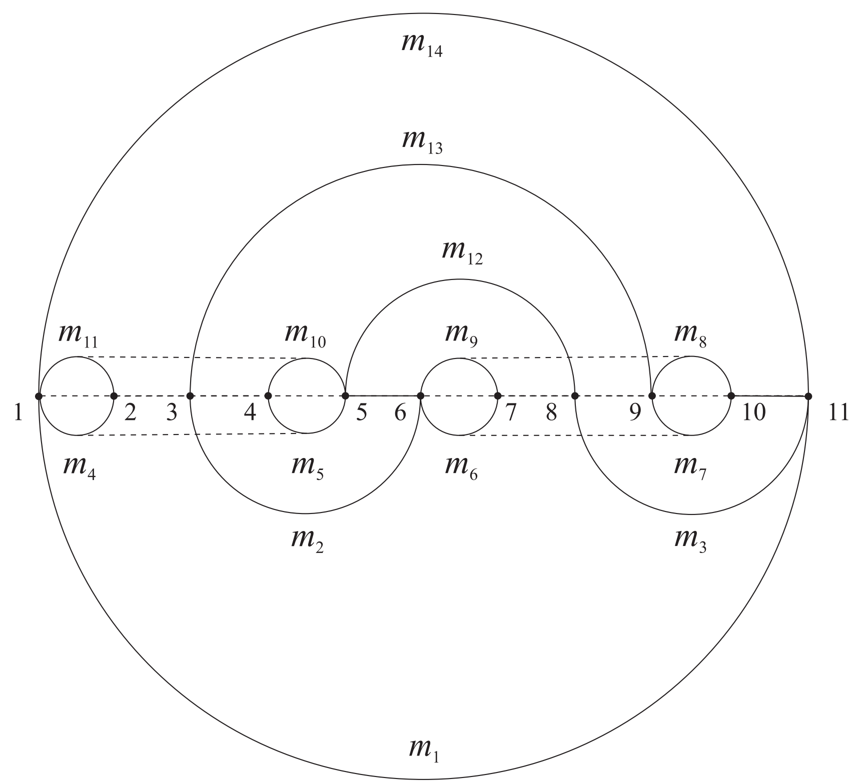
Table A.2: Formal Grammar for Chain Complex with Diagonal

S'	->	program
program	->	group_list differential_list coproduct_list
program	->	group_list differential_list
group_list	->	group_list statement_group
group_list	->	statement_group
differential_list	->	differential_list statement_differential
differential_list	->	statement_differential
coproduct_list	->	coproduct_list statement_coproduct
coproduct_list	->	statement_coproduct
statement_group	->	GROUP EQUALS LBRACE identifier_list RBRACE
statement_differential	->	PARTIAL IDENTIFIER EQUALS expression
statement_coproduct	->	DELTA IDENTIFIER EQUALS expression
identifier_list	->	IDENTIFIER
identifier_list	->	identifier_list COMMA IDENTIFIER
expression	->	expression PLUS IDENTIFIER
expression	->	expression PLUS IDENTIFIER OTIMES IDENTIFIER
expression	->	expression PLUS IDENTIFIER OTIMES LPAREN expression RPAREN
expression	->	expression PLUS LPAREN expression RPAREN OTIMES IDENTIFIER
expression	->	IDENTIFIER OTIMES IDENTIFIER
expression	->	IDENTIFIER OTIMES LPAREN expression RPAREN
expression	->	LPAREN expression RPAREN OTIMES IDENTIFIER
expression	->	LPAREN expression RPAREN OTIMES LPAREN expression RPAREN
expression	->	IDENTIFIER

