TWO PROBLEMS IN APPLIED TOPOLOGY

by

Nathanael Cox

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Mathematics

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Saugata Basu, Chair

Department of Mathematics and Computer Science

Dr. Andrei Gabrielov

Department of Mathematics

Dr. Hans Walther

Department of Mathematics

Dr. Tamal Dey

Department of Computer Science

Approved by:

Dr. Plamen Stefanov

Associate Head of Graduate Studies, Department of Mathematics

First and foremost, this thesis is dedicated to my wife, Alexia Cox, and my children. Without their love and support this would not have been possible.

Without the guidance and encouragement of my parents and the rest of my family, I would not be where I am and who I am today, so I dedicate this thesis to them.

Finally, I want to thank all my friends and colleagues for their continued support.

ACKNOWLEDGMENTS

Without the help of all the invaluable skills and knowledge I have learned from the many teachers and mentors I have had, I would not be where I am today, and this document would not be here. I especially want to thank my advisor, Dr. Saugata Basu, for his advice and expertise through the years. His help was invaluable.

Contents

ABSTRACT

In this thesis, we present two main results in applied topology. In our first result, we describe an algorithm for computing a semi-algebraic description of the quotient map of a proper semi-algebraic equivalence relation given as input. The complexity of the algorithm is doubly exponential in terms of the size of the polynomials describing the semi-algebraic set and equivalence relation. In our second result, we use the fact that homology groups of a simplicial complex are isomorphic to the space of harmonic chains of that complex to obtain a representative cycle for each homology class. We then establish stability results on the harmonic chain groups.

1. INTRODUCTION

In this thesis we present two results. The first result is an algorithm that makes effective in the semi-algebraic category a result due to van den Dries, [1, Theorem 10.2.15], that says that if a definable equivalence relation E on a definable set X is definably proper over X then X/E exists as a definably proper quotient of X. Our algorithm takes a semi-algebraic description of X and E and returns a semi-algebraic description of X/E.

Our second result concerns persistent homology. We explore the space of the harmonic chains on a simplicial complex K. We expand on ideas introduced by Beno Eckmann [2]. We use the fact that the homology groups of K are isomorphic to the space of harmonic chains to address the problem of determining a representative cycle for a homology class. Harmonic chains give a natural canonical way to define representative cycles of homology classes. We develop a theoretical framework to study "harmonic persistent homology subspaces." We define a distance on these chain groups and show that the distance is stable with respect to the functions that generate the filtration on the simplicial complex. Finally, we show that this stability is implied by the traditional bottleneck stability used for persistence diagrams.

These two sets of results, while seemingly coming from different fields, have motivations arising from topological data analysis. Topological Data Analysis is a rich field of interesting problems with applications in mathematics, computer science, medical imaging, astronomy and more. A brief introduction to basic tools and applications of Topological data analysis can be found in [3].

1.1 Effective Quotient Algorithm

Given a topological space X and an equivalence relation $E \subset X \times X$ on X, the quotient space of X by E, denoted X/E, is an important object of study in topology and algebraic geometry.

However, quotients do not always behave nicely with their respective morphisms. It is a well-known fact from elementary topology that quotients of Hausdorff spaces need not remain Hausdorff. Similarly, simply connectedness, contractibility, and local compactness are all properties that are not preserved in general under taking quotients. Hence, it is an important problem to determine under what conditions a quotient space inherits a given property from the original topological space. In the Hausdorff case, for example, we know that the quotient space remains Hausdorff if and only if the kernel of the quotient map is closed. A more substantial example from algebraic geometry is presented by Frances Kirwan in [4]. We consider the action of $SL(2)$ on complex projective space $\mathbb{P}^n$, where we identify $\mathbb{P}^n$ with the space of binary forms of degree n. Equivalently, consider the unordered sets of n points on the projective line $\mathbb{P}^1$. The orbit in which all n points coincide is contained in the closure of every other orbit, and hence the topological quotient cannot be given the structure of a projective variety. Once again, we have a situation where the quotient space does not in general inherit the desirable properties of the original space (in this case the property is "being a variety"). To obtain a quotient space in this above example requires determining which orbits are "bad" and omitting them (see the following introduction to Geometric Invariant Theory due to Mumford for more information: [5]).

Our first result was motivated by prior work on Reeb spaces [6]. A Reeb graph is a mathematical object first introduced by Georges Reeb in [7] as a tool in Morse Theory. Given a topological space X and a function $f : X \to \mathbb{R}$, we define an equivalence relation E on X as follows: $(x_1, x_2) \in E$ if and only if $f(x_1) = y = f(x_2)$ and x_1, x_2 are in the same connected component of $f^{-1}(y)$. The quotient space X/E is called the Reeb graph of f and is denoted Reeb$f$. A natural generalization of the Reeb graph is to no longer require $\mathrm{im}(f) \subset \mathbb{R}$. If instead $f : X \to Y$, where Y is an arbitrary topological space, using a similarly defined equivalence relation, we refer to $X/\sim$ as the Reeb space of f, still denoted Reeb$f$.

Reeb spaces have been studied from both a theoretical and practical perspective. Mapper, a concept introduced in [8], gives a discrete approximation of the Reeb space of a multivariate mapping, allowing for more efficient computation of the underlying data structure. Munch et. al. [9] define the *interleaving distance* for Reeb spaces to show the convergence between the Reeb space and Mapper. Reeb spaces are often used in 3D graphics applications such as data skeletonization, shape comparison and surface denoising, see [10], [11], and [12]; respectively. In [6] and [13], a bound on the first betti numbers of the reeb space in terms of the defining space X is presented. This result is extended here, [14], to higher betti numbers.

Reeb spaces have been shown to be of great interest, and it was shown in [6, Theorem 1] that if f is a definable map, then Reeb$f$ can be realized as a definably proper map. In fact, [1, Theorem 10.2.15] proves an entire class of quotients that inherit the property of definablity from their originating space. Whenever the equivalence relation E is definable and definably proper over X, then X/E will exist as a definably proper quotient of X. For this reason, we expand our focus beyond just Reeb spaces and topological data analysis. We focus more broadly on algorithmic semi-algebraic geometry to develop an algorithm to produce quotients of this more general class of equivalence relations.

Algorithmic semi-algebraic geometry is a vast field in its own right. Looking at this problem from that perspective is also motivation enough to solve this problem. Semi-algebraic algorithms use first-order formulae in the language of the reals and apply "a computational procedure that takes an input and after a finite number of allowed operations produces an output" [15]. Prior results in this field include a quantifier-elimination algorithm, which takes a formula with quantifiers $(\exists, \forall)$ and returns a quantifier free formula. This algorithm is used to determine the validity of a formula in the language. The best known complexity of quantifier elimination is doubly exponential in the number of quantifier alternations. However, if this number is fixed, the algorithm becomes singly exponential. Another algorithm of interest and applicability produces a semi-algebraic description of a triangulation of a semi-algebraic set. Triangulated spaces tend to have nice geometry and are easier to work with. The most efficient triangulation algorithms rely on cylindrical decomposition algorithms which are intrinsically doubly exponential. It is an open problem of some interest to determine a singly exponential algorithm for triangulation. We rely on these two algorithms frequently in our main result, however there are many more useful and powerful algorithms that have been produced. The interested reader should see [15] for an introduction to this area of study and for a much more comprehensive list of results. However, a general quotienting algorithm has yet to be developed. As quotients are of considerable interest, we choose to pursue a more general quotient algorithm in our paper.

There is a "meta theorem" in algorithmic semi-algebraic geometry that upper bounds on topological complexity of objects are closely related to the worst case scenario complexity of algorithms computing topological invariants of such objects. Hence, [6, Theorem 1] hints at the possibility that an algorithm to solve our problem exists that runs in singly-exponential time. The technique in that paper is not suitable for this purpose, because the upper bound comes from approximating the Reeb space as the E_1-term of a spectral sequence converging to it. This is not sufficient to describe the Reeb space itself unfortunately. Making effective the steps of van den Dries's proof, we use semi-algebraic triangulation to obtain the general quotient space. While the geometric nature of Reeb spaces could allow for more efficient computation using other methods, in this paper the lack of an existing general quotient algorithm in semi-algebraic geometry motivates us to sacrifice potential efficiency for broader applicability.

If the polynomials defining a semi-algebraic formula have coefficients in a domain D, the complexity of an algorithm involving those polynomials is the number of arithmetic operations that must be performed in D. For a general domain, these operations are addition, subtraction, and multiplication. Over a real closed field $\mathbf{R}$, the operations also include division by nonzero elements and comparing elements by the natural ordering $<$ on $\mathbf{R}$. The number of operations that are necessary tends to scale with the number of polynomials, their degree bound, and the number of variables.

Presented formally, our first result is the following [16]:

Theorem 1. *Let $\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m]$ and $\mathcal{P}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$. Given a $\mathcal{P}_1$-formula Φ_X, whose realization is a semi-algebraic set X, and a $\mathcal{P}_2$-formula Φ_E, whose realization is a proper semi-algebraic equivalence relation $E \subset X \times X$, then there exist formulas Φ_f, describing the graph of the map from X to the semi-algebraic realization of X/E, and $\Phi_{X/E}$, representing the semi-algebraic realization of X/E.*
Moreover, there is an algorithm to determine these formulas. Let $k_1 = |\mathcal{P}_1|$, $d_1 \geq \deg(P)$ for all $P \in \mathcal{P}_1$, $k_2 = |\mathcal{P}_2|$ and $d_2 \geq \deg(P)$ for all $P \in \mathcal{P}_2$. This algorithm has complexity

$$\begin{cases} (mkd)^{2^{\mathcal{O}(m^3)}} & \text{if } k_1^{2^{\mathcal{O}(m)}} \approx k_2 (\approx k) \text{ and } d_1^{2^{\mathcal{O}(m)}} \approx d_2 (\approx d) \\ (mk_2d_2)^{2^{\mathcal{O}(m^3)}} & \text{if } k_2 >> k_1^{2^{\mathcal{O}(m^3)}} \text{ and } d_2 >> d_1^{2^{\mathcal{O}(m)}} \\ (mk_1d_1)^{2^{\mathcal{O}(m^4)}} & \text{if } k_1^{2^{\mathcal{O}(m)}} >> k_2 \text{ and } d_1^{2^{\mathcal{O}(m)}} >> d_2 \end{cases}$$

1.2 Harmonic Chains

In chapter 3, we explore a different aspect of topological data analysis. Persistent homology is a way to study how topological features change over time and to observe which topological features "persist." This allows the user a means of determining which features are more important/intrinsic to the space being studied. Based on the user's application, persistent homology is a valuable tool to handle noise in data. For an introduction to persistent homology, with an eye toward computational applications, we direct the reader to [17].

Given a filtration $\mathcal{F}$ on a topological space X, the p-th persistent homology groups of X, denoted $H_p^{s,t}(X) = \mathrm{im}(\mathrm{i}_p^{s,t})$, where $\mathrm{i}_p^{s,t} : H_p(X_s) \to H_p(X_t)$ is the map induced by the inclusion $\mathrm{i}^{s,t} : X_s \to X_t$. X_s is the s-th space in the filtration $\mathcal{F}$ of X. The topological features, in this case homology classes, that exist through multiple stages of the filtration are said to persist. The user can filter out the classes that don't persist long enough through the filtrations, discarding them as noise.

We consider persistent homology on a simplicial complex K with coefficents in a field k. In this case all the homology groups will be k-vector spaces. The barcode of a filtration on K encodes the information about how long homology classes persist. A new homology class that is "born" at a given time is defined modulo a certain subspace in the homology of the complex. This quotienting makes identifying a given bar of the barcode with a particular homology class problematic. It has become an important problem in applications of persistent homology to obtain a representative cycle for a given homology class. In applications, the simplices of K often have special significance. As such one would then want to determine how the these features are represented in the original data. See [18] and [19] for examples of applications where simply knowing the persistence diagram was not enough; a representative cycle was desired.

In this chapter, we briefly survey previous attempts to determine a representative cycle before proposing our own method. We use the fact that homology groups are isomorphic to the space of harmonic chains (chains on simplicial complexes whose boundary and coboundary are both 0). Harmonic chains have a natural inner product that we use to determine a cycle to represent each class as it is born. If the filtration is simplex-wise (i.e. only one simplex is added at a time) then at most one cycle is either born or dies at any given time. We are able to define a space of harmonic chains that are born at time s and die at time t. When the filtration is simplex wise, this subspace is 1-dimensional. Therefore, up to multiplication by a constant, we have a single chain that represents this cycle. See 3.3.1.

If $f : K \to \mathbb{R}$ induces a simplex-wise filtration, f also induces what we call a "persistence function" of dimension p, definition 23. For every p, $F : \mathbb{R} \to \amalg_d Gr(d, C_p(K))$ such that F maps a real number t to the set of harmonic chains of the simplicial complex $f^{-1}((-\infty, t])$.

By developing this theory, we provide a natural, canonical method of determining a representative cycle of the persistent homology cycles of a simplicial complex. In addition to developing the definitions, we have the following stability results that are reminiscent of prior stability results [20]:

Theorem 2. *Let K be a finite simplicial complex. For each $p \geq 0$, there exists a c depending only on K such that such that if F, G are persistence functions induced f, g, respectively, then $d_{\mathfrak{h}}(F, G) \leq c \cdot |f - g|$, where $|\cdot|$ is the L_∞ norm.*

Here $d_{\mathfrak{h}}(F, G)$ is a metric on the space of p-dimensional persistence functions, see equation 3.3.4. We also prove the following association between persistence functions and persistence diagrams.

Theorem 3. *Each persistence function of dimension p can be associated with a distinct p-dimensional barcode.*

We finally prove a slightly different stability result relating the harmonic distance to the traditional bottleneck distance on persistence diagrams.

Theorem 4. *Let K be a finite simplicial complex. For each $p \geq 0$, there exists a $c = c(K)$ depending only on K such that if F and G are persistence functions of dimension p then $d_{\mathfrak{h}}(F, G) \leq c \cdot W_\infty(\mathrm{Dgm}_p(F), \mathrm{Dgm}_p(G))$.*

Here, $Dgm_p(F)$ is the persistence barcode obtained from Theorem 3, with the diagonal added with infinite multiplicity, and W_∞ is the bottleneck distance between persistence diagrams.

2. EFFECTIVE DEFINABLE QUOTIENT ALGORITHM

2.1 Background and Preliminary Results

This chapter of the thesis is devoted to developing an algorithm which proves Theorem 1. Our algorithm makes effective in the semi-algebraic category the following theorem due to van den Dries:

Theorem 5. *[1, Theorem 10.2.15] Suppose a definable equivalence relation E on a definable set X is definably proper over X. Then X/E exists as a definably proper quotient of X.*

We begin this chapter by providing background definitions and prior results that are needed for our algorithm.

We begin with o-minimal structures, a set of "nice" subsets of real closed fields. For details beyond what are mentioned here, the interested reader should consult [1], [21] for a broad introduction to the theory of o-minimal structures.

Definition 1. *An* **o-minimal structure** *on a real closed field $(\mathbf{R}, <)$ is a sequence $\mathcal{S} = (\mathcal{S})_{m \in \mathbf{N}}$ such that for each $m \geq 0$:*

- *(i) $\mathcal{S}_m$ is a Boolean algebra of subsets of $\mathbf{R}^m$.*
- *(ii) If $A \in \mathcal{S}_m$, then $A \times \mathbf{R}, \mathbf{R} \times A \in \mathcal{S}_{m+1}$.*
- *(iii) The set $\{(x_1, \ldots, x_m) \in \mathbf{R}^m | x_1 = x_m\}$ is in $\mathcal{S}_m$.*
- *(iv) If $A \in \mathcal{S}_{m+1}$ and $\boldsymbol{\pi} : \mathbf{R}^{m+1} \to \mathbf{R}^m$ is the projection onto the first m coordinates, then $\boldsymbol{\pi}(A) \in \mathcal{S}_m$.*
- *(v) The set $\{(x, y) \in \mathbf{R}^2 | x < y\} \in \mathcal{S}_2$.*
- *(vi) The sets in $\mathcal{S}_1$ are exactly the finite unions of intervals and points.*

If $A \in \mathcal{S}$, we say A is a **definable set***. A map f is called* **definable** *if the graph of f is a definable set.*

Examples of o-minimal structures on the real line include semi-linear sets, semi-algebraic sets, and sub-analytic sets. The semi-algebraic sets, subsets of $\mathbf{R}^m$ defined by polynomial equalities and inequalities, are of particular interest to us. Semi-algebraic sets can be described as realizations of first order formulae in the language of the reals. We go into more detail about such formulae below, but first we need a few more definitions from the theory of o-minimal structures. We first consider quotients in the category of definable sets.

Definition 2. *Given a set X, an equivalence relation $E \subset X \times X$ on X is a* **definable equivalence relation** *if E is a definable set. Furthermore, E is* **definably proper** *if either $p_1 : E \to X$ or $p_2 : E \to X$ is a definably proper map, where p_1 and p_2 represent the restriction to E of the two projections from $X \times X \to X$.*

Given a map $f : X \to Y$, between definable sets X and Y, we define an equivalence relation $E_f = \{(x, y) \in X \times X | f(x) = f(y)\}$. E_f is a definable equivalence relation. Moreover, if f is continuous E_f is closed in $X \times X$.

Definition 3. *Given a definable equivalence relation E on a definable set X, a* **definable quotient of** X **by** E *is a pair (p, Y) consisting of a definable set Y and a definable continuous surjective map $p : X \to Y$ such that:*

(i) $E = E_p$, *i.e.* $(x_1, x_2) \in E$ *if and only if* $p(x_1) = p(x_2)$ *for all* $x_1, x_2 \in X$

(ii) p *is "definably identifying": for all definable $K \subset Y$, if $p^{-1}(K)$ is closed in X, then K is closed in Y.*

If p is definably proper, instead of simply definably identifying, then we say that (p, Y) is a definably proper quotient of X by E. Given a definable quotient (p, Y) of X by E, Y is unique up to definable isomorphism. We write $Y = X/E$ and say X/E is the definable quotient of X by E. In our next definition, we relate definable quotients to semi-algebraic sets.

Definition 4. *Given an equivalence relation E, a semi-algebraic map $f : X \to Y$ is a* **map to the semi-algebraic realization of** X/E *if the following diagram commutes:*

$$
\begin{tikzcd}
X \arrow[d, "q"] \arrow[dr, "f"] & \\
X/E \arrow[r, "h"] & Y
\end{tikzcd}
$$

where q is the standard quotient map, h is a homeomorphism, and Y is a semi-algebraic set. We refer to Y in this case as **the semi-algebraic realization of** X/E*.*

We have two more concepts from o-minimal structures that we rely on in our algorithm: disjoint sums and completions.

Definition 5. *A* **disjoint sum** *of definable sets $S_1 \subset \mathbf{R}^{m_1}, \ldots, S_k \subset \mathbf{R}^{m_k}$ is a tuple $(h_1, \ldots, h_k, T)$ consisting of a definable set $T \subset \mathbf{R}^n$, for some n, and definable maps $h_\mathrm{i} : S_\mathrm{i} \to T$ such that :*

(i) h_i is a homeomorphism onto $h_\mathrm{i}(S_\mathrm{i})$ and $h_\mathrm{i}(S_\mathrm{i})$ is open in T, for $\mathrm{i} = 1, \ldots, k$

(ii) T is the disjoint union of the sets $h_1(S_1), \ldots, h_k(S_k)$

Remark 1. Let $n = 1 + \max\{m_\mathrm{i} | 1 \leq \mathrm{i} \leq k\}$ and $h_\mathrm{i} : S_\mathrm{i} \to \mathbf{R}^n$ by $h_\mathrm{i}(x) = (x, \underbrace{\mathrm{i}, \ldots, \mathrm{i}}_{n-m_\mathrm{i}})$. Then $(h_1, \ldots, h_k, \cup_\mathrm{i} h_\mathrm{i}(S_\mathrm{i}))$ is clearly a disjoint sum of $S_1, \ldots, S_k$. A disjoint sum is unique up to isomorphism, so we use the above representation for our disjoint sums. We write $S_1 \amalg \cdots \amalg S_k$ for T, and we identify S_i with its image in $S_1 \amalg \cdots \amalg S_k$ via h_i.

Now that we have the concept of a disjoint sum in the definable category, we would like to construct quotients on these sums. We take definable sets $X \subset \mathbf{R}^m$ and $Y \subset \mathbf{R}^n$, with a definable map $f : A \to Y$ for some definable $A \subset X$. We would like to describe the quotient space obtained by attaching X to Y via f. Let $\Delta(X)$ and $\Delta(Y)$ denote the diagonals of X and Y, respectively. Then

$$E(f) = \Delta(X) \cup \Delta(Y) \cup \{(a, f(a)) | a \in A\}$$

$$\cup \{(f(a), a) | a \in A\} \cup \{(a_1, a_2) \in A \times A | f(a_1) = f(a_2)\}$$

is the smallest equivalence relation on $X \amalg Y$ such that each $a \in A$ is equivalent to $f(a) \in Y$. If the definable quotient of $X \amalg Y$ by $E(f)$ exists (the quotient exists if $E(f)$ is definably proper over X), we denote it by $X \amalg_f Y$.

Definition 6. *A* **completion** *of a definable set $S \subset \mathbf{R}^m$ is a pair $(h, S\text{j})$ consisting of a closed and bounded definable set $S^* \subset \mathbf{R}^n$ (for some n) and a definable map $h : S \to S^*$ such that h is a homeomorphism from S onto $h(S)$ and $h(S)$ is dense in S^*.*

Informally we say $h : S \to S^*$ is a completion of S. Note that completions for a definable set always exist and they are not necessarily unique.