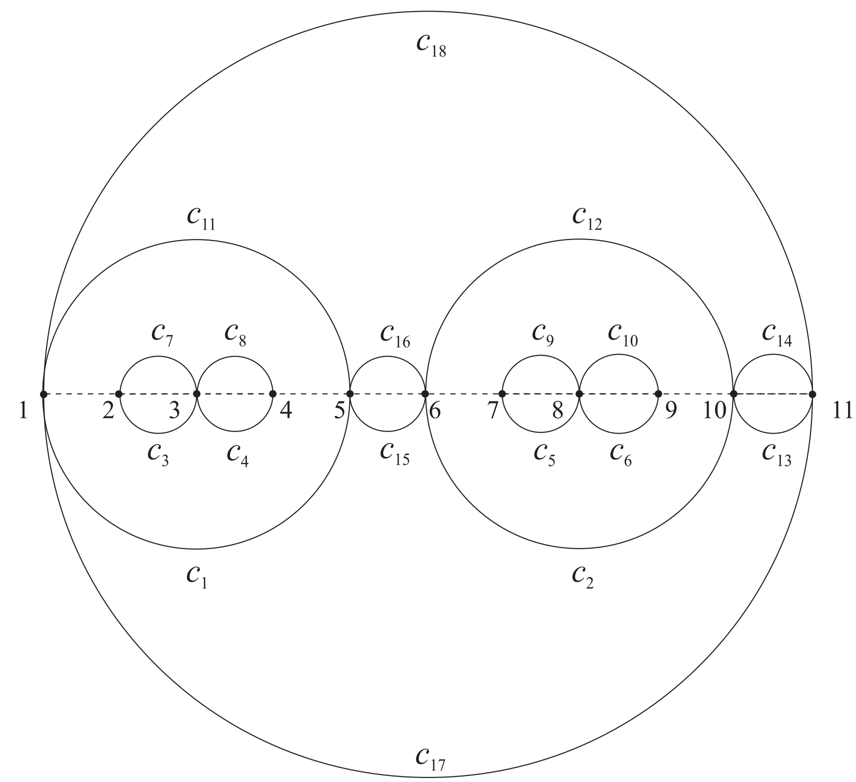
Appendix B

Cellular Decomposition of $S^3 \setminus BR$

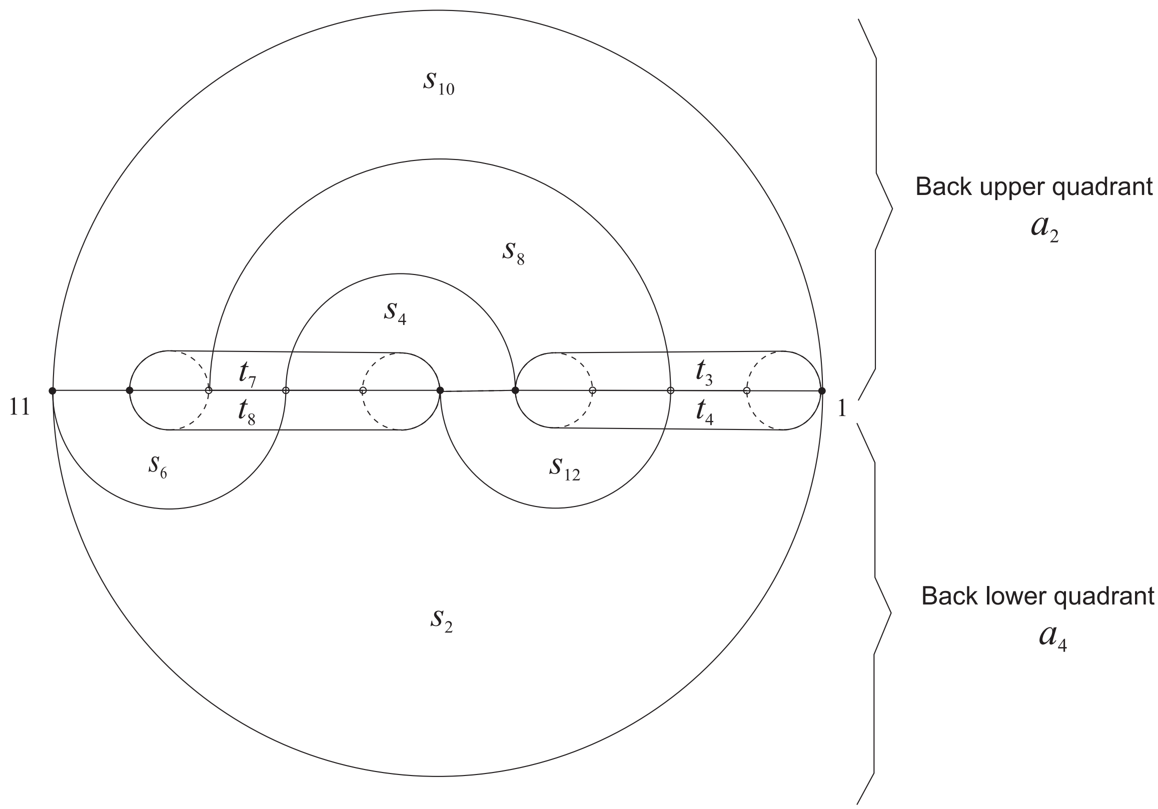
Figures in this appendix are reproduced from [10] and correspond to the cellular decomposition used for the input to the transfer algorithm.