Definition 7. *Given $f : S \to T$, a definable continuous map between definable sets $S \subset \mathbf{R}^m$ and $T \subset \mathbf{R}^n$, a* **completion of** $f : S \to T$ *is a triple consisting of a completion $\mathrm{i} : S \to S^*$ of S and $\mathrm{j} : T \to T^*$ and a definable continuous map $f^* : S^* \to T^*$ such that $f^* \circ \mathrm{i} = \mathrm{j} \circ f$. In other words, we obtain the following commutative diagram, which we call a* **completion diagram of** $f : S \to T$*:*

$$\begin{tikzcd} S \arrow[r, "\mathrm{i}"] \arrow[d, "f"] & S^* \arrow[d, "f^*"] \\ T \arrow[r, "\mathrm{j}"] & T^* \end{tikzcd}$$

As with completions of a set, completion diagrams always exist.

As mentioned before, our algorithm relies on semi-algebraic triangulation to obtain our desired output. Here we discuss basic simplicial complex definitions necessary for triangulation.

Definition 8. *Let $a_0, \ldots, a_k \in \mathbf{R}^n$ be an affine independent tuple. We say*

$$(a_0, \ldots, a_k) = \left\{ \sum t_\mathrm{i} a_\mathrm{i} | t_\mathrm{i} \geq 0, \sum t_\mathrm{i} = 1 \right\} \subset \mathbf{R}^n$$

*is a k-***simplex** *in $\mathbf{R}^n$. In the case where $k = n-1$ and $a_0 = (1, 0, \ldots, 0), a_1 = (0, 1, 0, \ldots, 0), \ldots, a_{n-1} = (0, \ldots, 0, 1)$, we say that $(a_0, \ldots, a_{n-1})$ is the* **standard** $n-1$**-simplex** *in $\mathbf{R}^n$. The standard $n-1$ simplex is homeomorphic to an arbitrary $n-1$ simplex, and any k-simplex in $\mathbf{R}^n$ can be embedded in the standard $n-1$ simplex.*
A **face** *of $(a_0, \ldots, a_k)$ is any simplex spanned by a nonempty subset of $\{a_0, \ldots, a_k\}$.*

Definition 9. *A* **simplicial complex** *K in $\mathbf{R}^n$ is a finite collection K of simplicies in $\mathbf{R}^n$ such that for all simplicies $\sigma = (a_0, \ldots, a_k)$ and $\tau = (b_0, \ldots, b_l)$ of K either $\sigma \cap \tau = \emptyset$ or $\sigma \cap \tau = \gamma \in K$.*

Triangulations can be defined more generally, but for our purposes it is sufficient just to consider definable triangulations.

Definition 10. *Let $X \subset \mathbf{R}^m$ be a definable set. A* **triangulation** *in $\mathbf{R}^n$ of X is a pair (Φ, K) consisting of a complex K in $\mathbf{R}^n$ and a definable homeomorphism $\Phi : X \to |K|$, where $|K|$ is the union of the simplices of K in $\mathbf{R}^n$. Without loss of generality, we may assume that K is a subset of the standard $n-1$ simplex.*

Consider $\Phi^{-1}(K) = \{\Phi^{-1}(\sigma)|\sigma \in K\}$, a partition of X. Given a subset $A \subset X$, $(\Phi|_A, \Phi|_A(A))$ is a triangulation of A in $\mathbf{R}^n$ if A is is a union of elements of $\Phi^{-1}(K)$.

We finish our preliminary definitions by defining first order formulae in the language of the reals.

Definition 11. *We begin by defining formulae and the set of free variables of those formulae:*

- *$P = 0$ and $P \neq 0$, for $P \in \mathbf{R}[X_1, \ldots, X_m]$ are formulae with set free variables $Free(P = 0) = Free(P \neq 0) = \{X_1, \ldots, X_m\}$.*

- *If Φ_1 and Φ_2 are formulae, then $\Phi_1 \wedge \Phi_2$ and $\Phi_1 \vee \Phi_2$ are formulae with $Free(\Phi_1 \vee \Phi_2) = Free(\Phi_1 \wedge \Phi_2) = Free(\Phi_1) \cup Free(\Phi_2)$*

- *If Φ is a formula, then $\neg\Phi$ is a formula with $Free(\neg\Phi) = Free(\Phi)$*

- *If Φ is a formula and $X \in Free(\Phi)$, then $(\exists X)\Phi$ and $(\forall X)\Phi$ are formulae with $Free((\exists X)\Phi) = Free((\forall X)\Phi) = Free(\Phi) \setminus \{X\}$.*

A formula is **quantifier free** *if no quantifiers, neither $\exists$ nor $\forall$, appear.*
The **realization of a formula** *Φ with $Free(\Phi) = \{Y_1, \ldots, Y_m\}$ is the set $Reali(\Phi) = \{y \in \mathbf{R}^m|\Phi(y)$ is true$\}$.*
Two formulae Φ and Ψ such that $Free(\Phi) = Free(\Psi)$ are **equivalent** *if $Reali(\Phi) = Reali(\Psi)$.*

Definition 12. *We say a formula is written in* **prenex normal form** *if it is of the form*

$$(Q_1 X_1) \cdots (Q_m X_m) \Psi(X_1, \ldots, X_m, Y_1, \ldots, Y_k),$$

where $Q_{\mathrm{i}} \in \{\exists, \forall\}$ and Ψ is a quantifer free formula. All formulae are equivalent to a formula written in prenex normal form.

For a more in-depth explanation of formulae and logic, see Introduction to Mathematical Logic by Elliott Mendelson [22].

With the definitions out of the way, we finish this section by introducing some common formulae and shorthand that will be used in the presented algorithms.

(1.) Given a formula Φ_X, whose realization is a semi-algebraic set X, set

$$\widetilde{\Phi_{\mathrm{cl}(X)}}(Y) \leftarrow \forall \epsilon > 0 \exists X (\Phi_X(X) \wedge ||X - Y||^2 < \epsilon^2),$$

a formula whose realization is the semi-algebraic closure of X. We will let $\Phi_{\mathrm{cl}(X)}(Y)$ denote an equivalent quantifier free formula describing the semi-algebraic closure of X.

(2.) For each i, set $\Phi_{\Delta_{\mathrm{i}}}(\lambda_0, \ldots, \lambda_{\mathrm{i}}) \leftarrow \sum \lambda_{\mathrm{j}} = 1 \wedge \bigwedge_{\mathrm{j}=0}^{\mathrm{i}} \lambda_{\mathrm{j}} > 0$ a formula whose realization is the standard i-simplex.

(3.) Given formulae Φ_X and Φ_Y, whose realizations are semi-algebraic sets X and Y with $X \subset Y$, let

$$\Phi_{d_{X,Y}}(Y, t) \leftarrow \forall X [\Phi_X(X) \wedge \Phi_Y(Y)$$

$$\Rightarrow ||X - Y||^2 \geq t^2] \wedge [\exists X' \Phi_X(X') \wedge ||X' - Y||^2 = t^2]$$

be a formula whose realization is the graph of the distance function of all elements in Y from the set X. We will need this formula written in prenex normal form:

$$\Phi_{d_{X,Y}}(Y, t) = \exists X' \forall X [\Phi_X(X) \wedge \Phi_Y(Y)$$

$$\Rightarrow ||X - Y||^2 \geq t^2 \wedge \Phi_X(X') \wedge ||X' - Y||^2 = t^2].$$

(4.) Throughout we use the following convention: Given a formula in prenex normal form $\Phi_X(X)$, we let $M_X(X, Z_1, \ldots, Z_n)$ denote the **quantifier free portion of the formula** Φ_X, where $Z_1, \ldots, Z_n$ are the quantified variables of Φ_X. In other words,

$$\Phi_X(X) = Q_1 Z_1, \ldots, Q_n Z_n M_X(X, Z_1, \ldots, Z_n),$$

where $Q_\mathrm{i} \in \{\exists, \forall\}$.

In our algorithm, we use several well known algorithms in algorithmic algebraic geometry. We describe effective quantifier elimination and semi-algebraic triangulation below. The complexity of quantifier elimination is doubly exponential in terms of the number of quantifier alternations in the formula. However, if the number of alternations is fixed, the complexity becomes singly exponential. Semi-algebraic triangulation is doubly exponential in terms of the input variables. Algorithmic algebraic geometry relies heavily on these two algorithms, so in general one expects algorithms of these complexity. The most efficient method of performing semi-algebraic triangulation requires a cylindrical decomposition of the input formula. Cylindrical Decomposition is intrinsically doubly exponential in the number of variables, so one could not expect better from triangulation. It is an open problem of considerable interest to obtain an algorithm to triangulate a space in singly exponential time.

2.1.1 Effective Quantifier Elimination

Given a formula Φ, an important question to ask is whether or not we obtain a quantifier free formula Ψ that is equivalent to Φ. We then want to know, if such a Ψ exists, is there an algorithmic way to determine Ψ. If Φ is equivalent to a quantifier free formula, we say that the realization of Φ is a constructible set. In complete generality, not every formula is constructible. However the theory of real closed fields admits quantifier elimination in the language of ordered fields as shown below:

Theorem 6. *[15, Theorem 2.77] Let $\Phi(Y)$ be a formula in the language of ordered fields with coefficients in an ordered ring D contained in the real closed field* $\mathbf{R}$*. Then there is a* *quantifier free formula $\Psi(Y)$ with coefficients in D such that for every $y \in \mathbf{R}^k$, the formula $\Phi(y)$ is true if and only if the formula $\Psi(y)$ is true.*

In particular, this theorem shows that every formula defined in terms of polynomial equalities and inequalities with coefficients in $\mathbf{R}$ is equivalent to a quantifier free formula. The next challenge is to obtain an algorithm which produces a quantifier free formula given a quantified formula. This method is described in Algorithm 14.5 of [15]. The result is described in the following theorem.

Theorem 7. *[15, Theorem 14.16] Let $\mathcal{P}$ be a set of at most k polynomials each of degree at most d in $n+m$ variables with coefficients in a real closed field $\mathbf{R}$, and let Π denote a partition of the list of variables $(X_1, \ldots, X_n)$ into blocks, $X_{[1]}, \ldots, X_{[\omega]}$, where each block $X_{[\mathrm{i}]}$ has size n_{i} for $1 \leq \mathrm{i} \leq \omega$. Given $\Phi(Y)$, a $(\mathcal{P}, \Pi)$-formula, there exists a quantifier free formula*

$$\Psi(Y) = \bigvee_{\mathrm{i}=1}^{I} \bigwedge_{\mathrm{j}=1}^{J_{\mathrm{i}}} \left(\bigvee_{k=1}^{K_{\mathrm{i},\mathrm{j}}} sign(P_{\mathrm{ij}k}(Y)) = \sigma_{\mathrm{ij}k} \right)$$

where $P_{\mathrm{ij}k}(Y)$ are polynomials in the variables Y, $\sigma_{\mathrm{ij}k} \in \{0, 1, -1\}$,

$$\begin{aligned} I &\leq s^{(n_\omega+1)\cdots(n_1+1)(m+1)} d^{\mathcal{O}(n_\omega)\cdots\mathcal{O}(n_1)\mathcal{O}(m)}, \\ J_{\mathrm{i}} &\leq s^{(n_\omega+1)\cdots(n_1+1)(m+1)} d^{\mathcal{O}(n_\omega)\cdots\mathcal{O}(n_1)}, \\ K_{\mathrm{ij}} &\leq d^{\mathcal{O}(n_\omega)\cdots\mathcal{O}(n_1)}, \end{aligned}$$

and the degrees of the polynomials $P_{\mathrm{ij}k}(y)$ are bounded by $d^{\mathcal{O}(n_\omega)\cdots\mathcal{O}(n_1)}$. Morever, there is an algorithm to compute $\Psi(Y)$ with complexity

$$k^{(n_\omega+1)\cdots(n_1+1)(m+1)} d^{\mathcal{O}(n_\omega)\cdots\mathcal{O}(n_1)\mathcal{O}(m)}$$

in D, where D is the ring generated by the coefficients of P.

We use this result extensively in the algorithm that proves our main theorem.

2.1.2 Semi-Algebraic Triangulation Algorithm

Triangulation is an important topological tool. Triangulating a space is useful because working with simplicial complexes can be easier than working with a general semi-algebraic set. There exists an algorithm that will produce a semi-algebraic triangulation from a $\mathcal{P}$-semi-algebraic set. We see below that the ease that comes from working with a simplicial complex has a steep computational complexity cost.

Theorem 8. *[23, Theorem 4.5] Let $S \subset \mathbf{R}^m$ be a closed and bounded semi-algebraic set, and let $S_1, \ldots, S_k$ be semi-algebraic subsets of S. There exists a simplicial complex K in $\mathbf{R}^m$ and a semi-algebraic homeomorphism $h : |K| \to S$ such that each S_{j} is the union of images by h of open simplices of K. Moreover, the vertices of K can be chosen with rational coordinates. Moreover, if S and each S_{j} are $\mathcal{P}$-semi-algebraic sets, for some $\mathcal{P} \subset \mathbf{R}[X_1, \ldots, X_m]$ containing k polynomials bounded by degree at most d, the the semi-algebraic triangulation (K, h) can be computed in time $(kd)^{2^{\mathcal{O}(m)}}$.*

Our main algorithm uses a triangulation, and we rely on triangulation in our algorithm that extends a semi-algebraic function. Therefore, our algorithm will end up having doubly exponential complexity.

2.2 Proof of Theorem 1

We prove theorem 1 with an algorithm that follows the proof of theorem 5. We present here a brief summary of the steps in the proof of theorem 5. We reference the corresponding algorithms that line up with each step.

Proof. The effective steps of the proof of theorem 5 are contained in the General Quotient Algorithm 2.2.2 unless otherwise noted. We input a space X and equivalence relation $E \subset X \times X$ on X that is definably proper. We proceed by induction on the dimension of X. If $\dim(X) \leq 0$, then X is finite, and the theorem holds trivially. For $\dim(X) = d > 0$, we generate a subset of X that has dimension less than d. In the General Quotient Algorithm 2.2.2, this set, call it X', is the realization of the formula $\Phi_{X'}$, described [here]. Next we define a formula $\Phi_{E'}$ whose realization is $E' = E \cap (X' \times X')$. E' is definably proper over X' and

$\dim(X') < d$, so we may apply our inductive hypothesis. Algorithmically, this translates to applying the General Quotient Algorithm 2.2.2 again, [shown here]. We obtain $f' : X' \to Y'$ onto a definable set Y' with $E' = E_{f'}$. From here we need to construct $Y = X/E$. We do this by "gluing" another subset of X, which we call $\mathrm{cl}(S_D)$ defined [here], to Y' with theorem 10.2.12 in [1] (which in our case is the Second Gluing Quotient Algorithm). The Second Gluing Quotient Algorithm 2.2.2 relies on both the First Gluing Quotient Algorithm 2.2.2 and the Completion Algorithm 2.2.1. Both of these rely on the Extension Algorithm 2.2.1, which uses the Semi-Algebraic Path Algorithm 2.2.1, which in turn relies on the Partition of Unity Algorithm 2.2.1. After this chain of algorithms finishes, we will have obtained a space $Y = X/E$ as a definably proper quotient of $c(S_d) \amalg Y'$ via the map $p : c(S_d) \amalg Y' \to Y$.

We now have the desired quotient space Y, but obviously the domain of p is not X, so we need to obtain a new map. The remaining steps of the General Quotient Algorithm 2.2.2 define this new map whose domain is X and we are done, see [here]. □

Now that the general direction of the proof has been explained, we can define the algorithms that will construct a quotient space. We build our algorithm from the ground up. Presenting first the basic algorithms, we build up to the more complex algorithms that rely on these first results.

2.2.1 Preliminary Algorithms

Partition of Unity Algorithm

In our first algorithm, we input a $\mathcal{P}$-formula Φ_B and a family of n $\mathcal{P}$-open-formulae $\{\Phi_{U_\mathrm{i}}\}_{\mathrm{i}=1}^{n}$, describing a semi-algebraic set B and a semi-algebraic open cover $\{U_\mathrm{i}\}$ of B such that $U_\mathrm{i} \subset B$ for $\mathrm{i} = 1, \ldots, n$. From this we produce a family of formulae $\Phi_{f_1}, \ldots, \Phi_{f_n}$, which describe the graphs of semi-algebraic functions $f_1, \ldots, f_n$ which are a definable partition of unity for the covering $U_1, \ldots, U_n$.

Partition of Unity Algorithm

Input($\mathcal{P} \subset \mathbf{R}[X_1, \ldots, X_m]$, Φ_B a $\mathcal{P}$-formula describing a semi-algebraic set B, a family of $\mathcal{P}$-open-formulae $\{\Phi_{U_\mathrm{i}}\}_{\mathrm{i}=1}^{n}$ describing a family of semi-algebraic subsets $\{U_\mathrm{i}\}$ of B that cover B)

Output($\mathcal{Q} \subset \mathbf{R}[X_1, \ldots, X_{m+1}]$, $\Phi_{f_1}, \ldots, \Phi_{f_n}$ a family of $\mathcal{Q}$-formulae which describe the graphs of semi-algebraic functions which form a partition of unity for the covering $\{U_{\mathrm{i}}\}$)

Procedure:

(1) For i $= 1$ to n do the following:

(a) Set $\Phi_{A_0} \leftarrow \Phi_B \wedge (\neg \Phi_{U_{\mathrm{i}}})$ and $\Phi_{A_1} \leftarrow \Phi_B \wedge \neg \left(\bigvee_{\mathrm{j}=1}^{\mathrm{i}-1} \Phi_{V_{\mathrm{j}}} \vee \bigvee_{k=\mathrm{i}+1}^{n} \Phi_{U_k} \right)$.

(b) Set

$$\widetilde{\Phi_{V_{\mathrm{i}}}}(X) \leftarrow \exists X_1, X_2, t_1, t_2 \forall X_3, X_4 (\Phi_B(X)$$

$$\wedge M_{d_{A_0,B}}(X, t_1, X_1, X_3) \wedge M_{d_{A_1,B}}(X, t_2, X_2, X_4) \wedge t_2 < t_1).$$

(c) Let $\mathcal{Q}_0 = \emptyset$. Apply theorem 7 with inputs

$$(\mathcal{P} \cup \mathcal{Q}_{\mathrm{i}-1} \subset \mathbf{R}[X_1, \ldots, X_m, X_1', \ldots, X_{4m}', t_1, t_2],$$

$$\Pi = [(X_1, \ldots, X_m), (X_1', \ldots, X_{2m}', t_1, t_2), (X_{2m+1}', \ldots, X_{4m}')], \widetilde{\Phi_{V_{\mathrm{i}}}})$$

to obtain a set of polynomials $\mathcal{Q}_{\mathrm{i}} \subset \mathbf{R}[X_1, \ldots, X_m]$ and an equivalent quantifier free $\mathcal{Q}_{\mathrm{i}}$-formula $\Phi_{V_{\mathrm{i}}}$.

(d) Set $\Phi_{V_{\mathrm{i}}^C}(X) \leftarrow \Phi_B(X) \wedge (\neg \Phi_{V_{\mathrm{i}}}(X))$.

(e) Set

$$\widetilde{\Phi_{g_{\mathrm{i}}}}(X, t) \leftarrow \exists X_1, X_2, \exists t' > 1, \forall X_3, X_4$$

$$[(t \leq 1 \wedge M_{d_{V_{\mathrm{i}}^C,B}}(X, t, X_1, X_2)) \vee (t = 1 \wedge M_{d_{V_{\mathrm{i}}^C,B}}(X, t', X_2, X_4))].$$

(f) Apply theorem 7 with inputs

$$(\mathcal{P} \cup \mathcal{Q}_{\mathrm{i}} \subset \mathbf{R}[X_1, \ldots, X_{m+1}, X_1', \ldots, X_{4m}', t'],$$

$$\Pi = [(X_1, \ldots, X_{m+1}), (X_1', X_{2m}', t'), (X_{2m+1}', \ldots, X_{4m}')], \widetilde{\Phi_{g_{\mathrm{i}}}})$$

to obtain a set of polynomials $\mathcal{Q}_{\mathrm{i}}' \subset \mathbf{R}[X_1, \ldots, X_{m+1}]$ and an equivalent quantifier free $\mathcal{Q}_{\mathrm{i}}'$-formula $\Phi_{g_{\mathrm{i}}}$.

(2) Set $\mathcal{Q}' = \bigcup_{\mathrm{i}=1}^{n} \mathcal{Q}'_{\mathrm{i}}$.

(3) For $\mathrm{i} = 1$ to n, set

$$\widetilde{\Phi_{f_{\mathrm{i}}}}(X,t) \leftarrow \exists t_1,\dots,t_n \left(\bigwedge_{\mathrm{j}=1}^{n} \Phi_{g_{\mathrm{j}}}(X,t_{\mathrm{j}}) \wedge t \cdot \left(\sum_{\mathrm{j}} t_{\mathrm{j}} \right) = t_{\mathrm{i}} \right).$$

(4) For $\mathrm{i} = 1$ to n, apply theorem 7 with inputs

$$(\mathcal{Q}' \subset \mathbf{R}[X_1,\dots,X_{m+1},t_1,\dots,t_n], \Pi = [(X_1,\dots,X_{m+1}),(t_1,\dots,t_n)], \widetilde{\Phi_{f_{\mathrm{i}}}})$$

to obtain a set of polynomials $\mathcal{T}_{\mathrm{i}} \subset \mathbf{R}[X_1,\dots,X_{m+1}]$ and an equivalent quantifier free $\mathcal{T}_{\mathrm{i}}$-formula $\Phi_{f_{\mathrm{i}}}$.

(5) Let $\mathcal{Q} = \bigcup_{\mathrm{i}=1}^{n} \mathcal{T}_{\mathrm{i}} \subset \mathbf{R}[X_1,\dots,X_{m+1}]$.

(6) return$(\mathcal{Q}, \Phi_{f_1},\dots,\Phi_{f_n})$.

Complexity Analysis for Partition of Unity Algorithm:

We input a family of k polynomials $\mathcal{P} \subset \mathbf{R}[X_1,\dots,X_m]$ of degree at most d. In addition, we input a family of n $\mathcal{P}$-formulae.

(1) In step (1) we loop from $\mathrm{i} = 1$ to n to obtain formulae $\Phi_{V_{\mathrm{i}}}$ and $\Phi_{g_{\mathrm{i}}}$. Each iteration applies theorem 7 twice, returning two sets of polynomials $\mathcal{Q}_{\mathrm{i}}$ and $\mathcal{Q}'_{\mathrm{i}}$. Each formula and set of polynomials depends on the iteration prior, so we estabilish a reccurence relation. Let $q_{\mathrm{i}} = |\mathcal{Q}_{\mathrm{i}}|$ and $q'_{\mathrm{i}} = |\mathcal{Q}'_{\mathrm{i}}|$. Let $\mathfrak{d}_{\mathrm{i}}$ and $\mathfrak{d}'_{\mathrm{i}}$ bound the degrees of the polynomials of $\mathcal{Q}_{\mathrm{i}}$ and $\mathcal{Q}'_{\mathrm{i}}$, respectively. From the complexity analysis in [15] of Algorithm 14.5, $q_{\mathrm{i}}(k,d) \sim [q_{\mathrm{i}-1}(k,d)\mathfrak{d}_{\mathrm{i}-1}(k,d)]^{m^{\mathcal{O}(c)}}$. Similarly $\mathfrak{d}_{\mathrm{i}}(k,d) = \mathfrak{d}_{\mathrm{i}-1}(k,d)^{m^{\mathcal{O}(c)}}$. From here, we must solve the reccurence relation. Noting the initial condition $q_1(k,d) = (kd)^{m^{\mathcal{O}(c)}}$ and $\mathfrak{d}_1(k,d) = d^{m^{\mathcal{O}(c)}}$, we have that $q_{\mathrm{i}}(k,d) = (kd)^{m^{\mathcal{O}(c\mathrm{i})}}$ and $\mathfrak{d}_{\mathrm{i}}(k,d) = d^{m^{\mathcal{O}(c\mathrm{i})}}$. With this, we see that $q'_{\mathrm{i}}(k,d) = (q_{\mathrm{i}}(k,d)\mathfrak{d}_{\mathrm{i}}(k,d))^{(m+1)^{\mathcal{O}(c)}} = (kd)^{m^{\mathcal{O}(c\mathrm{i})}}$ and $\mathfrak{d}'_{\mathrm{i}}(k,d) = \mathfrak{d}_{\mathrm{i}}(k,d)^{(m+1)^{\mathcal{O}(c)}} = d^{m^{\mathcal{O}(c\mathrm{i})}}$. The computational complexity of step (1) of the algorithm is $(kd)^{\mathcal{O}(m^n)}$.

(2) In step (4), we apply theorem 7 n times. Each application inputs the set of polynomials $\mathcal{Q}' \subset \mathbf{R}[X_1,\dots,X_{m+1}]$ and contains on the order of $q'_n(k,d) = (kd)^{m^{\mathcal{O}(cn)}}$ polynomials

whose degrees are bounded by $\mathfrak{d}'_{\mathrm{i}}(k,d) = d^{m^{\mathcal{O}(cn)}}$. Applying quantifier elimination has complexity $(kd)^{m^{\mathcal{O}(cn)}}$ and returns a set of polynomials in $m+1$ variables of size $(kd)^{m^{\mathcal{O}(cn)}}$ whose degrees are bounded by $d^{m^{\mathcal{O}(cn)}}$.

(3) The total complexity of the algorithm is dominated by the final step, which occurs n times, and so is

$$(kd)^{m^{\mathcal{O}(n)}}.$$

Proof of Correctness for Partition of Unity Algorithm: The correctness follows from Lemma 6.3.7 of [1] and from the correctness of Algorithm 14.5 from [15].

Semi-Algebraic Path Algorithm

In the next algorithm, we present a way to generate a "path" between semi-algebraic sets. In other words, we input a $\mathcal{P}$-formula Φ_B, representing a semi-algebraic set B, and two $\mathcal{P}$-closed-formulae Φ_{A_0} and Φ_{A_1} representing disjoint closed semi-algebraic subsets A_0 and A_1 of B. The algorithm produces a formula Φ_f, representing the graph of a continuous semi-algebraic function $f : B \to [0,1]$ such that $f^{-1}(0) = A_0$ and $f^{-1}(1) = A_1$. In other words we produce a semi-algebraic path that starts at A_0 and ends at A_1.

Semi-Algebraic Path Algorithm

Input($\mathcal{P} \subset \mathbf{R}[X_1,\ldots,X_m]$, Φ_B a $\mathcal{P}$-formula describing a semi-algebraic set B, $\mathcal{P}$-closed-formulae Φ_{A_0} and Φ_{A_1} describing disjoint closed semi-algebraic sets $A_0, A_1 \subset B$)

Output($\mathcal{Q} \subset \mathbf{R}[X_1,\ldots,X_{m+1}]$, Φ_f a $\mathcal{Q}$-formula representing the graph of a semi-algebraic map f from B to the interval $[0,1]$ with the property that $f^{-1}(0) = A_0$ and $f^{-1}(1) = A_1$)

Procedure:

(1) Set

$$\widetilde{\Phi_{U_0}}(X) \leftarrow \exists t, t', X_0, X_1 \forall X'_0, X'_1 (\Phi_B(X)$$

$$\wedge M_{d_{A_0,B}}(X, t, X_0, X'_0) \wedge M_{d_{A_1,B}}(X, t', X_1, X'_1) \wedge t < t').$$

(2) Set

$$\widetilde{\Phi_{U_1}}(X) \leftarrow \exists t, t', X_0, X_1 \forall X_0', X_1' (\Phi_B(X)$$

$$\wedge M_{d_{A_0,B}}(X, t, X_0, X_0') \wedge M_{d_{A_1,B}}(X, t', X_1, X_1') \wedge t' < t).$$

(3) Apply theorem 7 with inputs

$$(\mathcal{P} \subset \mathbf{R}[X_1, \ldots, X_m, X_1', \ldots, X_{4m}', t_1, t_2],$$

$$\Pi = [(X_1, \ldots, X_m), (X_1', \ldots, X_{2m}', t_1, t_2), (X_{2m+1}', \ldots, X_{4m}')], \widetilde{\Phi_{U_0}})$$

and

$$(\mathcal{P} \subset \mathbf{R}[X_1, \ldots, X_m, X_1', \ldots, X_{4m}', t_1, t_2],$$

$$\Pi = [(X_1, \ldots, X_m), (X_1', \ldots, X_{2m}', t_1, t_2), (X_{2m+1}', \ldots, X_{4m}')], \widetilde{\Phi_{U_1}})$$

to obtain sets of polynomials $\mathcal{Q}_1$ and $\mathcal{Q}_2$ and an equivalent quantifier free $\mathcal{Q}_1$-formula Φ_{U_0} and an equivalent quantifier free $\mathcal{Q}_2$-formula Φ_{U_1}, respectively. Let $\mathcal{Q}' = \mathcal{Q}_1 \cup \mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_m]$.

(4) Apply the Partition of Unity Algorithm 2.2.1 with inputs

$$(\mathcal{P} \cup \mathcal{Q}', \Phi_B, \{\Phi_{U_i}\}_{i=0}^{1}, \Phi_B \wedge \neg(\Phi_{A_0} \vee \Phi_{A_1}))$$

to obtain a set of polynomials $\mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_{m+1}]$ and $\mathcal{Q}_3$-formulae $\Phi_{f_0}, \Phi_{f_1}, \Phi_{f_2}$.

(5) Set

$$\widetilde{\Phi_f}(X, t) \leftarrow \exists t_1, t_2, t_3, t_4, t_5, Y_1, Y_2 \forall Y_1', Y_2'$$

$$(\Phi_{f_0}(X, t_1) \wedge M_{d_{A_0,B}}(X, t_2, Y_1, Y_1') \wedge \Phi_{f_1}(X, t_3)$$

$$\wedge M_{d_{A_1,B}}(X, t_4 - \frac{1}{2}, Y_2, Y_2') \wedge \Phi_{f_2}(X, t_5) \wedge t = t_1 t_2 + t_3 t_4 + \frac{1}{2} t_5).$$

(6) Apply theorem 7 with inputs

$$(\mathcal{P} \cup \mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_{m+1}, X_1', \ldots, X_{4m}', t_1, t_2, t_3, t_4, t_5],$$

$$\Pi = [(X_1, \ldots, X_{m+1}), (X'_1, \ldots, X'_{2m}, t_1, t_2, t_3, t_4, t_5),$$

$$(X'_{2m+1}, \ldots, X'_{4m})], \widetilde{\Phi_f})$$

to obtain a set of polynomials $\mathcal{Q} \subset \mathbf{R}[X_1, \ldots, X_{m+1}]$ and an equivalent quantifier free $\mathcal{Q}$-formula Φ_f.

(7) return($\mathcal{Q}$, Φ_f).

Complexity Analysis for Semi-Algebraic Path Algorithm:
We input a family of k polynomials $\mathcal{P} \subset \mathbf{R}[X_1, \ldots, X_m]$ of degree at most d.

(1) In step (3) we apply quantifier elimination (theorem 7) to $\widetilde{\Phi_{U_0}}$ and $\widetilde{\Phi_{U_1}}$. In each case, the complexity is on the order $(kd)^{m^{\mathcal{O}(c)}}$. Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be the two sets of polynomials returned. Set $\mathcal{Q}' = \mathcal{Q}_1 \cup \mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_m]$ containing at most $(kd)^{m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $d^{m^{\mathcal{O}(c)}}$.

(2) In step (4) we apply the Partition of Unity Algorithm 2.2.1 with $n = 3$ and set of polynomials $\mathcal{P} \cup \mathcal{Q}'$. This has complexity $(kd)^{m^{\mathcal{O}(c)}}$. This returns a set of polynomials $\mathcal{Q}'' \subset \mathbf{R}[X_1, \ldots, X_m]$ of size $(kd)^{m^{\mathcal{O}(c)}}$ with degrees bounded by $d^{m^{\mathcal{O}(c)}}$.

(3) In step (6) we apply quantifier elimination to $\widetilde{\Phi_f}$. This step in the algorithm has complexity $(3kd)^{m^{\mathcal{O}(c)}}$, returning a set of polynomials $\mathcal{Q} \subset \mathbf{R}[X_1, \ldots, X_m]$ of size $(kd)^{m^{\mathcal{O}(c)}}$ whose degrees are bounded by $d^{m^{\mathcal{O}(c)}}$.

(4) Therefore the total complexity of the algorithm is

$$(kd)^{m^{\mathcal{O}(c)}}.$$

Proof of Correctness of Semi-Algebraic Path Algorithm: The correctness of this algorithm follows from the proof of Lemma 6.3.8 in [1], and from the correctness of Algorithm 14.5 in [15] and the correctness of the Partition of Unity Algorithm 2.2.1.

Extension Algorithm

From here, we use the preceding algorithm to help implement an algorithm that will extend a semi-algebraic function f. In this case, we input three sets of polynomials $\mathcal{P}_1, \mathcal{P}_2$, and $\mathcal{P}_3$. We input $\mathcal{P}_1$-formulae Φ_X and Φ_A, representing a semi-algebraic set X and a closed subset A, and a $\mathcal{P}_2$-formula Φ_φ, representing the graph of a semi-algebraic contraction $\varphi : B(n) \to \{0\}$, and a $\mathcal{P}_3$-formula Φ_f, representing the graph of a semi-algebraic function $f : A \to Y$, where Y is a space that we never need to directly reference so its formula is not needed as an input. The formula for Y can of course be obtained by projecting onto the last coordinates of Φ_f, if needed. We output a formula $\Phi_{f'}$, which represents the graph of a semi-algebraic map $f' : X \to Y$ such that $f'|_A = f$.

This algorithm is more complicated (in the colloquial sense of the word) than the previous two algorithms, so we preface it with some explanation. We first present the three results from [1] that we are making a effective. We then provide a brief summary of the proofs, linking to the appropriate corresponding lines in the algorithm.

We also will need a few more definitions to describe objects utilized in this algorithm.

Definition 13. *Let K be a simplicial complex. Given a definable set $A \subset |K|$. We define the* **star** *of A in K, denoted $st_K(A)$ to be the union of all simplices $\sigma \in K$ such that $\sigma \cap A \neq \emptyset$.*

Definition 14. *Let K be a complex.*

For $\sigma = (a_0, \dots, a_m) \in K$, the **barycenter of** *σ is the point*

$$b(\sigma) = \frac{1}{m+1}(a_0 + \dots + a_m)$$

A K-**flag** *is a sequence*

$$\mathfrak{F} : \sigma_0 < \sigma_1 < \dots < \sigma_m$$

of simplices of K such that each σ_{i} is a proper face of σ_{j} whenever $\mathrm{i} < \mathrm{j}$.

To each K-flag $\mathfrak{F}$ we associate an f-simplex $b(\mathfrak{F}) := (b(\sigma_0), \dots, b(\sigma_m))$ whose vertices are the barycenters of the simplices of $\mathfrak{F}$. If $\mathfrak{F}_1$ and $\mathfrak{F}_2$ are distinct flags, then $b(\mathfrak{F}_1)$ and $b(\mathfrak{F}_2$ are *disjoint. The (first) barycentric subdivision of K is the complex K' whose simplices are the simplices $b(\mathfrak{F})$ for each K-flag.*

Theorem 9 (Theorem 8.3.3 [1])**.** *Let K be a complex and L a subcomplex of K, closed in K. Let K' denote the first barycentric subdivision of K. Then there is a definable retraction $r : st_{K'}(|L|) \to |L|$ such that for each $x \in st_{K'}(|L|) - |L|$ the open line interval $(x, r(x))$ lies entirely in the simplex of K' that contains the point x.*

Theorem 10 (Theorem 8.3.9 [1])**.** *Let A be a definable closed subset of the definable set $B \subset \mathbf{R}^m$. Then there are a definable open subset U of B containing A, and a definable retraction $r : \mathrm{cl}(U) \cap B \to A$.*

Theorem 11 (Theorem 8.3.10 [1])**.** *Let A be a definable closed subset of the definable set $B \subset \mathbf{R}^m$, for some m. Let $f : A \to C$ be a definable continuous map into a definable set $C \subset \mathbf{R}^n$, for some n, that is definably contractible to a point $c \in C$. Then f can be extended into a definable continuous function $\tilde{f} : B \to C$.*

We begin by triangulating A and B into simplicial complexes L and K, respectively, in [line 1]. In this algorithm we will need the spaces K' (the first barycentric subdivision of K) and its simplices, and $st_{K'}(|L|)$, which we define in [line 2]. We also need to define several functions. First, for each vertex $\mathrm{e} \in \mathrm{Vert}(K)$, we define $\lambda_{\mathrm{e}} : |K| \to [0,1]$ as follows: for $x \in (\mathrm{e}_0, \ldots, \mathrm{e}_k)$, where $(\mathrm{e}_0, \ldots, \mathrm{e}_k)$ a k-simplex of K set

$$\lambda_{\mathrm{e}}(x) = \begin{cases} 0 & \mathrm{e} \notin \{\mathrm{e}_0, \ldots, \mathrm{e}_k\} \\ \lambda_{\mathrm{i}} & \mathrm{e} = \mathrm{e}_1 \end{cases},$$

where λ_{i} is the i-th barycentric coordinate of x. Now if $\{b(\sigma) | \sigma \in K\}$ is the set of vertices of K', we similarly define $\lambda_{b(\sigma)} : |K'| \to [0,1]$. For convenience, we denote this as λ_σ for σ a simplex of K. Next for each σ a simplex of K, we define a function $\omega_\sigma : |K| \to [0,1]$ as follows: $\omega_\sigma(x) = \begin{cases} 1 & \sigma \in L \\ 0 & \sigma \in K - L \end{cases}$. We now define for each σ a simplex of K the function $\mu_\sigma : |K| \to [0,1]$ given by $\mu_\sigma(x) = \omega_\sigma(x)\lambda_\sigma(x)$, we define μ_σ (indirectly defining ω_σ and λ_σ) in [line 6]. It is important to note here that μ_σ is both definable and continuous. Now we are ready to define the retraction promised by theorem 9: let $r(x) : st_{K'}(|L|) \to |L|$ by $r(x) = \frac{\sum_{\substack{\sigma \subset cl(K) \\ \sigma \text{simplex}}} \mu_\sigma(x) \cdot b(\sigma)}{\sum_{\substack{\sigma \subset cl(K) \\ \sigma \text{ simplex}}} \mu_\sigma(x)}$ which we define in [line 8]. This retraction has all the properties stated in the conclusion of theorem 9. We can use theorem 6.3.5 applied to L and $K - st_{K'}(L)$ to obtain a definable U open in K containing L such that $cl(U) \cap K \subset st_{K'}(L)$, see [lines 10-13]. To finish the proof of theorem 10, we restrict the retraction function from above to $cl(U) \cap K$, [line 14]. To obtain the function from theorem 11 we take the U and r that we found and begin by defining a function $\lambda : B \to [0,1]$ such that $\lambda^{-1}(0) = A$ and $\lambda^{-1}(1) = B - U$. The space C is contractible, so let ϕ denote a contraction. We define $\tilde{f} : B \to C$ by

$$\tilde{f}(x) = \begin{cases} \phi(f(r(x)), \lambda(x)) & x \in \mathrm{cl}(U) \cap B \\ c & x \in B - U \end{cases}$$

[line 15]

Extension Algorithm

Input($\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m]$, Φ_X a $\mathcal{P}_1$-formula representing a space X, Φ_A a $\mathcal{P}_1$ -closed-formula representing a closed subset A of X, $\mathcal{P}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m+1}]$, Φ_φ a $\mathcal{P}_2$-formula representing the graph of a semi-algebraic contraction φ, $\mathcal{P}_3 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$, Φ_f a $\mathcal{P}_3$-formula representing the graph of a semi-algebraic map f from A to a set Y)

Output($\mathcal{Q} \subset \mathbf{R}[X_1, \ldots, X_{2m}]$, $\Phi_{f'}$ a $\mathcal{Q}$-formula representing the graph of a semi-algebraic map f' from X to Y which extends f)

Procedure:

(1) Apply theorem 8 (semi-algebraic triangulation) with inputs $(\mathcal{P}_1, \Phi_X, \Phi_A)$ to obtain a triangulation (h, K) of X compatible with A. Let $L = h|_A(A)$. We obtain sets of polynomials $\mathcal{Q}_\sigma$ and $\mathcal{Q}_\sigma$-formulae $\Phi_{K,\sigma}$, for each simplex σ of K.

By restricting to A, we also obtain formulae $\Phi_{L,\sigma}$ for each $\sigma \in L$. Let $\mathcal{Q}_1 = \bigcup_\sigma \mathcal{Q}_\sigma \subset \mathbf{R}[X_1, \ldots, X_{2m+1}]$.

(2) Given $\sigma = (v_0, \ldots, v_p)$, let $b(\sigma) = \frac{1}{p+1}(v_0 + \cdots + v_p)$. For $\tau = (r_0, \ldots, r_q) \in K'$, the first barycentric subdivision of K, with $\dim(\tau) = q$, set

$$\widetilde{\Phi_{K',\tau}}(s_0, \ldots, s_q, X) = \exists t_0, \ldots, t_m$$

$$\bigvee_{\mathrm{j}_q = q}^{m} \bigvee_{\substack{\sigma_0 \subset \cdots \subset \sigma_q \in K \\ \sigma_{\mathrm{i}} = (u_0, \ldots, u_{\mathrm{j_i}}) \\ u_k \in \mathrm{Vert}(K) \\ \mathrm{j}_0 \le \mathrm{j}_1 \le \cdots \le \mathrm{j}_q}} \Phi_{K,\sigma_q}(t_0, \ldots, t_{\mathrm{j}_q}, X) \wedge$$

$$\bigwedge_{\mathrm{i}=0}^{q} r_{\mathrm{i}} = b(\sigma_{\mathrm{i}}) \wedge s_0 r_0 + \cdots + s_q r_q = t_0 u_0 + \cdots + t_{\mathrm{j}_q} u_{\mathrm{j}_q}$$

Similarly for $\tau = (r_0, \ldots, r_q) \in L'$, the first barycentric subdivision of L, with $\dim(\tau) = q$, set

$$\widetilde{\Phi_{L',\tau}}(s_0, \ldots, s_q, X) = \exists t_0, \ldots, t_m$$

$$\bigvee_{\mathrm{j}_q = q}^{m} \bigvee_{\substack{\sigma_0 \subset \cdots \subset \sigma_q \in L \\ \sigma_{\mathrm{i}} = (u_0, \ldots, u_{\mathrm{j_i}}) \\ u_k \in \mathrm{Vert}(L) \\ \mathrm{j}_0 \le \mathrm{j}_1 \le \cdots \le \mathrm{j}_q}} \Phi_{L,\sigma_q}(t_0, \ldots, t_{\mathrm{j}_q}, X) \wedge$$

$$\bigwedge_{\mathrm{i}=0}^{q} r_{\mathrm{i}} = b(\sigma_{\mathrm{i}}) \wedge s_0 r_0 + \cdots + s_q r_q = t_0 u_0 + \cdots + t_{\mathrm{j}_q} u_{\mathrm{j}_q}.$$

(3) Apply theorem 7 for each $\sigma \in K'$ and each $\sigma \in L'$ with inputs:

$$\Big(\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_{2m+1}, X_1', \ldots, X_{m+1}'],$$

$$\Pi = [(X_1, \ldots, X_{2m+1}), (X_1', \ldots, X_{m+1}')], \widetilde{\Phi_{K',\sigma}}\Big)$$

and

$$\Big(\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_{2m+1}, X_1', \ldots, X_{m+1}'],$$

$$\Pi = [(X_1, \ldots, X_{2m+1}), (X_1', \ldots, X_{m+1}')], \widetilde{\Phi_{L',\sigma}}\Big)$$

to obtain several sets of polynomials. Let $\mathcal{Q}_2$ denote the union of all sets of polynomials obtained for each simplex of K', and $\mathcal{Q}_3$ for each simplex in L'. Then $\mathcal{Q}_2, \mathcal{Q}_3 \subset$

$\mathbf{R}[X_1, \ldots, X_{2m+1}]$ and for each $\sigma \in K'$ we obtain equivalent quantifier free $\mathcal{Q}_2$-formulae $\Phi_{K',\sigma}$. Similarly for each $\sigma \in L'$, we obtain equivalent quantifer free $\mathcal{Q}_3$-formulae $\Phi_{L',\sigma}$.