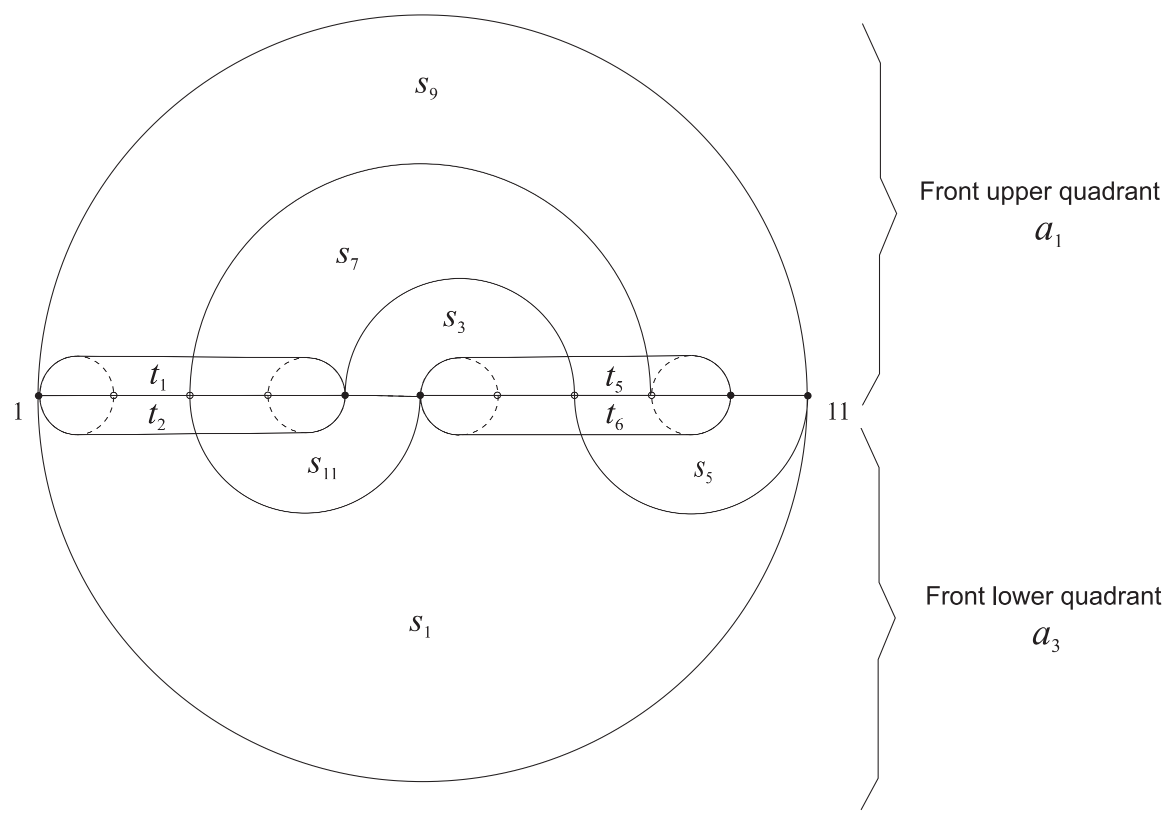
Edges and vertices (side view)