(4) Set

$$\widetilde{\Phi_{st_{K'}(L)}}(X) \leftarrow \exists t_0, \ldots t_m, s_0, \ldots, s_m, w_0, \ldots, w_m, Y$$

$$\bigvee_{\mathrm{i}=0}^{m} \bigvee_{\substack{\sigma \in K' \\ \dim(\sigma)=\mathrm{i}}} \Phi_{K',\sigma}(t_0, \ldots, t_{\mathrm{i}}, X)$$

$$\wedge \bigvee_{\mathrm{j}=0}^{\mathrm{i}-1} \bigvee_{\substack{\sigma' < \sigma \\ \dim(\sigma')=\mathrm{j}}} \Phi_{K',\sigma'}(s_0, \ldots, s_{\mathrm{j}}, Y)$$

$$\wedge \bigvee_{k=0}^{m} \bigvee_{\substack{\gamma \in L \\ \dim(\gamma)=k}} \Phi_{L,\gamma}(w_0, \ldots, w_k, Y).$$

(5) Apply theorem 7 with inputs

$$\Big(\mathcal{Q}_1 \cup \mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_m, X'_1, \ldots, X'_{4m+3}],$$

$$\Pi = [(X_1, \ldots, X_m), (X'_1, \ldots, X'_{4m+3})], \widetilde{\Phi_{st_{K'}(L)}}\Big)$$

to obtain a set of polynomials $\mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_m]$ and an equivalent quantifier free $\mathcal{Q}_4$-formula $\Phi_{st_{K'}(L)}$.

(6) For $\sigma \in K - L$, set $\Phi_{\mu_{\sigma,K,L}}(X,t) = (t = 0)$.

For $\sigma = (v_0, \ldots, v_p) \in L$, set

$$\Phi_{\mu_{\sigma,K,L}}(X,t) = \exists t_0, \ldots, t_m \bigvee_{\mathrm{i}=0}^{m} \bigvee_{\substack{\sigma' \in L' \\ \dim(\sigma')=\mathrm{i} \\ \sigma'=(v'_0,\ldots,v'_{\mathrm{i}})}} \Phi_{L',\sigma'}(t_0, \ldots, t_{\mathrm{i}}, X)$$

$$\wedge \left(\left(\bigvee_{\mathrm{j}=0}^{\mathrm{i}} v'_{\mathrm{j}} = \frac{1}{p+1}(v_0 + \cdots + v_p) \wedge t = t_{\mathrm{j}} \right) \vee t = 0 \right).$$

(7) For $\sigma \in L$, apply theorem 7 with inputs

$$(Q_3 \subset \mathbf{R}[X_1, \ldots, X_{2m+1}, X_{,1}, \ldots, X'_{m+1}],$$

$$\Pi = [(X_1, \ldots, X_{m(m+1)}), (X'_1, \ldots, X'_{m+1})], \widetilde{\Phi_{\mu_{\sigma,K,L}}})$$

to obtain a set of polynomials $\mathcal{Q}_5 \subset \mathbf{R}[X_1, \ldots, X_{m+1}]$ and equivalent quantifier free $\mathcal{Q}_5$-formulae $\Phi_{\mu_{\sigma,K,L}}$.

(8) As before, given $\sigma = (v_0, \ldots, v_p)$, let $b(\sigma) = \frac{1}{p+1}(v_0 + \cdots + v_p)$. We set

$$\widetilde{\Phi_{r'}}(X, Y) \leftarrow \exists t_0, \ldots t_m, s_0, \ldots, s_m \bigvee_{\mathrm{i}=0}^{m} \left(\bigvee_{\substack{\tau \in K' \\ \dim(\tau)=\mathrm{i}}} \Phi_{K',\tau}(s_0, \ldots, s_{\mathrm{i}}, X) \wedge \right.$$

$$\left. \bigwedge_{\mathrm{j}=0}^{\mathrm{i}} \bigwedge_{\substack{\sigma_{\mathrm{j}} \in K \\ b(\sigma_{\mathrm{j}}) \in \tau}} \Phi_{\mu_{\sigma_{\mathrm{j}},K,L}}(X, t_{\mathrm{j}}) \wedge Y \cdot \sum t_{\mathrm{j}} = \sum (t_{\mathrm{j}} \cdot b(\sigma_{\mathrm{j}})) \right).$$

(9) Apply theorem 7 with inputs

$$\Big(\mathcal{Q}_5 \subset \mathbf{R}[X_1, \ldots, X_{m+1}, X'_1, \ldots, X'_{2m+2}],$$

$$\Pi = [(X_1, \ldots, X_{m+1}), (X'_1, \ldots, X'_{2m+2})], \widetilde{\Phi_{r'}}(X, Y)\Big)$$

to obtain a set of polynomials $\mathcal{Q}_6 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ and an equivalent quantifier free $\mathcal{Q}_6$-formula $\Phi_{r'}(X, Y)$.

(10) Set $\Phi_{K - st_{K'}(L)}(X) \leftarrow \Phi_K(X) \wedge \neg \Phi_{st_{K'}(L)}(X)$.

(11) Set

$$\widetilde{\Phi_U}(X) \leftarrow \exists t_0, \ldots, t_m, s, s', X'_1, Y'_1 \forall X'_2, Y'_2$$

$$\bigvee_{\mathrm{i}=0}^{m} \bigvee_{\substack{\sigma \in K \\ \dim(\sigma)=\mathrm{i}}} \Phi_{K,\sigma}(t_0, \ldots, t_{\mathrm{i}}, X) \wedge M_{d_{K - st_{K'}(L),X}}(X, s, X'_1, X'_2)$$

$$\wedge M_{d_{A,X}}(X, t', Y'_1, Y'_2) \wedge (t' < t).$$

(12) Apply theorem 7 with inputs

$$\left(\mathcal{P}_1 \cup \mathcal{Q}_1 \cup \mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_m, X'_1, \ldots, X'_{4m}, t, t'],\right.$$

$$\left.\Pi = [(X_1, \ldots, X_m), (X'_1, \ldots, X'_{2m}, t, t'), (X'_{2m+1}, \ldots, X'_{4m})], \widetilde{\Phi_U}\right)$$

to obtain a set of polynomials $\mathcal{Q}_7 \subset \mathbf{R}[X_1, \ldots, X_m]$ and an equivalent quantifier free $\mathcal{Q}_7$-formula Φ_U.

(13) Apply the Semi-Algebraic Path Algorithm 2.2.1 with inputs

$$(\mathcal{P}_1 \cup \mathcal{Q}_7, \Phi_X, \Phi_A, \Phi_X \wedge (\neg \Phi_U))$$

to obtain a set of polynomials $\mathcal{Q}_8 \subset \mathbf{R}[X_1, \ldots, X_{m+1}]$ and a $\mathcal{Q}_8$-formula $\Phi_g(X, t)$ representing the graph of a semi-algebraic map from X to $[0, 1]$.

(14) Set

$$\Phi_r(X, Y) \leftarrow \forall \varepsilon > 0 \exists X' (\Phi_{r'}(X, Y) \wedge M_{cl(U)}(X, \varepsilon, X') \wedge \Phi_X(X)).$$

(15) Set

$$\widetilde{\Phi_{f'}}(X, Y) \leftarrow \exists Z_1, Z_2, t \forall \varepsilon > 0 \exists X'$$

$$[(M_r(X, Z_1, \varepsilon, X') \wedge \Phi_g(X, t) \wedge \Phi_f(Z_1, Z_2) \wedge \Phi_\varphi(Z_2, t, Y))$$

$$\vee (\Phi_X(X) \wedge \neg \Phi_U(X) \wedge Y = 0)].$$

(16) Apply theorem 7 with inputs

$$\left(\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3 \cup \mathcal{Q}_6 \cup \mathcal{Q}_7 \cup \mathcal{Q}_8 \subset \mathbf{R}[X_1, \ldots, X_{2m}, X'_1, \ldots, X'_{3m}, t, \varepsilon],\right.$$

$$\left.\Pi = [(X_1, \ldots, X_{2m}), (X'_1, \ldots, X'_{2m}, t), (\varepsilon), (X'_{2m+1}, \ldots, X'_{3m})], \widetilde{\Phi_{f'}}\right)$$

to obtain a set of polynomials $\mathcal{Q}_9 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ and an equivalent quantifier free $\mathcal{Q}_9$-formula $\Phi_{f'}$.

(17) return($\mathcal{Q}_9$, $\Phi_{f'}(X,Y)$).

Complexity Analysis for Extension Algorithm: We input three families of polynomials: $\mathcal{P}_1 \subset \mathbf{R}[X_1,\ldots,X_m]$ of size k_1 whose degrees are bounded by d_1, $\mathcal{P}_2 \subset \mathbf{R}[X_1,\ldots,X_{2m+1}]$ of size k_2 whose degrees are bounded by d_2, and $\mathcal{P}_3 \subset \mathbf{R}[X_1,\ldots,X_{2m}]$ of size k_3 whose degrees are bounded by d_3.

(1) In step (1) we apply the triangulation algorithm, theorem 8, once. This has complexity $(k_1d_1)^{2^{\mathcal{O}(m)}}$. From this, we obtain a set of polynomials $\mathcal{Q}_1 \subset \mathbf{R}[X_1,\ldots,X_m]$ of size $(k_1d_1)^{2^{\mathcal{O}(m)}}$ whose degrees are bounded by $d_1^{2^{\mathcal{O}(m)}}$.

(2) In step (3) we apply quantifier elimination to $\widetilde{\Phi_{K',\tau}}$ and $\widetilde{\Phi_{L',\tau}}$ for each $\tau \in K$ and L, respectively. Each application has complexity $(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$. The number of simplices of K is bounded above by $(k_1d_1)^{2^{\mathcal{O}(m)}}$. Each simplex of K and L, contributes up to $m^{\mathcal{O}(m)}$ simplices to K' and L'. There fore K' and L' contain at most $(k_1d_1)^{2^{\mathcal{O}(m)}}m^{\mathcal{O}(m)}$ simplices. Since we apply quantifier elimination for each simplex of K' and L', this step has complexity

$$m^{\mathcal{O}(m)}(k_1d_1)^{2^{\mathcal{O}(m)}}.$$

Let $\mathcal{Q}_2, \mathcal{Q}_3 \subset \mathbf{R}[X_1,\ldots,X_{2m+1}]$ denote the unions of the polynomials obtained from each $\tau \in K'$ and L', respectively. Therefore $\mathcal{Q}_2$ and $\mathcal{Q}_3$ contain at most $m^{\mathcal{O}(m)}(k_1d_1)^{2^{\mathcal{O}(m)}}$ polynomials with degrees bounded by $(d_1)^{2^{\mathcal{O}(m)}}$.

(3) In step (5) we apply quantifier elimination to $\widetilde{\Phi_{st_{K'}(L)}}$. This has complexity

$$(mk_1d_1)^{2^{\mathcal{O}(m)}}.$$

We return a set of polynomials $\mathcal{Q}_4 \subset \mathbf{R}[X_1,\ldots,X_m]$ of size $(mk_1d_1)^{2^{\mathcal{O}(m)}}$ whose degrees are bounded by $(d_1)^{2^{\mathcal{O}(m)}}$.

(4) In step (7), we apply quantifier elimination for each $\sigma \in L$. For each iteration, this has complexity $(mk_1d_1)^{2^{\mathcal{O}(m)}}$. The number of simplices in L is bounded above by the number

of simplices in K, so we perform this step at most $(k_1 d_1)^{2^{\mathcal{O}(m)}}$ times. Hence the total complexity of this step is

$$(mk_1 d_1)^{2^{\mathcal{O}(m)}}.$$

Each iteration returns a set of polynomials. Let $\mathcal{Q}_5 \subset \mathbf{R}[X_1, \ldots, X_{m+1}]$ denote the union of these polynomials. Then $\mathcal{Q}_5$ contains at most $(mk_1 d_1)^{2^{\mathcal{O}(m)}}$ polynomials of degree at most $(d_1)^{2^{\mathcal{O}(m)}}$.

(5) In step (9) we apply quantifier elimination to $\widetilde{\Phi_{r'}}$. This has complexity

$$(mk_1 d_1)^{2^{\mathcal{O}(m)}}.$$

We obtain a set of polynomials $\mathcal{Q}_6 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ containing at most $(mk_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_1)^{2^{\mathcal{O}(m)}}$.

(6) In step (12) we apply quantifier elimination to $\widetilde{\Phi_U}$. This has complexity $(mk_1 d_1)^{2^{\mathcal{O}(m)}}$ and returns a set of polynomials $\mathcal{Q}_7 \subset \mathbf{R}[X_1, \ldots, X_m]$ containing at most $(k_1 d_1)^{2^{\mathcal{O}(m)}}$ polynomials whose degrees are bounded by $(d_1)^{2^{\mathcal{O}(m)}}$.

(7) In step (13), we apply the Semi-Algebraic Path Algorithm 2.2.1 with $\mathcal{P}_1 \cup \mathcal{Q}_7$ as the set of polynomials. This has complexity $(mk_1 d_1)^{2^{\mathcal{O}(m)}}$ and returns a set of polynomials $\mathcal{Q}_8 \subset \mathbf{R}[X_1, \ldots, X_{m+1}]$ containing at most $(mk_1 d_1)^{2^{\mathcal{O}(m)}}$ polynomials whose degrees are bounded by $(d_1)^{2^{\mathcal{O}(m)}}$.

(8) In step (16) we apply quantifier elimination to $\widetilde{\Phi_{f'}}$. This has complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (k_3)^{m^{\mathcal{O}(c)}} + (mk_1 d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_3^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right).$$

It returns a set of polynomials $\mathcal{Q}_9 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ containing at most

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (k_3)^{m^{\mathcal{O}(c)}} + (mk_1 d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_3^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right).$$

polynomials whose degrees are bounded by $\left(d_2^{m^{\mathcal{O}(c)}} + d_3^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$.

(9) The last step dominates the complexity of the other steps, so the total complexity of Algorithm 8.3 is bounded above by

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (k_3)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_3^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

Proof of Correctness of Extension Algorithm: The correctness of the algorithm follows from the proofs of Proposition 8.3.3, Corollary 8.3.9, and Corollary 8.3.10 in [1], and from the correctness of Algorithm 14.5 [15], the triangulation algorithm, and the Semi-Algebraic Path Algorithm.

Completion Algorithm

From here, the next step is to create a specific completion diagram for a given function. We input two sets of polynomials $\mathcal{P}_1$ and $\mathcal{P}_2$. We input three $\mathcal{P}_1$-formulae, Φ_X, Φ_A, and Φ_Y, representing semi-algebraic sets X, A, and Y, respectively, such that $A \subset X$ and Y is bounded. We input a single $\mathcal{P}_2$-formula, Φ_f, representing the graph of a semi-algebraic map $f : A \to Y$. Because Y is bounded, $\mathrm{j} : Y \to \mathrm{cl}(Y)$ is a completion of Y. Using this completion, we are able to obtain a completion diagram of f:

$$\begin{array}{ccc} A & \xrightarrow{\mathrm{i}} & A' \\ \big\downarrow{\scriptstyle f} & & \big\downarrow{\scriptstyle f'} \\ Y & \xrightarrow{\mathrm{j}} & \mathrm{cl}(Y) \end{array}$$

These sets are outputted by our algorithm as formulae $\Phi_{f'}$ and $\Phi_{A'}$, representing the graph of f' and the semi-algebraic set A', respectively. We will generate a formula $\Phi_{X'}$, representing X' the image of completion of X, that we will need in the next algorithm.

Completion Algorithm

Input($\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m]$, Φ_X a $\mathcal{P}_1$-formula describing a semi-algebraic set X, Φ_A a $\mathcal{P}_1$-closed-formula describing a semi-algebraic subset A of X, Φ_Y a $\mathcal{P}_1$ bounded formula describing a semi-algebraic set Y, $\mathcal{P}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$, Φ_f a $\mathcal{P}_2$-formula representing the graph of a semi-algebraic map from A to Y)

Output($\mathcal{Q} \subset \mathbf{R}[X_1, \ldots, X_{2m}]$, $\Phi_{X'}$ a $\mathcal{Q}$-formula describing a completion X' of X, $\Phi_{A'}$ a $\mathcal{Q}$-formula describing a completion A' of A, $\mathcal{Q}' \subset \mathbf{R}[X_1, \ldots, X_{3m}]$, $\Phi_{f'}$ a $\mathcal{Q}'$-formula describing a map from A' to $\mathrm{cl}(Y)$)

Procedure:

(1) For any r, set $\Phi_{B(r)}(X) \leftarrow ||X||^2 \leq r^2$.

(2) Apply Algorithm 14.3 from [15] with input

$$(\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m, X'_1, \ldots, X'_{2m}, \varepsilon],$$

$$\Pi = [(X_1, \ldots, X_m), (X'_1, \ldots, X'_m), (\varepsilon), (X'_{m+1}, \ldots, X'_{2m})],$$

$$\forall Y \exists \varepsilon > 0 \forall X (\Phi_Y(X) \wedge ||X - Y||^2 < \varepsilon^2 \rightarrow \Phi_{B(r)}(Y))\big)$$

for increasing values of $r = 1, 2, \ldots$ until the algorithm returns true. Set n equal to the first value that returns true.

(3) Set $\Phi_\varphi(X, t, Y) \leftarrow \Phi_{B(n)}(X) \wedge 0 \leq t \leq 1 \wedge [Y = X \cdot (1 - t)]$, representing the graph of a semi-algebraic contraction from $B(n)$ to $\{0\}$.

(4) Apply the Extension Algorithm 2.2.1 with inputs

$$(\mathcal{P}_1, \Phi_X, \Phi_A, \mathcal{P}_1(X) \cup \mathcal{P}_1(Y) \subset \mathbf{R}[X_1, \ldots, X_m, t, Y_1, \ldots, Y_m], \Phi_\varphi, \mathcal{P}_2, \Phi_f)$$

to obtain a set of polynomials $\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ and a $\mathcal{Q}_1$-formula Φ_{f^*} representing the graph of a semi-algebraic map from X to $B(n)$.

(5) Set $\Phi_\mu(X_1, \ldots, X_m, Y_1, \ldots, Y_m) \leftarrow \bigwedge_{\mathrm{i}=1}^{m} \Big((2X_\mathrm{i}Y_\mathrm{i} + 1)^2 = 1 + 4X_\mathrm{i}^2\Big)$, a formula representing the graph of a semi-algebraic homeomorphism from $\mathbf{R}^m \rightarrow (-1, 1)^m$.

(6) Set

$$\widetilde{\Phi_{X'}}((X, Y)) \leftarrow \forall \varepsilon > 0 \exists A, B, C$$

$$(\Phi_X(A) \wedge \Phi_{B(n)}(Y) \wedge \Phi_\mu(A, B) \wedge \Phi_{f^*}(A, C) \wedge ||(B, C) - (X, Y)||^2 < \varepsilon^2).$$

(7) Apply theorem 7 with inputs

$$(\mathcal{Q}_1 \cup \mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_{2m}, X'_1, \ldots, X'_{3m}\varepsilon],$$

$$\Pi = [(X_1, \ldots, X_{2m}), (X'_1, \ldots, X'_{3m}), (\varepsilon)], \widetilde{\Phi_{X'}})$$

to obtain a set of polynomials $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ and an equivalent quantifier free $\mathcal{Q}_2$-formula $\Phi_{X'}$.

(8) Set $\Phi_{f''}((X,Y),Z) \leftarrow (\Phi_{X'}((X,Y)) \wedge X = Z)$.

(9) Set $\Phi_h(X,(Y,Z)) \leftarrow \Phi_\mu(X,Y) \wedge \Phi_{f^*}(X,Z) \wedge \Phi_X(X) \wedge \Phi_{B(n)}(Z)$.

(10) Set

$$\widetilde{\Phi_{A'}}(Y) \leftarrow \forall \varepsilon > 0 \exists Z_1, Z_2(\Phi_A(Z_1) \wedge \Phi_h(Z_1, Z_2) \wedge ||Y - X_2||^2 < \varepsilon^2 \wedge \Phi_{X'}(Y)).$$

(11) Apply theorem 7 with inputs

$$(\mathcal{P}_1 \cup \mathcal{Q}_1 \cup \mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}, \varepsilon, X'_1, \ldots, X'_{2m}],$$

$$\Pi = [(X_1, \ldots, X_{2m}), (\varepsilon), (X'_1, \ldots, X'_{2m})], \widetilde{\Phi_{A'}})$$

to obtain a set of polynomials $\mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ and an equivalent quantifier free $\mathcal{Q}_3$-formula $\Phi_{A'}$.

(12) Set $\Phi_{f'}(X,Y) \leftarrow \Phi_{A'}(X) \wedge \Phi_{f''}(X,Y)$.

(13) Set $\widetilde{\Phi_{X_{\mathrm{new}}}}((X,Y)) \leftarrow \exists A(\Phi_X(A) \wedge \Phi_\mu(A,X) \wedge \Phi_{f^*}(A,Y))$.

(14) Apply theorem 7 with inputs

$$(\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_{2m}, X'_1, \ldots, X'_m],$$

$$\Pi = [(X_1, \ldots, X_{2m}), (X'_1, \ldots, X_m)], \widetilde{\Phi_{X_{\mathrm{new}}}})$$

to obtain a set of polynomials $\mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ and an equivalent quantifier free $\mathcal{Q}_4$-formula $\Phi_{X_{\text{new}}}$.

(15) return$(\mathcal{Q}_2 \cup \mathcal{Q}_3 \cup \mathcal{Q}_4, \Phi_{X_{\text{new}}}, \Phi_{X'}, \Phi_{A'}, \mathcal{Q}_2(X) \cup \mathcal{Q}_3(Y) \subset \mathbf{R}[X_1, \ldots, X_{3m}], \Phi_{f'})$.

Complexity Analysis for Completion Algorithm: We input two families of polynomials: $\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m]$ of size k_1 whose degrees are bounded by d_1, and $\mathcal{P}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ of size k_2 whose degrees are bounded by d_2.

(1) In step (2) we apply a general decision algorithm n times. This has complexity on the order of $n(k_1 d_1)^{m^{\mathcal{O}(c)}}$.

(2) In step (4) we apply the Extension Algorithm 2.2.1. This step has complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right).$$

This returns a set of polynomials $\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ containing at most $\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$ polynomials whose degrees are bounded by $(d_2^{m^{\mathcal{O}(c)}} + (d_1)^{2^{\mathcal{O}(m)}})$.

(3) In step (7) we apply quantifier elimination to $\widetilde{\Phi_{X'}}$. This has complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

and returns a set of polynomials $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ containing at most $\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$ polynomials whose degrees are bounded by $(d_2^{m^{\mathcal{O}(c)}} + (d_1)^{2^{\mathcal{O}(m)}})$.

(4) In step (11) we apply quantifier elimination to $\widetilde{\Phi_{A'}}$. This has complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

and returns a set of polynomials $\mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ containing at most $\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$ polynomials whose degrees are bounded by $(d_2^{m^{\mathcal{O}(c)}} + (d_1)^{2^{\mathcal{O}(m)}})$.

(5) In step (13) we apply quantifier elimination to $\widetilde{\Phi_{X_{\mathrm{new}}}}$. This has complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

and returns a set of polynomials $\mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_m]$ containing at most $\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$ polynomials whose degrees are bounded by $(d_2^{m^{\mathcal{O}(c)}} + (d_1)^{2^{\mathcal{O}(m)}})$.

(6) Therefore the entire algorithm has complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right).$$

Proof of Correctness of Completion Algorithm: The correctness of this algorithm follows from Lemma 10.2.7 in [1], and from the correctness of Algorithms 14.3 and 14.5 in [15] and the Extension Algorithm.

2.2.2 Quotient Algorithms

First Gluing Quotient Algorithm

While we need the Completion Algorithm to obtain quotient spaces of more general situations, the Extension Algorithm is itself enough to generate a quotient in certain instances. In the following algorithm, we input two sets of polynomials $\mathcal{P}_1$ and $\mathcal{P}_2$. We need three $\mathcal{P}_1$-formulae, Φ_X, Φ_A, and Φ_Y, representing three semi-algebraic sets X, A, and Y such that A is a closed subset of X. We need a single $\mathcal{P}_2$-formula Φ_f describing the graph of a semi-algebraic map $f : A \to Y$. With these inputs, we are able to generate a formula Φ_Z, a formula whose realization Z is semi-algebraically homeomorphic to the quotient space $X \amalg_f Y$, and a formula Φ_p describing the graph of a map $p : X \amalg Y \to Z$.

As we did with the Extension Algorithm, we state van den Dries's theorem with a brief proof, linking to the corresponding steps in the algorithm.

Theorem 12 (Theorem 10.2.11 [1]). *Let $X \subset \mathbf{R}^m$ and $Y \subset \mathbf{R}^n$ be definable sets. Let $A \subset X$ be definable, closed, and bounded in the ambient space $\mathbf{R}^m$ of X. Let $f : A \to Y$ be a definable continuous map. Then $X \amalg_f Y$ exists as a definably proper quotient of $X \amalg Y$.*

Proof. First if $A = \emptyset$, the identity map $X \amalg Y \to X \amalg Y$ is a sufficient definably proper quotient of $X \amalg Y$ by $E(f)$, lines [1-2]. Otherwise, we let $\mathbf{R}^M$ be the ambient space of $X \amalg Y$ (so $M = \max\{m, n\} + 1$). We identify X, A, and Y with their images in $X \amalg Y$, noting that then A is closed and bounded in $\mathbf{R}^M$, line [6]. Let $\tilde{f} : X \to \mathbf{R}^M$ be a definable continuous extension of $f : A \to Y$, line [5,7] and let $d_A : \mathbf{R}^M \to \mathbf{R}$ be the distance function on A. Finally, we define a map $p : X \amalg Y \to \mathbf{R}^{2M+1}$ by the formula

$$p(x) = \begin{cases} (\tilde{f}(x), d_A(x) \cdot x, d_A(x)) & x \in X \\ (x, 0, 0) & x \in Y \end{cases},$$

[line 8]. Let $Z = p(X \amalg Y)$, [line 10], then (p, Z) is the desired definable quotient of $X \amalg Y$ by $E(f)$. $\square$

First Gluing Quotient Algorithm

Input($\mathcal{P}_1 \subset \mathbf{R}[X_1, \dots, X_m]$, Φ_X a $\mathcal{P}_1$-formula describing a semi-algebraic set X, Φ_A a $\mathcal{P}_1$ -closed-formula describing a semi-algebraic subset A of X, Φ_Y a $\mathcal{P}_1$-formula describing a semi-algebraic set Y, $\mathcal{P}_2 \subset \mathbf{R}[X_1, \dots, X_{2m}]$, Φ_f a $\mathcal{P}_2$-formula representing a semi-algebraic map from A to Y)

Output($\mathcal{Q} \subset \mathbf{R}[X_1, \dots, X_{2m+3}]$, Φ_Z a $\mathcal{Q}_1$-formula describing the quotient space $X \amalg_f Y$, $\mathcal{Q}' \subset \mathbf{R}[X_1, \dots, X_{3m+4}]$, Φ_p a formula describing the graph of the quotient map p from $X \amalg Y$ to $X \amalg_f Y$)

Procedure:

(1) Apply Algorithm 14.3 from [15] with inputs

$$(\mathcal{P}_1 \subset \mathbf{R}[X_1, \dots, X_{2m}], \Pi = [(X_1, \dots, X_m), (X_{m+1}, \dots, X_{2m})], \exists X \Phi_A(X))$$

to determine if A is empty or not.

(2) If $A = \emptyset$, return

$$(\mathcal{P}_1, \Phi_X \vee \Phi_Y, \mathcal{P}_1(X_1, \ldots, X_m) \cup \mathcal{P}_1(X_{m+1}, \ldots, X_{2m}) \subset \mathbf{R}[X_1, \ldots, X_{2m}],$$

$$\Phi_{\mathrm{id}}(X, Y) \leftarrow (X = Y)).$$

(3) If $A \neq \emptyset$, set $M = m + 1$.

(4) Set $\Phi_\varphi(X, t, Y) \leftarrow 0 \leq t \leq 1 \wedge Y = X \cdot (1 - t)$.

(5) Apply the Extension Algorithm 2.2.1 with inputs

$$(\mathcal{P}_1, \Phi_X, \Phi_A, \mathcal{P}_1(X) \cup \mathcal{P}_1(Y) \subset \mathbf{R}[X_1, \ldots, X_m, Y_1, \ldots, Y_m, t], \Phi_\varphi, \mathcal{P}_2, \Phi_f)$$

to obtain a set of polynomials $\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ and a $\mathcal{Q}_1$-formula $\Phi_{f'}$ representing the graph of a semi-algebraic map f' from X to $\mathbf{R}^m$.

(6) We need to identify $X, Y,$ and A with their images in $X \amalg Y$. To this end we define formulae whose realizations are subsets of $\mathbf{R}^M$ as follows:

i. $\Phi_{X'}(X') \leftarrow \exists X \Phi_X(X) \wedge X' = (X, 1)$

ii. $\Phi_{Y'}(Y') \leftarrow \exists Y \Phi_Y(Y) \wedge Y' = (Y, 2)$

iii. $\Phi_{A'}(A') \leftarrow \exists A \Phi_A(A) \wedge A' = (A, 1)$

(7) We also need to redefine f' on the sets we have just defined:

$$\Phi_{f''}(X', Y') \leftarrow \exists X, Y$$

$$(\Phi_{X'}(X') \wedge \Phi_{Y'}(Y') \wedge X' = (X, 1) \wedge Y' = (Y, 2) \wedge \Phi_{f'}(X, Y)).$$

(8) Set

$$\widetilde{\Phi_p}(X, Z) \leftarrow \forall X_4 \exists X_1, X_2, X_3, Y_1, Y_2, Y_3, t, A, X_5, X_6$$

$$[(M_{X'}(X, X_1) \wedge M_{f''}(X, Y_1, X_2, Y_2) \wedge (M_{A'}(X_4, A) \wedge M_{X'}(X, X_5) \Rightarrow$$

$$||X_4 - X||^2 \geq t^2 \wedge M_{X'}(X_3, X_6)||X_3 - X||^2 = t^2)$$

$$\wedge Z = (Y_1, X \cdot t, t)) \vee (M_{Y'}(X, Y_3) \wedge Z = (X, 0, 0))].$$

(9) Apply theorem 7 with inputs

$$(\mathcal{P}_1 \cup \mathcal{Q}_1 \subset \mathbf{R}[(X_1, \ldots, X_{2m}, X'_1, \ldots, X'_{10m+4}],$$

$$\Pi = [(X_1, \ldots, X_{2m}), (X'_1, \ldots, X'_{m+1}), (X'_{m+2}, \ldots, X'_{10m+4})], \widetilde{\Phi_p})$$

to obtain a set of polynomials $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{3M+1}]$ and an equivalent quantifier free $\mathcal{Q}_2$-formula Φ_p.

(10) Set $\widetilde{\Phi_Z}(Z) \leftarrow \exists X \Phi_p(X, Z)$.

(11) Apply theorem 7 with inputs

$$(\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{3M+1}, X'_1, \ldots, X'_{m+1}],$$

$$\Pi = [(X_1, \ldots, X_{3M+1}), (X'_1, \ldots, X'_{m+1})], \widetilde{\Phi_Z})$$

to obtain a set of polynomials $\mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_{2M+1}]$ and an equivalent quantifier free $\mathcal{Q}_3$-formula Φ_Z.

(12) return$(\mathcal{Q}_3, \Phi_Z, \mathcal{Q}_2, \Phi_p)$.

Complexity Analysis for First Gluing Quotient Algorithm: We input a set of polynomials $\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m]$ of size k_1 whose degrees are bounded by d_1, and we input a set of polynomials $\mathcal{P}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ of size k_2 whose degrees are bounded by d_2.

(1) In step (1) we apply a decision algorithm to determine if the set A, described by the $\mathcal{P}_1$-formula Φ_A, is empty or not. This has complexity $(kd)^{\mathcal{O}(m)}$. If A is empty, we are done and this is the entire complexity of the algorithm.

(2) If A is not empty, we apply the Extension Algorithm 2.2.1 in step (5). This has complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

and returns a set of polynomials $\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ containing at most

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

polynomials whose degrees are bounded by $(d_2^{m^{\mathcal{O}(c)}} + (d_1)^{2^{\mathcal{O}(m)}})$.

(3) In step (9) we apply quantifier elimination to $\widetilde{\Phi_p}$. This has complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

and returns a set of polynomials $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{3m+4}]$ containing at most $\Big((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}}}$ polynomials whose degrees are bounded by $(d_2^{m^{\mathcal{O}(c)}} + (d_1)^{2^{\mathcal{O}(m)}})$.

(4) In step (11) we apply quantifier elimination to $\widetilde{\Phi_Z}$. This has complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

and returns a set of polynomials $\mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_{2m+3}]$ containing at most $\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$ polynomials whose degrees are bounded by $(d_2^{m^{\mathcal{O}(c)}} + (d_1)^{2^{\mathcal{O}(m)}})$.

(5) Therefore the total complexity of the algorithm is bounded by

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

Proof of Correctness of the First Gluing Quotient Algorithm: The correctness of the algorithm follows from Lemma 10.2.11 in [1], and from the correctness of Algorithms 14.3 and 14.5 in [15] and the Extension Algorithm 2.2.1.

Second Gluing Quotient Algorithm

By taking the above algorithm and applying the Completion Algorithm 2.2.1, we can obtain an algorithm that applies to a more general class of inputs. With this we obtain our second quotient algorithm that uses gluing. In this case, we input two sets of polynomials $\mathcal{P}_1$ and $\mathcal{P}_2$. We input the $\mathcal{P}_1$-formulae Φ_X, Φ_A, and Φ_Y, representing semi-algebraic sets X, A, and Y, with $A \subset X$ (here notice that A does not have to be closed). We input the $\mathcal{P}_2$-formula Φ_f, representing the graph of a semi-algebraic map $f : A \to Y$. Applying the previous algorithms we are able to obtain formulae Φ_Z, whose realization is a semi-algebraic set Z that is semi-algebraically homeomorphic to $X \amalg_f Y$, and Φ_p, representing the graph of the semi-algebraic map $p : X \amalg Y \to Z$.

We again present the statement of the corresponding theorem from van den Dries with a brief summary of the proof with appropriate links to the steps in the algorithm.

Theorem 13 (Theorem 10.2.12 [1])**.** *Suppose A is closed in X and $f : A \to Y$ is definably proper. Then $X \amalg_f Y$ exists as a definably proper quotient of $X \amalg Y$.*

We begin by applying lemma 10.2.7 from [1] to assume that X and Y are bounded in their ambient spaces and to extend f to a definable continuous map $\mathrm{cl}(f) : \mathrm{cl}(A) \to \mathrm{cl}(Y)$, line [4]. In order to apply lemma 10.2.7, we need a completion of Y, which we call $(\mu, Y_{bounded})$ in lines [1-2]. From 10.2.7, we obtain a function f' and sets X_{new}, X' and A', where X' is a completion of X, X_{new} and A' are the images of X and A, respectively, in X', and $f' : A' \to cl(Y_{bounded})$ extends f. We apply theorem 10.2.11 with inputs $X', A', cl(Y_{bounded})$, and f'. We obtain a space Z' and a quotient map $p' : X' \amalg cl(Y_{bounded}) \to Z'$, line [5]. We view $X_{new} \amalg Y_{bounded}$ as a subset of $X' \amalg \mathrm{cl}(Y_bounded)$ and define $Z = p'(X_{new} \amalg Y_{bounded})$, line [6]. Now $p'^{-1}(Z) = X_{new} \amalg Y_{bounded}$ and $E(f') \cap (X_new \amalg Y_{bounded})^2 = E(f)$. Therefore $p := p'|_{X_{new} \amalg Y_{bounded}} : X_{new} \amalg Y_{bounded} \to Z$, line [8], is a definably proper quotient of $X \amalg Y$ by $E(f)$.

<u>Second Gluing Quotient Algorithm</u>

Input($\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m]$, Φ_X a $\mathcal{P}_1$-formula describing a semi-algebraic set X, Φ_A a $\mathcal{P}_1$-formula describing a semi-algebraic subset A of X, Φ_Y a $\mathcal{P}_1$-formula describing a semi-algebraic set Y, $\mathcal{P}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$, Φ_f a $\mathcal{P}_2$-formula representing the graph of a semi-algebraic map f from A to Y)

Output($\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_{4m+3}]$, Φ_Z a $\mathcal{Q}_1$-formula describing a semi-algebraic quotient Z of $X \amalg Y$, $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{6m+4}]$, Φ_p a $\mathcal{Q}_2$-formula describing the graph of the quotient map from $X \amalg Y$ to $X \amalg_f Y = Z$)

Procedure:

(1) Set $\Phi_\mu(X_1, \ldots, X_m, Y_1, \ldots, Y_m) \leftarrow \bigwedge_{\mathrm{i}=1}^{m} \left((2X_\mathrm{i}Y_\mathrm{i} + 1)^2 = 1 + 4X_\mathrm{i}^2\right)$, a formula representing the graph of a semi-algebraic homeomorphism from $\mathbf{R}^m \to (-1, 1)^m$.

(2) Set $\widetilde{\Phi_{Y_{bounded}}}(Y) \leftarrow \exists X(\Phi_Y(X) \wedge \Phi_\mu(X, Y))$.

(3) Apply theorem 7 with inputs

$$(\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m, X'_1, \ldots, X'_m],$$

$$\Pi = [(X_1, \ldots, X_m), (X'_1, \ldots, X'_m)], \widetilde{\Phi_{Y_{bounded}}})$$

and

$$(\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m, \varepsilon, X'_1, \ldots, X'_m],$$

$$\Pi = [(X_1, \ldots, X_m), (\varepsilon), (X'_1, \ldots, X'_m)], \widetilde{\Phi_{cl(Y_{bounded})}})$$

to obtain two sets of polynomials. Let $\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_m]$ be the union of these sets of polynomials. We obtain equivalent quantifier free $\mathcal{Q}_1$-formulae $\Phi_{Y_{bounded}}$ and $\Phi_{cl(Y_{bounded})}$, respectively.

(4) Apply the Completion Algorithm 2.2.1 with inputs

$$(\mathcal{P}_1 \cup \mathcal{Q}_1, \Phi_X, \Phi_A \Phi_{Y_{bounded}}, \mathcal{P}_2, \Phi_f)$$

to obtain a set of polynomials $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ and $\mathcal{Q}_2$-formulae $\Phi_{X_{new}}$, $\Phi_{X'}$ and $\Phi_{A'}$, and a set of polynomials $\mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_{3m}]$ and a $\mathcal{Q}_3$-formula $\Phi_{f'}$, representing the graph of a semi-algebraic map from $r(\Phi_{A'})$ to $r(\Phi_{cl(Y_{bounded})})$.

(5) Apply the First Gluing Quotient Algorithm 2.2.2 with inputs

$$(\mathcal{Q}_1 \cup \mathcal{Q}_2, \Phi_{X'}, \Phi_{A'}, \Phi_{cl(Y_{bounded})}, \mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_{4m}], \Phi_{f'})$$

to obtain a set of polynomials $\mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_{4m+3}]$ and a $\mathcal{Q}_4$-formula $\Phi_{Z'}$ and a set of polynomials $\mathcal{Q}_5 \subset \mathbf{R}[X_1, \ldots, X_{6m+4}]$ and a $\mathcal{Q}_5$-formula $\Phi_{p'}$ representing the graph of a semi-algebraic map from $X' \amalg cl(Y)$ to $r(\Phi_{Z'})$.

(6) Set

$$\widetilde{\Phi_Z}(Z) = \exists X, Y \left[(\Phi_{X_{new}}(X) \wedge \Phi_{p'}(X, 1, Z)) \right.$$
$$\left. \vee (\Phi_{Y_{bounded}}(Y) \wedge \Phi_{p'}(Y, \underbrace{2, \ldots, 2}_{m+1}, Z)) \right].$$

(7) Apply theorem 7 with inputs

$$(\mathcal{Q}_1 \cup \mathcal{Q}_2 \cup \mathcal{Q}_5 \subset \mathbf{R}[X_1, \ldots, X_{6m+4}, X'_1, \ldots, X'_{3m}],$$

$$\Pi = [(X_1, \ldots, X_{6m+4}), (X'_1, \ldots, X'_{3m})], \widetilde{\Phi_Z})$$

to obtain a set of polynomials $\mathcal{Q}_6 \subset \mathbf{R}[X_1, \ldots, X_{4m+3}]$ and an equivalent quantifier free $\mathcal{Q}_6$-formula Φ_Z.

(8) Set

$$\widetilde{\Phi_p}(X, Z) \leftarrow \exists X', Y' (\Phi_{p'}(X', 1, Z) \wedge \Phi_{X_{new}}(X')$$

$$\wedge X = (X', 1)) \vee (\Phi_{p'}(X, Z) \wedge \Phi_{Y_{bounded}}(Y') \wedge X = (Y', \underbrace{2, \ldots, 2}_{m+1})).$$

(9) Apply theorem 7 with inputs

$$(\mathcal{Q}_1 \cup \mathcal{Q}_2 \cup \mathcal{Q}_5 \subset \mathbf{R}[X_1, \ldots, X_{6m+4}, X'_1, \ldots, X'_{3m}],$$

$$\Pi = [(X_1, \ldots, X_m), (X'_1, \ldots, X'_{3m})], \widetilde{\Phi_p})$$

to obtain a set of polynomials $\mathcal{Q}_7 \subset \mathbf{R}[X_1, \ldots, X_{6m+4}]$ and an equivalent quantifier free formula Φ_p.

(10) return $(\mathcal{Q}_6, \Phi_Z, \mathcal{Q}_5, \Phi_p)$.