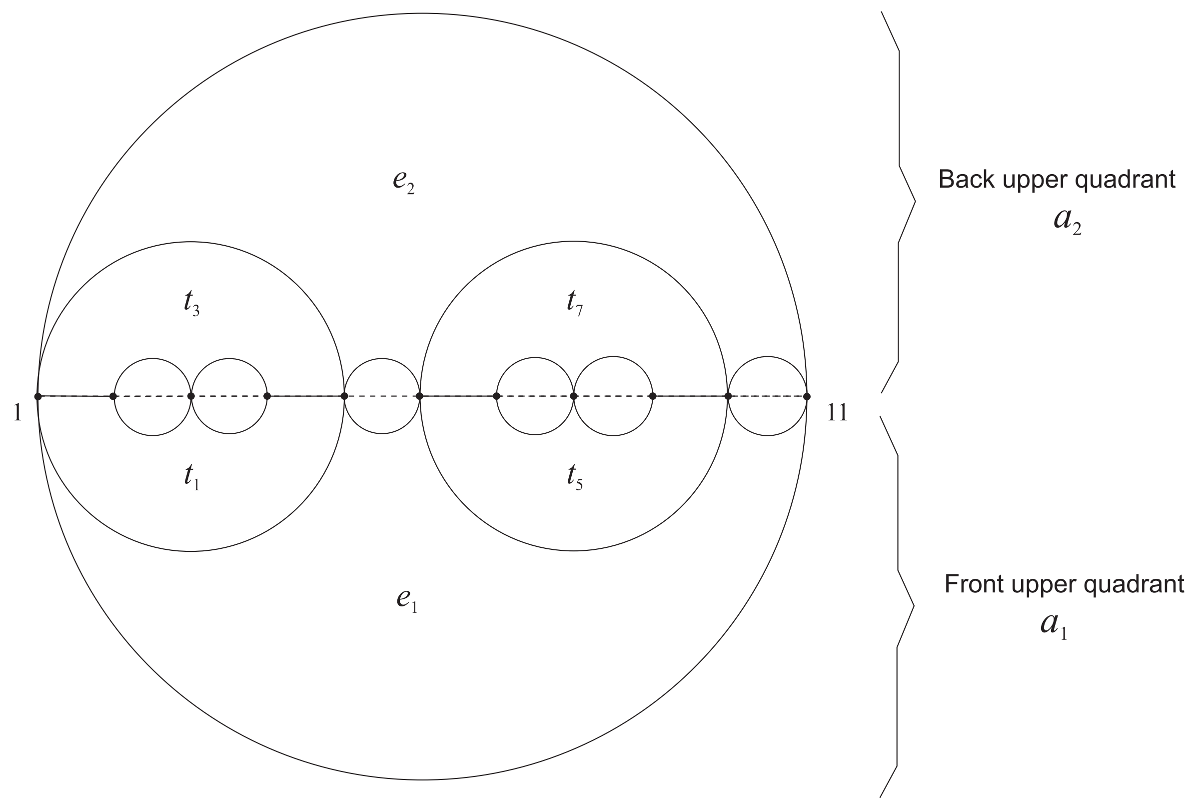
Edges and vertices (top view)



2-cells (back side view)



2-cells (front side view)



2-cells (top view)

Bibliography

[1] J.G. Dumas and C. Pernet. Computational linear algebra over finite fields. In Gary L. Mullen and Daniel Panario, editors, *Handbook of Finite Fields*, Discrete Mathematics and Its Applications, pages 514–528. Chapman & Hall / CRC, June 2013.

[2] T. Kaczynski, K. Mischaikow, and M. Mrozek. *Computational Homology*. Springer Verlag, 2004.

[3] B. Nimersheim. A Diagonal Approximation on the Link Complement of an n-component Brunnian Link in S^3. Unpublished notes, 2015.

[4] S. Saneblidze and R. Umble. Morphisms of A-infinity Bialgebras and Applications. *ArXiv e-prints*, June 2011.

[5] S. Saneblidze and R. Umble. A Non-vanishing 3-ary Operation in a Small Coassociative A_∞-Coalgebra. Unpublished private communication, 2015.

[6] J.D. Stasheff. Homotopy associativity of H-spaces I. *Transactions of the American Mathematical Society*, 108:275–292, 1963.

[7] R. Umble. A Diagonal Approximation on the Link Complement of the Borromean Rings in S^3. Unpublished notes, 2010.

[8] R. Umble. A Non-vanishing 3-ary Operation in a Small Non-Coassociative A_∞-Coalgebra. Unpublished private communication, 2016.

[9] Ron Umble, Barbara Niemersheim, and Merv Fansler. Detecting linkage in an n-component brunnian link. Talk at Bi-College Mathematics Colloquium (Haverford U.), November 2015.

[10] Umble, R. Transfering an A_∞-Coalgebra Structure on Chains to Homology. Unpublished notes, April 2016.