Complexity Analysis for the Second Gluing Quotient Algorithm: We input a set of polynomials $\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m]$ of size k_1 whose degrees are bounded by d_1, and we input a set of polynomials $\mathcal{P}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ of size k_2 whose degrees are bounded by d_2.

(1) In step (3) we apply quantifier elimination to $\Phi_{\widetilde{Y_{bounded}}}$ and $\Phi_{\widetilde{cl(Y_{bounded})}}$. Each application adds complexity $(k_1 d_1)^{m^{\mathcal{O}(c)}}$ and returns a set of polynomials. We let $\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_m]$ be the union of the two returned sets of polynomials. $\mathcal{Q}_1$ contains at most $(k_1 d_1)^{m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $d_1^{m^{\mathcal{O}(c)}}$.

(2) In step (4) we apply the Completion Algorithm 2.2.1. This adds complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

and returns two set of polynomials. First $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ containing at most $\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$ polynomials of degree at most $(d_2^{m^{\mathcal{O}(c)}} + (d_1)^{2^{\mathcal{O}(m)}})$. Second $\mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_{3m}]$ with the same cardinality and degree bounds as $\mathcal{Q}_2$.

(3) In step (5) we apply the First Gluing Quotient Algorithm 2.2.2. Let $k_1^*, k_2^*, d_1^*, d_2^*$ equal the number of polynomials in $\mathcal{Q}_2$ and $\mathcal{Q}_3$ and their respective degree bounds. Then

$$k_1^* = k_2^* = \left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$$

and $d_1^* = d_2^* = \left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)$. Applying this algorithm has complexity

$$\left((k_2^*)^{m^{\mathcal{O}(c)}} + (mk_1^*d_1^*)^{2^{\mathcal{O}(m)}}\right)\left((d_2^*)^{m^{\mathcal{O}(c)}} + (d_1^*)^{2^{\mathcal{O}(m)}}\right)$$

Because $k_1^* = k_2^*$ and $d_1^* = d_2^*$, this complexity simplifies to $(mk_1^*d_1^*)^{2^{\mathcal{O}(m)}}$. Substituting the values for k_1^* and d_1^* gives the following, after some simplification:

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}} \left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}}$$

This returns two sets of polynomials: $\mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_{4m+3}]$ and $\mathcal{Q}_5 \subset \mathbf{R}[X_1, \ldots, X_{6m+4}]$. Both sets of polynomials have cardinality on the order $\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}} \left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}}$ with degrees bounded by $\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(}}$

(4) In step (7) we apply quantifier elimination to $\widetilde{\Phi_Z}$. This adds complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}} \left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}}$$

and returns a set of polynomials $\mathcal{Q}_6 \subset \mathbf{R}[X_1, \ldots, X_{4m+3}]$ containing at most

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}} \left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}}$$

polynomials whose degrees are bounded by $\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}}$.

(5) In step (9) we apply quantifier elimination to $\widetilde{\Phi_p}$. This adds complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}} \left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}}$$

and returns a set of polynomials $\mathcal{Q}_7 \subset \mathbf{R}[X_1, \ldots, X_{6m+4}]$ containing at most

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}} \left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}}$$

polynomials whose degrees are bounded by $\left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}}$.

(6) Therefore the entire algorithm has complexity

$$\left((k_2)^{m^{\mathcal{O}(c)}} + (mk_1d_1)^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}} \left(d_2^{m^{\mathcal{O}(c)}} + d_1^{2^{\mathcal{O}(m)}}\right)^{2^{\mathcal{O}(m)}}$$

Proof of Correctness of the Second Gluing Quotient Algorithm: The correctness of the algorithm follows from Proposition 10.2.12 in [1], and from the correctness of Algorithm 14.5 [15], the Completion Algorithm 2.2.1, and the First Gluing Quotient Algorithm 2.2.2.

General Quotient Algorithm

Finally, with all these preliminary algorithms out of the way, we have the tools we need to present our final algorithm. This algorithm is able to take any semi-algebraic set X and any semi-algebraically proper equivalence relation E on X and produces the quotient space X/E. More specifically, we input two sets of polynomials $\mathcal{P}_1$ and $\mathcal{P}_2$. We input a $\mathcal{P}_1$-formula Φ_X, describing a semi-algebraic set X, and a $\mathcal{P}_2$-formula Φ_E, representing a semi-algebraically proper equivalence relation E on X. We are able to produce a formula $\Phi_{X/E}$, whose realization is semi-algebraically homeomorphic to the quotient space X/E, and a formula Φ_f, representing the graph of a semi-algebraic map f from X to the realization of $\Phi_{X/E}$.

Recall that this algorithm makes effective theorem 5.

<u>**General Quotient Algorithm**</u>

Input($\mathcal{P}_1 \subset \mathbf{R}[X_1, \ldots, X_m]$, Φ_X a $\mathcal{P}_1$-formula describing a semi-algebraic set X, $\mathcal{P}_2 \subset \mathbf{R}[X_1, \cdots X_{2m}]$, Φ_E a $\mathcal{P}_2$-formula describing an equivalence relation $E \subset X \times X$ on X)

Output($\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots,]$, Φ_f a $\mathcal{Q}_1$-formula describing the graph of the map from X to the semi-algebraic realization of X/E, $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots,]$, $\Phi_{X/E}$ a $\mathcal{Q}_2$-formula describing semi-algebraic realization of the quotient space of X under the equivalence relation E)

Procedure:

(1) Apply Algorithm 4 from [24] with inputs $(\mathcal{P}_1, \Phi_X)$ to calculate $D = \dim(X)$.

(2) If $D = 0$, perform the following:

  (a) Set $\widetilde{\Phi_f}(X, Y) \leftarrow \forall Z(\Phi_E(X, Y) \wedge \neg(\Phi_E(X, Z) \wedge Z < Y)$.

(b) Apply theorem 7 with inputs

$$(\mathcal{P}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}, X'_1, \ldots, X'_m],$$

$$\Pi = [(X_1, \ldots, X_{2m}), (X'_1, \ldots, X'_m)], \widetilde{\Phi_f})$$

to obtain a set of polynomials $\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ and an equivalent quantifier free $\mathcal{Q}_1$-formula Φ_f.

(c) Set $\widetilde{\Phi_{X/E}}(Y) \leftarrow \exists X \Phi_f(X, Y)$.

(d) Apply theorem 7 with inputs

$$(\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_m, X'_1, \ldots, X'_m],$$

$$\Pi = [(X_1, \ldots, X_m), (X'_1, \ldots, X'_m)], \widetilde{\Phi_{(X/E)}})$$

to obtain a set of polynomials $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_m]$ and an equivalent quantifier free $\mathcal{Q}_2$-formula $\Phi_{X/E}$.

(e) return $(\mathcal{Q}_1, \Phi_f, \mathcal{Q}_2, \Phi_{X/E})$.

(3) If $D > 0$, perform the following:

(a) Apply the triangulation algorithm, theorem 8 with inputs $(\mathcal{P}_1, \Phi_X)$ to obtain a triangulation (h, K) of X. We obtain sets of polynomials $\mathcal{Q}_\sigma$ and $\mathcal{Q}_\sigma$-formulae $\Phi_{K,\sigma}$, for each simplex σ of K. Let $\mathcal{Q}_2 = \bigcup_\sigma \mathcal{Q}_\sigma \subset \mathbf{R}[X_1, \ldots, X_{2m+1}]$.

(b) Set $\Phi_S(X) \leftarrow \forall Z(\Phi_E(X, Z) \Rightarrow X \leq Z)$.

(c) Set $\widetilde{\Phi_{S_D}}(X) \leftarrow \exists t_0, \ldots, t_D \forall Z \bigvee_{\substack{\sigma \in K \\ \dim(\sigma) = D}} \Phi_{K,\sigma}(t_0, \ldots, t_D, X) \wedge M_S(X, Z)$.

(d) Apply theorem 7 with inputs

$$(\mathcal{P}_2 \cup \mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}, X'_1, \ldots, X'_{m+D+1}],$$

$$\Pi = [(X_1, \ldots, X_{2m}), (X'_1, \ldots, X'_m), (X'_{m+1}, \ldots, X'_{m+D+1})], \widetilde{\Phi_{S_D}})$$

to obtain a set of polynomials $\mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_m]$ and an equivalent quantifier free $\mathcal{Q}_3$-formula Φ_{S_D}.

(e) Set $\widetilde{\Phi_{X'}}(X) \leftarrow \forall \varepsilon > 0 \exists X'(M_{\mathrm{cl}(S)}(X, X', \varepsilon) \wedge \neg \Phi_{S_D}(X))$.

(f) Apply theorem 7 with inputs

$$(\mathcal{P}_2 \cup \mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_{2m}, X'_1, \ldots, X'_m, \varepsilon],$$

$$\Pi = [(X_1, \ldots, X_{2m}), (X'_1, \ldots, X'_m), (\varepsilon)], \widetilde{\Phi_{X'}})$$

to obtain a set of polynomials $\mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_m]$ and an equivalent quantifier free $\mathcal{Q}_4$-formula $\Phi_{X'}$.

(g) Set $\Phi_{E'}(X, Y) \leftarrow \Phi_E(X, Y) \wedge \Phi_{X'}(X) \wedge \Phi_{X'}(Y)$.

(h) Apply the General Quotient Algorithm 2.2.2 with inputs

$$(\mathcal{Q}_4, \Phi_{X'}, \mathcal{P}_2 \cup \mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_{2m}], \Phi_{E'})$$

to obtain two sets of polynomials $\mathcal{Q}_5 \subset \mathbf{R}[X_1, \ldots, X_{4^{D-1}m+4^{D-1}-1}]$ and $\mathcal{Q}_6 \subset \mathbf{R}[X_1, \ldots, X_{(4^{D-1}+1)m+4^{D-1}-1}]$ and a $\mathcal{Q}_5$-formula $\Phi_{Y'}$ and a $\mathcal{Q}_6$-formula $\Phi_{f'}$, representing the graph of the semi-algebraic quotient map from $X' \to Y'$.

(i) Apply theorem 7 with inputs

$$(\mathcal{P}_1 \cup \mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_m, X'_1, \ldots, X'_m, \varepsilon],$$

$$\Pi = [(X_1, \ldots, X_m), (X'_1, \ldots, X'_m), (\varepsilon)],$$

$$\forall \varepsilon > 0 \exists X' M_{\mathrm{cl}(S_D)}(X, X', \varepsilon) \wedge \Phi_X(X)]$$

to obtain a set of polynomials $\mathcal{Q}_7 \subset \mathbf{R}[X_1, \ldots, X_m]$ and an equivalent quantifer free formula $\Phi_{\mathrm{cl}(S_D)}$.

(j) Set $\Phi_A(X) \leftarrow \Phi_{cl(S_D)}(X) \wedge \Phi_{X'}(X)$.

(k) Set $\Phi_{f''}(X, Y) \leftarrow \Phi_{f'}(X, Y) \wedge \Phi_A(X)$.

(l) Apply the Second Gluing Quotient Algorithm 2.2.2 with inputs

$$(\mathcal{Q}_4 \cup \mathcal{Q}_5 \cup \mathcal{Q}_7, \Phi_{cl(S_D)}, \Phi_A, \Phi_{Y'}, \mathcal{Q}_6 \cup \mathcal{Q}_7, \Phi_{f''})$$

to obtain two sets of polynomials $\mathcal{Q}_8 \subset \mathbf{R}[X_1, \ldots, X_{4^D m + 4^D - 1}]$ and $\mathcal{Q}_9 \subset \mathbf{R}[X_1, \ldots, X_{2(4^D m + 4^D - 1)}]$ and a $\mathcal{Q}_8$-formula $\Phi_{X/E}$ and a $\mathcal{Q}_9$-formula Φ_p, representing the graph of the semialgebraic quotient map from $cl(S_D) \amalg Y'$ to the quotient space $X/E = cl(S_D) \amalg_{f''} Y'$.

(m) Set

$$\Phi_g(X, Y) \leftarrow \exists Z((\Phi_{cl(S_D)}(X) \wedge \Phi_p(X, 1, \ldots, 1, Y))$$

$$\vee(\Phi_{X'}(X) \wedge \Phi_{f'}(X, Z) \wedge \Phi_p(Z, 2, Y))).$$

(n) Set

$$\widetilde{\Phi_f}(X, Y) \leftarrow \exists S, Z[\Phi_X(X) \wedge \Phi_{X/E}(Y)$$

$$\wedge \Phi_S(S) \wedge \Phi_E(X, S) \wedge M_g(S, Y, Z)].$$

(o) Apply theorem 7 with inputs

$$(\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{Q}_6 \cup \mathcal{Q}_7 \cup \mathcal{Q}_8 \cup \mathcal{Q}_9$$

$$\subset \mathbf{R}[X_1, \ldots, X_{2(4^D m + 4^D - 1)}, X'_1, \ldots, X'_{(4^D+1)m + 4^D - 1}],$$

$$\Pi = [(X_1, \ldots, X_{2(4^D m + 4^D - 1)}), (X'_1, \ldots, X'_{(4^D+1)m + 4^D - 1})], \widetilde{\Phi_f})$$

to obtain a set of polynomials $\mathcal{Q}_{10} \subset \mathbf{R}[X_1, \ldots, X_{(4^D+1)m + 4^D - 1}]$ and an equivalent quantifier free $\mathcal{Q}_{10}$-formula Φ_f.

(p) return $(\mathcal{Q}_8, \Phi_{X/E}, \mathcal{Q}_{10}, \Phi_f)$.

Complexity Analysis for the General Quotient Algorithm:

We input a set $\mathcal{P}_1$ of k_1 polynomials in m variables with degree at most d_1 and a set $\mathcal{P}_2$ of k_2 polynomials in $2m$ variables with degree at most d_2.

(1) In step (1) we compute the dimension of our space X. This step has complexity $(k_1 d_1)^{m^{\mathcal{O}(c)}}$.

(2) If $\dim(X) = D = 0$, then the only thing we have to do is apply quantifier elimination twice. We do this in step (2):

- In step (2b), we apply quantifier elimination to $\widetilde{\Phi_f}$. This step has complexity $(k_2 d_2)^{m^{\mathcal{O}(c)}}$ and returns a set of polynomials $\mathcal{Q}_1 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ containing at most $(k_2 d_2)^{m^{\mathcal{O}(c)}}$ polynomials of degree at most $d_2^{m^{\mathcal{O}(c)}}$.
- In step (2d), we apply quantifier elimination to $\widetilde{\Phi_{X/E}}$. This step has complexity $(k_2 d_2)^{m^{\mathcal{O}(c)}}$ and returns a set of polynomials $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ containing at most $(k_2 d_2)^{m^{\mathcal{O}(c)}}$ polynomials of degree at most $d_2^{m^{\mathcal{O}(c)}}$.

(3) In the case where $D > 0$, we must first generate spaces that are homeomorphic to subspaces of X of lower dimension, until we generate a space of dimension 0, then we can apply the previous line.

(4) In step (3a), we apply the triangulation algorithm to $(\mathcal{P}_1, \Phi_X)$. This step has complexity $(k_1 d_1)^{2^{\mathcal{O}(m)}}$ and returns a set of polynomials $\mathcal{Q}_2 \subset \mathbf{R}[X_1, \ldots, X_{2m+1}]$ containing at most $(k_1 d_1)^{2^{\mathcal{O}(m)}}$ polynomials of degree at most $d_1^{2^{\mathcal{O}(m)}}$.

(5) In step (3d), we apply quantifier elimination to $\widetilde{\Phi_{S_D}}$. Since $D \leq m$, this has complexity

$$\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{m^{\mathcal{O}(c)}}$$

and returns a set of polynomials $\mathcal{Q}_3 \subset \mathbf{R}[X_1, \ldots, X_m]$ containing at most $\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_2 + d_1^{2^{\mathcal{O}(m)}})^{m^{\mathcal{O}(c)}}$.

(6) In step (3f), we apply quantifier elimination to $\widetilde{\Phi_{X'}}$ This step has complexity

$$\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{m^{\mathcal{O}(c)}}$$

and returns a set of polynomials $\mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_m]$ containing at most $\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_2 + d_1^{2^{\mathcal{O}(m)}})^{m^{\mathcal{O}(c)}}$.

(7) In step (3h) we apply the General Quotient Algorithm on a space of dimension at most $D-1$, inputting the set of polynomials $\mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_m]$ containing $\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{m^{\mathcal{O}(c)}}$ polynomials with degrees bounded by $(d_2 + d_1^{2^{\mathcal{O}(m)}})^{m^{\mathcal{O}(c)}}$ and the set of polynomials $\mathcal{P}_2 \cup \mathcal{Q}_4 \subset \mathbf{R}[X_1, \ldots, X_{2m}]$ with the same cardinality and degree bound. In applying the General Quotient Algorithm, we first apply items (4-6) up to D times (because each call of the General Quotient Algorithm will call the General Quotient Algorithm after only completing items (4-6)). If we apply items (4-6) with $k_1^* = k_2^* = |\mathcal{Q}_4|$ and $d_1^* = d_2^* =$ the degree bounds of $\mathcal{Q}_4$, then we have complexity:

$$(k_1^* d_1^*)^{2^{\mathcal{O}(m)}}$$

Plugging in the values of k_1^* and d_1^* gives:

$$\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{2^{\mathcal{O}(m)}}$$

From this analysis, we see that with each iteration through the first part of the algorithm we end up with another copy of $2^{\mathcal{O}(m)}$ in the exponent. Hence, after up to D iterations of items (4-6) the complexity will be

$$\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{2^{\mathcal{O}(Dm)}} \leq \left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{2^{\mathcal{O}(m^2)}}$$

At this point, we will have two sets of polynomials (call them $\mathcal{R}, \mathcal{S}$) in m and $2m$ variables, respectively. They will both contain

$$\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{2^{\mathcal{O}(m^2)}}$$

polynomials with degrees bounded by

$$(d_2 + d_1^{2^{\mathcal{O}(m)}})^{2^{\mathcal{O}(m^2)}}.$$

From here we have reached a space of dimension 1 and can apply the rest of the algorithm up to D times to finish the application of the General Quotient Algorithm on the space of dimension $D-1$. The first step is to apply quantifier elimination to $\widetilde{\Phi_{cl(S_1)}}$. At each iteration, the complexity of this step is dominated by the complexity of the first half of iterating on the General Quotient Algorithm, so we omit its complexity (we will see below what affect this step has on the original space). From here we must apply the Second Gluing Quotient Algorithm with sets of polynomials that have the same order of polynomials and degree bounds as $\mathcal{R}, \mathcal{S}$. Now let $k_1^* = k_2^* = |\mathcal{R}|$ and $d_1^* = d_2^* =$ degree bounds of $\mathcal{R}$. Then this step has complexity:

$$(mk_1^* d_1^*)^{2^{\mathcal{O}(2m)}}$$

If we plug in the value for k_1^* and d_1^*, we see that the complexity for the Second Gluing Quotient Algorithm in the first iteration is:

$$\left(m(k_2 + k_1^{2^{\mathcal{O}(m)}})^{2^{\mathcal{O}(m^2)}}(d_2 + d_1^{2^{\mathcal{O}(m)}})^{2^{\mathcal{O}(m^2)}}\right]^{2^{\mathcal{O}(2m)}}$$

We then finish the iteration by applying quantifier elimination to $\widetilde{\Phi_f}$. This doesn't affect the order of the complexity. From this, we see that each iteration increase the exponents $2^{\mathcal{O}(m)}$ on the m term, the $(k_2 + k_1^{2^{\mathcal{O}(m)}})$ term, and the $(d_2 + d_1^{2^{\mathcal{O}(m)}})$ term. In addition, if we input polynomials in m variables into the Second Gluing Quotient Algorithm, we output a polynomial in $4m+3$ variables. Hence after up to D iterations, our polynomials will be in $4^D m + 4^D - 1 \sim \mathcal{O}(4^D m)$ variables. Therefore, the total complexity after up to D iterations of reapplying algorithm 10.2.15 will be:

$$\mathcal{C} = (4^D m)^{2^{\mathcal{O}(Dm)}}(k_2 + k_1^{2^{\mathcal{O}(m)}})^{2^{\mathcal{O}(m^2 D)}}(d_2 + d_1^{2^{\mathcal{O}(m)}})^{2^{\mathcal{O}(m^2 D)}}$$

In addition, this will return two sets of polynomials $\mathcal{Q}_5 \subset \mathbf{R}[X_1, \ldots, X_{4^{D-1}m+4^{D-1}-1}]$ and $\mathcal{Q}_6 \subset \mathbf{R}[X_1, \ldots, X_{(4^{D-1}+1)m+4^{D-1}-1}]$ each containing on the order of $\mathcal{C}$ polynomials of degree bounded by $(d_2 + d_1^{2^{\mathcal{O}(m)}})^{2^{\mathcal{O}(Dm^2)}}$.

(8) In step (3i) we apply quantifier elimination to the formula

$$\forall \varepsilon > 0 \exists X' M_{cl(S_D)}(X, X', \varepsilon) \wedge \Phi_X(X).$$

This has complexity

$$\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{m^{\mathcal{O}(c)}}$$

and returns a set of polynomials $\mathcal{Q}_7 \subset \mathbf{R}[X_1, \ldots, X_m]$ containing at most $\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{m^{\mathcal{O}(c)}}$ polynomials with degrees bounded by $(d_2 + d_1^{2^{\mathcal{O}(m)}})^{m^{\mathcal{O}(c)}}$. This shows that, as we stated above, the complexity of this step will pail in comparison to the complexity of the previous step for any D.

(9) In step (3l) we apply the Second Gluing Quotient Algorithm one final time. As we saw in our previous analysis, this will just add factors of $2^{\mathcal{O}(m)}$, so the complexity of this step is still $\mathcal{C}$. It returns a set of polynomials $\mathcal{Q}_8 \subset \mathbf{R}[X_1, \ldots, X_{4^Dm+4^D-1}]$ and $\mathcal{Q}_9 \subset \mathbf{R}[X_1, \ldots, X_{2(4^Dm+4^D-1)}]$ containing on the order of $\mathcal{C}$ polynomials of degree at most $(d_2 + d_1^{2^{\mathcal{O}(m)}})^{2^{\mathcal{O}(Dm^2)}}$.

(10) In step (3o) we apply quantifier elimination to $\widetilde{\Phi_f}$. Again this still has complexity on the order $\mathcal{C}$ and returns a set of polynomials $\mathcal{Q}_{10} \subset \mathbf{R}[X_1, \ldots, X_{(4^D+1)m+4^D-1}]$ containing $\mathcal{C}$ polynomials whose degrees are bounded by $(d_2 + d_1^{2^{\mathcal{O}(m)}})^{D^2 m^{\mathcal{O}(D^2)} 2^{\mathcal{O}(D^2 m)}}$.

(11) Therefore the entire complexity of the algorithm is

$$(4^D m)^{2^{\mathcal{O}(Dm)}} (k_2 + k_1^{2^{\mathcal{O}(m)}})^{2^{\mathcal{O}(m^2 D)}} (d_2 + d_1^{2^{\mathcal{O}(m)}})^{2^{\mathcal{O}(m^2 D)}}$$

(12) To simplify this, we can use the fact that $D \leq m$, and separate into the cases where $k_2 \approx k_1^{2^{\mathcal{O}(m)}}$ and $d_2 \approx d_1^{2^{\mathcal{O}(m)}}$, or $k_2 >> k_1^{2^{\mathcal{O}(m)}}$ and $d_2 >> d_1^{2^{\mathcal{O}(m)}}$, or $k_2 << k_1^{2^{\mathcal{O}(m)}}$ and $d_2 << d_1^{2^{\mathcal{O}(m)}}$. After simplification, an upper bound on the complexity becomes:

$$\begin{cases} (mkd)^{2^{\mathcal{O}(m^3)}} & \text{if } k_1^{2^{\mathcal{O}(m)}} \approx k_2 (\approx k) \text{ and } d_1^{2^{\mathcal{O}(m)}} \approx d_2 (\approx d) \\ (mk_2d_2)^{2^{\mathcal{O}(m^3)}} & \text{if } k_2 >> k_1^{2^{\mathcal{O}(m^3)}} \text{ and } d_2 >> d_1^{2^{\mathcal{O}(m)}} \\ (mk_1d_1)^{2^{\mathcal{O}(m^4)}} & \text{if } k_1^{2^{\mathcal{O}(m)}} >> k_2 \text{ and } d_1^{2^{\mathcal{O}(m)}} >> d_2 \end{cases}$$

Proof of Correctness of the General Quotient Algorithm: The correctness of the algorithm follows from the proof Theorem 10.2.15 in [1] and from the correctness of Algorithm 4 in [24], Algorithm 14.5 in [15], and the Second Gluing Quotient Algorithm 2.2.2.

2.2.3 Improved Bounds

As has been stated before, it is often the case that upper bounds on topological complexity of objects are closely related to the worst case scenario complexity of algorithms computing topological invariants of such objects. In light of this correlation, we present in this section an argument that would imply that a singly exponential algorithm to compute quotients exists. Finding such an algorithm is left for future work.

Definition 15. *For two maps $f_1 : X_1 \to Y$ and $f_2 : X_2 \to Y$, the* **fibred product** *of X_1 and X_2 is defined as*

$$X_1 \times_Y X_2 := \{(x_1, x_2) \in X_1 \times X_2 | f_1(x_1) = f_2(x_2)\}$$

Gabrielov et. al. were able to prove the following complexity bound on fibred products when $f_1 = f_2$.

Theorem 14. *[25, Theorem 1] Let $f : X \to Y$ be a closed surjective cellular map. There exists a spectral sequence $E^r_{p,q}$ converging to $H_*(Y)$ with*

$$E^1_{p,q} = H_q(W_p)$$

where

$$W_p = \underbrace{X \times_Y \cdots \times_Y X}_{p+1 \text{ times}}.$$

In particular,

$$b_k(Y) \leq \sum_{p+q=k} b_q(W_p)$$

for all k.

For our purposes, let $f = \boldsymbol{\pi} : X \to X/E$ be the standard quotient map. If E is a definably proper equivalence relation, then $\boldsymbol{\pi}$ is a proper map. Therefore by Theorem 14, we have for any k

$$b_k(\boldsymbol{\pi}(X)) = b_k(X/E) \leq \sum_{p+q=k} b_q(X \times_{X/E} \cdots \times_{X/E} X).$$

Let $X \subset \mathbf{R}^n$ be defined by a $\mathcal{P}$-formula $\phi(\cdot)$, for some set of polynomials $\mathcal{P}$, and $E \subset \mathbf{R}^n \times \mathbf{R}^n$ be defined by a $\mathcal{Q}$-formula $\psi(\cdot, \cdot)$. Using these formulae, $\underbrace{X \times_{\boldsymbol{\pi}} \cdots \times_{\boldsymbol{\pi}} X}_{p+1 \text{ times}}$ is defined by the formula

$$\bigwedge_{0 \leq \mathrm{i} < p} \psi(x^{(\mathrm{i})}, x^{(\mathrm{i}+1)}) \wedge \bigwedge_{\mathrm{i}=0}^{p} \phi(x^{(\mathrm{i})})$$

where there are p polynomials in $\mathcal{P}$ and q polynomials in $\mathcal{Q}$, and the degrees of the polynomials in $\mathcal{P}$ and $\mathcal{Q}$ are both bounded by d. It would then take $pq + (p+1)q$ polynomials of degree less than or equal to d in $(p+1)n$ variables to describe $\underbrace{X \times_{\boldsymbol{\pi}} \cdots \times_{\boldsymbol{\pi}} X}_{p+1 \text{ times}}$. Using results from [26] and [25], we get a betti number bound for the fibred product, and hence for X/E, as

$$b(X/E) \leq \mathcal{O}(nsd)^{(n+1)n}.$$

This bound is singly exponential, as desired, which leads one to suspect that a singly exponential algorithm to compute quotient spaces is possible.

2.3 Conclusion

In this chapter, we have shown that the given a $\mathcal{P}_1$-formula Φ_X, representing a semi-algebraic set X, and a $\mathcal{P}_2$-formula Φ_E, representing an equivalence relation on E, there exists a formula $\Phi_{X/E}$ with complexity that is doubly exponential in $\dim(X)$, $|\mathcal{P}_1|$, $|\mathcal{P}_2|$, the degree bounds of $\mathcal{P}_1$ and $\mathcal{P}_2$, and in the number of variables in $\mathcal{P}_1$ and $\mathcal{P}_2$. We have presented algorithms that make effective the method of obtaining this formula. However, doubly exponential complexity is never desirable if it can be avoided, so future work in this area can explore the possibility of finding an algorithm that runs in singly exponential time.

3. HARMONIC CHAINS

3.1 Background Definitions

This chapter of the thesis is devoted to developing a theoretical framework to using the harmonic chains of a simplicial complex as a means of determining a cycle to represent a homology class. We proceed to applying this framework to persistent homology. We use harmonic chains as a means to associate cycles with each bar of the persistent bar code. We finish by proving two stability results on functions that represent the space of harmonic chains.

3.1.1 Linear Algebra Facts

We begin this chapter by recalling some basic facts about linear algebra. Let V be a finite dimensional vector space over $\mathbb{R}$. We denote the dual of V by $V^* = \mathrm{Hom}_{\mathbb{R}}(V, \mathbb{R})$. For any subspace $W \subset V$, we denote

$$W^o = \{v^* \in V^* \mid v^*(w) = 0, \text{ for all } w \in W\}.$$

If U is an $\mathbb{R}$-vector space and $L : U \to V$ a linear map, then

$$\mathrm{Im}(L)^o = \ker({}^tL) \subset V^*. \tag{3.1.1}$$

If V is a Euclidean space equipped with an inner product $\langle \cdot, \cdot \rangle$, for any subspace $W \subset V$, we let $W^\perp$ denote the orthogonal complement of W in V. $\mathrm{proj}_W : V \to W$ shall denote the orthogonal projection to V.

There exists an isomorphism $\phi_V : V^* \to V$ which maps $v^* \in V^*$ to the uniquely defined $v \in V$, such that $v^*(v) = \langle v, v \rangle$ for every $v \in V$.

If $\mathcal{A} = \{\mathrm{e}_1, \ldots, \mathrm{e}_n\}$ is an orthonormal basis of V, and $\mathcal{A}^* = \{\mathrm{e}_1^*, \ldots, \mathrm{e}_n^*\}$ denotes the basis of V^* dual to $\mathcal{A}$, then for any $v \in V$, the coordinate $[v]_{\mathcal{A}} \in \mathbb{R}^n$ of v with respect to the basis $\mathcal{A}$, equals the coordinate $[\phi_V^{-1}(v)]_{\mathcal{A}^*} \in \mathbb{R}^n$ of $\phi_V^{-1}(v)$ with respect to the basis $\mathcal{A}^*$. In particular, the matrix of ϕ_V with respect to the bases $\mathcal{A}^*$ and $\mathcal{A}$ is the $n \times n$ identity matrix.

It is easy to see that for any subspace $W \subset V$, $\phi_V(W^o) = W^\perp$.

Thus identifying V and V^* using the isomorphism ϕ, we can for any subspace $W \subset V$, identify $W^o \subset V^*$ with $W^\perp \subset V$.

Using (3.1.1), for any linear map $L : U \to V$,

$$\operatorname{Im}(L)^\perp = \phi_V(\ker({}^tL)).$$

The following two lemmas in linear algebra will prove useful later, so we record them here for reference.

Lemma 1. *For any subspace $W \subset V$, there exists a canonically defined isomorphism $V/W \to W^\perp$, defined by*

$$v + W \mapsto \operatorname{proj}_{W^\perp}(v).$$

Lemma 2. *If $W_1, W_2 \subset V$ are subspaces with $W_1 \subset W_2$, then for any $w \in W_2$, $\operatorname{proj}_{W_1^\perp}(w) \in W_2$.*

3.1.2 Persistent Homology

We transition now to an introduction to persistent homology. [17] is a good source for the interested reader to learn more about persistent homology and its applications in computational topology. Persistent homology can be defined on more general topological spaces, but for our purposes we restrict our spaces to simplicial complexes.

Definition 16 (Filtrations)**.** *Let S be an ordered set and K be a finite simplicial complex. The tuple $\mathcal{F} = (K_s)_{s\in S}$ of subcomplexes of K such that whenever $s < t$, $K_s \subset K_t$, is called a* **filtration** *of K.*

Furthermore suppose K is equipped with a map $f : K \to \mathbb{R}$ such that $f(\sigma) < f(\tau)$ whenever σ is a face of τ. Enumerate the simplices of K, $\{\sigma_{s_1}, \ldots, \sigma_{s_n}\}_{s_i\in S}$, such that for $s \leq t$, $f(\sigma_s) \leq f(\sigma_t)$. Set $K_s = f^{-1}(-\infty, f(\sigma_s)]$. The conditions on f guarantee that K_s is a subcomplex of K for each s. The sequence

$$\emptyset = K_0 \subseteq K_1 \subset \cdots \subset K_n = K$$

is a filtration induced by f and we denote it by $\mathcal{F}_f$. If f is injective, then $K_s \setminus K_{s-1} = \{\sigma_s\}$ for each $s \in \{1, \ldots, n\}$. In this case, we say $\mathcal{F}_f$ is a **simplex-wise** *filtration.*

In the case of simplex-wise filtrations, a common choice of function is to have $f(\sigma_s) = s$ for each $s \in S$. Useful filtrations for applications are Čech and alpha complexes and lower star filtrations. We do not consider these filtrations, so the reader is directed to [17, Chapters 3 and 4] for more information.

Given a filtration, the inclusion maps, $K_s \hookrightarrow K_t$ for $s \leq t$, between the complexes induce maps on homology. Thus, a filtration corresponds to the following sequence of homology groups for each $p \geq 0$:

$$0 = H_p(K_0) \to H_p(K_1) \to \cdots \to H_p(K_n) = H_p(K)$$

Definition 17 (Persistent Homology Groups)**.** *Given a filtration $\mathcal{F}$ on a simplicial complex K, let $\mathrm{i}_p^{s,t} : H_p(K_s) \to H_p(K_t)$ be the map induced by the inclusion $K_s \hookrightarrow K_t$ for each $s \leq t$ and $p \geq 0$. The* **p-th persistent homology groups** *of a simplicial complex K are the images of $\mathrm{i}_p^{s,t}$, denoted $H_p^{s,t}(\mathcal{F}) = \mathrm{im}(\mathrm{i}_p^{s,t})$. The* **$p$-th persistent Betti numbers** *are the dimensions of these groups: $b_p^{s,t} = \dim(H_p^{s,t}(\mathcal{F}))$.*

Note that $H_p^{s,t}(\mathcal{F}) \subset H_p(K_t)$ and that for $s = t$, $H_p^{s,s}(\mathcal{F}) = H_p(K_s)$.

As the dimension of the homology groups in the sequence change, homology classes appear and disappear. We refer to this as the birth and death of a homology class and define these ideas more rigorously below.

Definition 18 (Birth and Death)**.** *For $s \leq t \in S$ and $p \geq 0$, let $\gamma \in H_p(K_s)$. We say that γ is* **born** *at K_s if $\gamma \notin H_p^{s-1,s}$. Furthermore if γ is born at K_s, it* **dies entering** *K_t if $\mathrm{i}_p^{s,t-1}(\gamma) \notin H_p^{s-1,t-1}$ but $\mathrm{i}_p^{s,t}(\gamma) \in H_p^{s-1,t}$.*

Remark 2. Note that the homology classes that are born at time s, and those that are born at time s and die entering time t, as defined above are not subspaces of $H_p(K_s)$. In order to be able to associate a "multiplicity" to the set of homology classes which are born at time s and dies at time t we interpret them as classes in certain *subquotients* of $H_*(K_s)$ in what follows.

First observe that it follows from Definition 17 that for all $s' \leq s \leq t$ and $p \geq 0$, $H_p^{s',t}(\mathcal{F})$ is a subspace of $H_p^{s,t}(\mathcal{F})$, and both are subspaces of $H_p(K_t)$. This is because the homomorphism $\mathrm{i}_p^{s',t} = \mathrm{i}_p^{s,t} \circ \mathrm{i}_p^{s',s}$, and so the image of $\mathrm{i}_p^{s',t}$ is contained in the image of $\mathrm{i}_p^{s,t}$. It follows that, for $s \leq t$, $\bigcup_{s'<s} H_p^{s',t}(\mathcal{F})$ is an increasing union of subspaces, and is itself a subspace of $H_p(K_t)$. In particular, setting $t = s$, $\bigcup_{s'<s} H^{s',s}(\mathcal{F})$ is a subspace of $H_p(K_s)$.

The following definition is taken from [27]. With the same notation as above:

Definition 3.1.1. *For $s \leq t$, and $p \geq 0$, we define*

$$\begin{aligned}
L_p^s(\mathcal{F}) &= H_p(K_s)/\bigcup_{s'<s} H_p^{s',s}(\mathcal{F}), \\
M_p^{s,t}(\mathcal{F}) &= \bigcup_{s'<s} (\mathrm{i}_p^{s,t})^{-1}(\mathrm{H}_p^{s',t}(\mathcal{F})), \\
N_p^{s,t}(\mathcal{F}) &= \bigcup_{s'<s\leq t'<t} (\mathrm{i}_p^{s,t'})^{-1}(H_p^{s',t'}(\mathcal{F})), \\
P_p^{s,t}(\mathcal{F}) &= M_p^{s,t}(\mathcal{F})/N_p^{s,t}(\mathcal{F}), \\
P_p^{s,\infty}(\mathcal{F}) &= H_p(K_s)/\bigcup_{s\leq t} M_p^{s,t}(\mathcal{F}).
\end{aligned}$$

(Note that that for every fixed $s \in S$, $M_p^{s,t}(\mathcal{F})$ is a subspace of $H_p(K_s)$ and $M_p^{s,t}(\mathcal{F}) \subset M_p^{s,t'}(\mathcal{F})$ for $t \leq t'$.)

We will call

1. *$L_p^s(\mathcal{F})$ the* **space of p-dimensional cycles born at time** *s;*
2. *$P_p^{s,t}(\mathcal{F})$ the* **the space of p-dimensional cycles born at time** *s* **and which died at time** *t; and*
3. *$P_p^{s,\infty}(\mathcal{F})$ the* **the space of p-dimensional cycles born at time** *s* **and which never dies**.

We will denote for $s \in S, t \in S \cup \{\infty\}$,

$$\mu_p^{s,t}(\mathcal{F}) = \dim P_p^{s,t}(\mathcal{F}), \tag{3.1.2}$$

and call $\mu_p^{s,t}(\mathcal{F})$ *the* **persistent multiplicity of** p**-dimensional cycles born at time** s **and dying at time** t **if** $t \neq \infty$**, or never dying in the case** $t = \infty$*.*

Finally, we will call the set

$$\mathcal{B}_p(\mathcal{F}) = \{(s, t, \mu_p^{s,t}(\mathcal{F})) \mid \mu_p^{s,t}(\mathcal{F}) > 0\}$$

the p**-dimensional barcode associated to the filtration** $\mathcal{F}$*.*

It is well-known, see [17], that for $s < t$ and for all $p \geq 0$

$$\mu_p^{s,t} = (b_p^{s,t-1} - b_p^{s,t}) - (b_p^{s-1,t-1} - b_p^{s-1,t}).$$

Persistence diagrams encode all necessary information about persistent homology groups. Persistence has many practical applications because barcodes can be computed efficiently. [17] uses sparse matrix representations of a boundary matrix to calculate persistence diagrams in $\mathcal{O}(m^3)$ time, where m is the number of simplices of K. For a survey of major computational results in persistent homology, we refer the reader to [28].

3.2 Representative Cyles

3.2.1 Prior Work

Given a homology class, it is an important problem in applications to determine a chain of simplices to represent that class. In applications, the simplices of the complex have inherent meaning to the data being analyzed. This importance has led to several prior means to determine a representative cycle.

A natural area to explore is to minimize the number of simplices in the given cycle. Volume-optimal cycles were proposed by Obayashi in [29]. Volume-optimal cycles are cycles of a homology class with the fewest number of simplices, and they can be found as solutions to a linear programming optimization problem. Volume-optimal cycles focus on minimizing the size of a cycle; an alternative approach is to determine which simplices are necessary for the cycle and using them to represent the cycle. Such simplices are called "essential," and they were introduced by Basu et. al. ([30]). In addition, the authors present an algorithmic means to determine which simplices are essential. Similar to volume-optimal cycles, Dey et. al. ([31], [32]) define and present an algorithm to determine optimal one-cycles that have a minimal weight, in some sense of weight. They give a polynomial time algorithm to compute "meaningful" persistent one-cycles. These cycles are not stable under perturbations with respect to the Hausdorff distance or the measure of using cycle length. Finally, in [33], Gamble et. al. obtain a representative cycle by tracking when the addition or removal of a simplex causes a class to be born or die. Their method works with standard persistence as well as zigzag persistence.

3.2.2 Harmonic Chains

In this section, we describe a new method of determining a representative cycle of a homology class. We use the fact that the p-th homology group of a simplicial complex is isomorphic to the space of p-dimensional harmonic chains in that simplicial complex [2]. This isomorphism allows us to associate to each homology class a specific harmonic chain representing that class. We begin by defining harmonic chains.

Definition 19 (Harmonic Chains)**.** *Let $p \geq 0$. Given a simplicial complex K, let $C_p(K) = C_p(K, \mathbb{R})$ denote the $\mathbb{R}$-vector space of p-dimensional chains of K. Let $\partial_p : C_p(K) \to C_{p-1}(K)$ be the standard boundary map. We let $Z_p = \ker(\partial_p)$ and $B_p = \operatorname{im}(\partial_{p+1})$. An element of Z_p is called a* **cycle***, and an element of B_p is called a* **boundary***. We can make $C_p(K)$ into an Euclidean space by defining an inner product such that*

$$\langle \sigma, \sigma \rangle = \delta_{\sigma,\sigma'}, \ \ \sigma, \sigma' \in K^{(p)}$$

Under this inner product we denote $\mathfrak{h}_p(K) = Z_p(K) \cap B_p(K)^{\perp}$. An element of $\mathfrak{h}_p(K)$ is called a harmonic chain.

Remark 3. We note that in the literature it is common to use the coefficient field $\mathbb{Z}_2$. However, we cannot have a well-defined inner product over a finite field. To define an inner product on p-chains on K, it is necessary to have $\mathbb{F} = \mathbb{R}$.

Proposition 3.2.1. *The map* $\mathfrak{f}_p(K)$ *defined by* $z + B_p(K) \to \mathrm{proj}_{B_p(K)^\perp}(z)$ *gives an isomorphism*

$$\mathfrak{f}_p(K) : \mathrm{H}_p(K) \to \mathfrak{h}_p(K).$$

Proof. Observe using lemma 2 and the fact that $B_p(K) \subset Z_p(K)$, we have that for $z \in Z_p(K)$, $\mathrm{proj}_{B_p(K)^\perp}(z) \in Z_p(K)$, and so $\mathfrak{f}_p(K)$ is well-defined. Injectivity and surjectivity follow immediately. □

In terms of matrices, we have the following description of $\mathfrak{h}_p(K)$ as a subspace of $C_p(K)$.

We denote by $\mathcal{A}_p(K)$ the orthonormal basis $\{\sigma | \sigma \in K^{(p)}\}$ for each $p \geq 0$, and by $M_p(K)$ the matrix of ∂_p with respect to the basis $\mathcal{A}_p(K)$ of $C_p(K)$, and the basis $\mathcal{A}_{p-1}(K)$ of $C_{p-1}(K)$. Then, $\mathfrak{h}_p(K)$ can be identified as the subspace of $C_p(K)$ which is is equal to the intersection of the nullspaces of the two matrices $M_p(K)$ and ${}^tM_{p+1}(K)$. More precisely,

$$z \in \mathfrak{h}_p(K) \Leftrightarrow [z]_{\mathcal{A}_p(K)} \in \mathrm{null}(M_p(K)) \cap \mathrm{null}({}^tM_{p+1}(K)).$$

3.3 Harmonic Chains and Persistence

For this section, let K be a simplicial complex of dimension n. Let $\mathcal{F}$ be a simplex-wise filtration on K indexed by an ordered set S. We use the fact that homology groups are isomorphic to the set of harmonic chains to obtain the following commutative diagram for each dimension p:

$$\begin{tikzcd}
H_p(K_0) \arrow[r] \arrow[d, "\cong"] & H_p(K_1) \arrow[r] \arrow[d, "\cong"] & \cdots \arrow[r] \arrow[d, "\cong"] & H_p(K_m) \arrow[d, "\cong"] \\
\mathfrak{h}_p(K_0) \arrow[r] & \mathfrak{h}_p(K_1) \arrow[r] & \cdots \arrow[r] & \mathfrak{h}_p(K_m)
\end{tikzcd}$$

we define the above maps as follows:

$$\mathrm{i}_p^{s,t} : H_p(K_s) \to H_p(K_{s+1}) \qquad x + B_p(K_s) \mapsto x + B_p(K_{s+1}) \tag{3.3.1}$$

$$\mathfrak{f}_p(K_s) : H_p(K_s) \to \mathfrak{h}_p(K_s) \qquad x + B_p(K_s) \mapsto \mathrm{proj}_{B_p(K_s)^\perp}(x) \tag{3.3.2}$$

$$\mathfrak{i}_p^{s,t} : \mathfrak{h}_p(K_s) \to \mathfrak{h}_p(K_{s+1}) \qquad x \mapsto \mathrm{proj}_{B_p(K_{s+1})^\perp}(x) \tag{3.3.3}$$

Proposition 1. *The above diagram, with the given maps, is commutative.*

Proof. We begin by noting that $\mathfrak{i}_p^{s,t}$ is well-defined. For $x \in Z_p(K_s) \subset Z_p(K_{s+1})$ and $B_p(K_{s+1}) \subset Z_p(K_{s+1})$, it follows almost immediately from the definitions that $\mathrm{proj}_{B_p(K_{s+1})^\perp}(x) \in Z_p(K_{s+1})$.

We can now prove the diagram is commutative. Let $x + B_p(K_s) \in H_p(K_s)$. In one direction, we have that $x+B_p(K_s) \mapsto x+B_p(K_{s+1}) \mapsto \mathrm{proj}_{B_p(K_{s+1})^\perp}(x)$. In the other direction, we have $x+B_p(K_s) \mapsto \mathrm{proj}_{B_p(K_s)^\perp}(x) \mapsto \mathrm{proj}_{B_p(K_{s+1})^\perp}(\mathrm{proj}_{B_p(K_s)^\perp}(x))$. Equality follows from the fact that if $A \subset B$ are two subspaces of a vector space V, then $\mathrm{proj}_A(\mathrm{proj}_B(x)) = \mathrm{proj}_A(x)$ for all $x \in V$ and from the fact that $B_p(K_s) \subset B_p(K_{s+1})$ implies $B_p(K_{s+1})^\perp \subset B_p(K_s)^\perp$. □

Remark 4. This construction is analogous to a similar construction in de Rham cohomology. In this case, the "p-chains" are differential p-forms. The kernel of the appropriate "coboundary operator" on p-forms can be quotiented by the image to obtain de Rham cohomology. In this setting one can define harmonic forms, differential forms in the kernel of the Laplace operator on p-forms. As in our case on harmonic chains, these harmonic forms are isomorphic to the de Rham cohomology groups of the same dimension. For more information on the connection between de Rham cohomology and harmonic forms see [34].

Definition 20 (Harmonic Persistent Homology Subspace)**.** *For each $p \geq 0$ and $s \leq t \in S$, let $\mathfrak{h}_p^{s,t}(\mathcal{F}) = \mathrm{im}(\mathfrak{i}_p^{s,t})$. We call $\mathfrak{h}_p^{s,t}(\mathcal{F})$ a* **harmonic persistent homology subspace***.*

We now define the harmonic analogue of definition 3.1.1. The following are all subspaces of $\mathfrak{h}_p(K)$.

Definition 3.3.1. *For $s \leq t$, and $p \geq 0$, we define*

$$\begin{aligned}
\mathfrak{L}_p^s(\mathcal{F}) &= \mathfrak{h}_p(K_s) \cap \left(\bigcup_{s'<s} \mathfrak{h}_p^{s',s}(\mathcal{F}) \right)^\perp, \\
\mathfrak{M}_p^{s,t}(\mathcal{F}) &= \bigcup_{s'<s} (\mathfrak{i}_p^{s,t})^{-1}(\mathfrak{h}_p^{s',t}(\mathcal{F})), \\
\mathfrak{N}_p^{s,t}(\mathcal{F}) &= \bigcup_{s'<s\leq t'<t} (\mathfrak{i}_p^{s,t'})^{-1}(\mathfrak{h}_p^{s',t'}(\mathcal{F})), \\
\mathfrak{P}_p^{s,t}(\mathcal{F}) &= \mathfrak{M}_p^{s,t}(\mathcal{F}) \cap \mathfrak{N}_p^{s,t}(\mathcal{F})^\perp, \\
\mathfrak{P}_p^{s,\infty}(\mathcal{F}) &= \mathfrak{h}_p(K_s) \cap \bigcap_{s\leq t} \mathfrak{M}_p^{s,t}(\mathcal{F})^\perp.
\end{aligned}$$

Finally, we will call the set

$$\mathfrak{B}_p(\mathcal{F}) = \{(s, t, \mathfrak{P}_p^{s,t}(\mathcal{F})) \mid \mathfrak{P}_p^{s,t}(\mathcal{F})) \neq 0\}$$

the p-dimensional harmonic barcode associated to the filtration *$\mathcal{F}$.*

3.3.1 Practical applications

One of our motivations behind our definition of harmonic barcodes comes from applications. In applications of topological data analysis the simplices of the simplicial complexes are significant and it is useful to be able to label the bars of a persistent diagram by a representative cycle. There is no obvious choice for such a cycle and several different suggestions exist in the literature.

The harmonic bar codes give a natural solution to the above problem. Suppose for a given filtration $\mathcal{F}$, and for all $s \in T, t \in T \cup \{\infty\}, s \leq t, p \geq 0$, $0 \leq \mu_p^{s,t}(\mathcal{F}) \leq 1$. This will be true if the filtration is induced by a 'generic' real valued function on the simplices of K.

Suppose, $\mu_p^{s,t}(\mathcal{F}) = 1$. Then, the coresponding subspace $\mathfrak{P}_p^{s,t}(\mathcal{F})$ is one dimensional. Let $z_p^{s,t} = \sum_{\sigma \in K^{(p)}} a_\sigma \sigma \in C_p(K)$ be an element such that $\{z_p^{s,t}\}$ is an orthonormal basis of $\mathfrak{P}_p^{s,t}(\mathcal{F})$ (i.e. $||z_p^{s,t}|| = 1, z_p^{s,t} \in \mathfrak{P}_p^{s,t}(\mathcal{F})$).

We will denote by $w_p^{s,t}(\mathcal{F}) = (|a_\sigma|)_{\sigma \in K^{(p)}}$ the weight vector corresponding to the bar with end points s, t.

We continue by discussing an example below.

Example 1. Let $K = \{(a), (b), (c), (d), (a, b), (a, c), (a, d)(b, c), (c, d)\}$. Let $\mathcal{F}$ be the following simplex-wise filtration on K:

$$K_0 = \emptyset \subset K_1 = (a) \subset K_2 = K_1 \cup (b) \subset K_3 = K_2 \cup (c) \subset K_4 = K_3 \cup (d)$$

$$\subset K_5 = K_4 \cup (a, b) \subset K_6 = K_5 \cup (a, c) \subset K_7 = K_6 \cup (a, d)$$

$$\subset K_8 = K_7 \cup (b, c) \subset K_9 = K_8 \cup (c, d) = K$$

We consider 1-dimensional chains in this problem. The spaces of harmonic chains are:

$$\mathfrak{h}_1(K_s) = 0 \text{ for } s \leq 7$$
$$\mathfrak{h}_1(K_8) = \operatorname{span}\{(a,b) + (b,c) - (a,c)\} = \operatorname{span}\{z_1\}$$
$$\mathfrak{h}_1(K_9) = \operatorname{span}\{(a,b) + (b,c) - (a,c), -(a,d) + (c,d) + (a,c)\} = \operatorname{span}\{z_1, z_2\}$$

where z_1 is the chain $(a,b) + (b,c) - (a,c)$ and z_2 is the chain $-(a,d) + (c,d) + (a,c)$. The harmonic persistent homology subspaces are

$$\begin{aligned} \mathfrak{h}_1^{s,t}(\mathcal{F}) &= \operatorname{span}\{z_1\} & s = 8, t = 9 \\ \mathfrak{h}_1^{s,t}(\mathcal{F}) &= 0 & \text{otherwise} \end{aligned}$$

We first determine the space of 1-dimensional harmonic chains that are born at time s. It is clear that $\mathfrak{L}_1^s(\mathcal{F}) = 0$ for $s \leq 7$ and:

$$\begin{aligned} \mathfrak{L}_1^8(\mathcal{F}) &= \mathfrak{h}_1(K_8) \cap \left(\bigcup_{s'<8} \mathfrak{h}_1^{s',8}(\mathcal{F}) \right)^{\perp} \\ &= \operatorname{span}\{z_1\} \cap (0)^{\perp} \\ &= \operatorname{span}\{z_1\} \\ \mathfrak{L}_1^9(\mathcal{F}) &= \mathfrak{h}_1(K_9) \cap \left(\bigcup_{s'<9} \mathfrak{h}_1^{s',9}(\mathcal{F}) \right)^{\perp} \\ &= \operatorname{span}\{z_1, z_2\} \cap \left(\mathfrak{h}_1^{8,9}(\mathcal{F})\right)^{\perp} \\ &= \operatorname{span}\{z_1, z_2\} \cap (\operatorname{span}\{z_1\})^{\perp} \\ &= \operatorname{span}\{z_1\}^{\perp} \\ &= \operatorname{span}\{(a,b) + 2(a,c) - 3(a,d) + (b,c) + 3(c,d)\} \end{aligned}$$

We proceed to calculate $\mathfrak{M}_1^{s,t}$. It is clear that $\mathfrak{M}_1^{s,t}(\mathcal{F}) = 0$ if $s \leq 8$ because $\mathfrak{h}_1^{s',t}(\mathcal{F}) = 0$ for $s' < s \leq 8$. The only nonzero space is

$$\begin{aligned}
\mathfrak{M}_1^{9,9}(\mathcal{F}) &= \bigcup_{s'<9} (\mathfrak{i}_1^{9,9})^{-1}(\mathfrak{h}_1^{s',9}(\mathcal{F})) \\
&= \bigcup_{s'<9} \mathfrak{h}_1^{s',9}(\mathcal{F})) \\
&= \mathfrak{h}_1^{8,9}(\mathcal{F}) \\
&= \operatorname{span}\{z_1\}
\end{aligned}$$

The subspaces $\mathfrak{N}_1^{s,t}(\mathcal{F}) = 0$ for all s and t because the only nonzero harmonic persistent homology subspace is $\mathfrak{h}_1^{8,9}(\mathcal{F})$, but $s' < s < t \leq 9$ implies $s' < 8$ and so $\mathfrak{h}_1^{s',t}(\mathcal{F}) = 0$.

All that remains is to calculate $\mathfrak{P}_1^{s,t}(\mathcal{F})$. It is clear that if t is finite, $\mathfrak{P}_1^{s,t}(\mathcal{F}) = 0$, because $\mathfrak{M}_1^{s,t}(\mathcal{F}) = 0$ for $s \neq t$. $\mathfrak{P}_1^{s,\infty}(\mathcal{F}) = 0$ for $s < 8$ because $\mathfrak{h}_1(K_s) = 0$ for $s < 8$. We determine the subspace for $s = 8$ and $s = 9$:

$$\begin{aligned}
\mathfrak{P}_1^{8,\infty} &= \mathfrak{h}_1(K_8) \cap \bigcap_{8 \leq t} \mathfrak{M}_1^{8,t}(\mathcal{F})^{\perp} \\
&= \operatorname{span}\{z_1\} \cap (0)^{\perp} \\
&= \operatorname{span}\{z_1\} \\
\mathfrak{P}_1^{9,\infty} &= \mathfrak{h}_1(K_9) \cap \mathfrak{M}_1^{9,9}(\mathcal{F})^{\perp} \\
&= \operatorname{span}\{z_1, z_2\} \cap (\operatorname{span}\{z_1\})^{\perp} \\
&= \operatorname{span}\{(a,b) + 2(a,c) - 3(a,d) + (b,c) + 3(c,d)\}
\end{aligned}$$

In the barcode, we use the unit vector to represent the subspaces $\mathfrak{P}_1^{s,\infty}$:

$$\mathfrak{B}_1(\mathcal{F}) = \left\{ \left(8, \infty, \operatorname{span}\left\{ \frac{1}{\sqrt{3}}(a,b) + \frac{1}{\sqrt{3}}(b,c) + \frac{1}{\sqrt{3}}(a,c) \right\} \right), \right.$$

$$\left. \left(9, \infty, \operatorname{span}\left\{ \frac{1}{\sqrt{24}}(a,b) + \frac{2}{\sqrt{24}}(a,c) - \frac{3}{\sqrt{24}}(a,d) + \frac{1}{\sqrt{24}}(b,c) + \frac{3}{\sqrt{24}}(c,d) \right\} \right) \right\}.$$

Example 2 (Complete Graph on 5 vertices)**.** Let K be the complete graph on 5 vertices, labeled a, b, c, d, e. Let $\mathcal{F}$ be the following simplex-wise filtration on K:

$$K_0 = \emptyset \subset \cdots \text{Here the 0-simplices can be added in any order} \cdots$$

$$\subset K_6 = \{(a), (b), (c), (d), (\mathrm{e}), (a, b)\} \subset K_7 = K_6 \cup (b, c)$$

$$\subset K_8 = K_7 \cup (c, d) \subset K_9 = K_8 \cup (d, \mathrm{e}) \subset K_{10} = K_9 \cup (a, \mathrm{e}) \subset K_{11} = K_{10} \cup (a, c)$$

$$\subset K_{12} = K_{11} \cup (a, d) \subset K_{13} = K_{12} \cup (b, d) \subset K_{14} = K_{13} \cup (b, \mathrm{e}) \subset K_{15} = K_{14} \cup (c, \mathrm{e}) = K$$

Notice that $H_1(K_{10}) \cong \mathbb{R}, H_1(K_{11}) \cong \mathbb{R}^2, \ldots, H_1(K_{15}) \cong \mathbb{R}^6$. Hence $\dim(\mathfrak{h}_1(K_\mathrm{i})) = \mathrm{i} - 9$ for $\mathrm{i} = 9, \ldots, 15$. This gives the following bases for each harmonic chain group:

$$\begin{aligned}
\mathfrak{h}_1(K_{10}) &= \operatorname{span}\{(a, b) + (b, c) + (c, d) + (d, \mathrm{e}) - (a, \mathrm{e})\} = \operatorname{span}\{z_1\} \\
\mathfrak{h}_1(K_{11}) &= \operatorname{span}\{z_1, (a, b) + (b, c) - (a, c)\} = \operatorname{span}\{z_1, z_2\} \\
\mathfrak{h}_1(K_{12}) &= \operatorname{span}\{z_1, z_2, (a, c) + (c, d) - (a, d)\} = \operatorname{span}\{z_1, z_2, z_3\} \\
\mathfrak{h}_1(K_{13}) &= \operatorname{span}\{z_1, z_2, z_3, (a, b) + (b, d) - (a, d)\} = \operatorname{span}\{z_1, z_2, z_3, z_4\} \\
\mathfrak{h}_1(K_{14}) &= \operatorname{span}\{z_1, z_2, z_3, z_4, (b, d) + (d, \mathrm{e}) - (b, \mathrm{e})\} = \operatorname{span}\{z_1, z_2, z_3, z_4, z_5\} \\
\mathfrak{h}_1(K_{15}) &= \operatorname{span}\{z_1, z_2, z_3, z_4, z_5, (b, c) + (c, \mathrm{e}) - (b, \mathrm{e})\} = \operatorname{span}\{z_1, z_2, z_3, z_4, z_5, z_6\}
\end{aligned}$$

Notice that $\mathfrak{h}_1^{s,t}(\mathcal{F}) = 0$ whenever $s \leq 9$ because $\mathfrak{h}_1(K_s) = 0$ for $s \leq 9$. The nonzero harmonic persistent homology subspaces are:

$s \setminus t$	10	11	12	13	14
10	span$\{z_1\}$	span$\{z_1\}$	span$\{z_1\}$	span$\{z_1\}$	span$\{z_1\}$
11	-	span$\{z_1, z_2\}$	span$\{z_1, z_2\}$	span$\{z_1, z_2\}$	span$\{z_1, z_2\}$
12	-	-	span$\{z_1, z_2, z_3\}$	$spn\{z_1, z_2, z_3\}$	span$\{z_1, z_2, z_3\}$
13	-	-	-	span$\{z_1, z_2, z_3, z_4\}$	span$\{z_1, z_2, z_3, z_4\}$
14	-	-	-	-	span$\{z_1, z_2, z_3, z_4, z_5\}$
15	-	-	-	-	-

s	$t = 15$
10	span$\{z_1\}$
11	span$\{z_1, z_2\}$
12	span$\{z_1, z_2, z_3\}$
13	span$\{z_1, z_2, z_3, z_4\}$
14	span$\{z_1, z_2, z_3, z_4, z_5\}$
15	span$\{z_1, z_2, z_3, z_4, z_5, z_6\}$

The next step is to determine the values of $\mathfrak{L}_1^s$. We again note that this space is 0 for $s \leq 9$. Hence we have:

$$\mathfrak{L}_1^{10}(\mathcal{F}) = \text{span}\{z_1\}$$

$$\mathfrak{L}_1^{11}(\mathcal{F}) = \text{span}\{z_1\}^{\perp} = \text{span}\left\{3(a,b) + 3(b,c) - 2(c,d) - 2(d,\text{e}) + 2(a,\text{e}) - 5(a,c)\right\}$$

$$\mathfrak{L}_1^{12}(\mathcal{F}) = \text{span}\{z_1, z_2\}^{\perp} = \text{span}\{2(a,b) + 2(b,c) + 6(c,d) - 5(d,\text{e}) + 5(a,\text{e}) + 4(a,c) + 11(a,d)\}$$

$$\begin{aligned}\mathfrak{L}_1^{13}(\mathcal{F}) &= \text{span}\{z_1, z_2, z_3\}^{\perp} \\ &= \text{span}\{11(a,b) - 10(b,c) - 9(c,d) + 4(d,\text{e}) - 4(a,\text{e}) + (a,c) - 8(a,d) + 21(b,d)\}\end{aligned}$$

$$\begin{aligned}\mathfrak{L}_1^{14}(\mathcal{F}) &= \text{span}\{z_1, z_2, z_3, z_4\}^{\perp} \\ &= \text{span}\{-3(a,b) + 2(b,c) + (c,d) + 4(d,\text{e}) + 4(a,\text{e}) - (a,c) + 3(b,d) - 8(b,\text{e})\}\end{aligned}$$

$$\mathfrak{L}_1^{15}(\mathcal{F}) = \text{span}\{z_1, z_2, z_3, z_4, z_5\}^{\perp} = \text{span}\{(b,c) - (c,d) - (d,\text{e}) - (a,\text{e}) + (a,c) - (b,\text{e}) + 3(c,\text{e})\}$$

Because every cycle that is born at a given time s never dies, we have $\mathfrak{L}_1^s(\mathcal{F}) = \mathfrak{P}_1^{s,\infty}(\mathcal{F})$ for each s. Hence the barcode for this filtration, after normalizing the above vectors and sorting the simplices in "alphabetical" order, is

$$
\begin{aligned}
\mathfrak{B}_1(\mathcal{F}) = \{&(10, \infty, \operatorname{span}\{\frac{1}{\sqrt{5}} z_1\}, \\
&(11, \infty, \operatorname{span}\{\frac{3}{\sqrt{55}}(a,b) - \frac{5}{\sqrt{55}}(a,c) + \frac{2}{\sqrt{55}}(a,\mathrm{e}) + \frac{3}{\sqrt{55}}(b,c) - \frac{2}{\sqrt{55}}(c,d) - \frac{2}{\sqrt{55}}(d,\mathrm{e})\}), \\
&(12, \infty, \operatorname{span}\{\frac{2}{\sqrt{231}}(a,b) + \frac{4}{\sqrt{231}}(a,c) + \frac{11}{\sqrt{231}}(a,d) + \frac{5}{\sqrt{231}}(a,\mathrm{e}) \\
&+ \frac{2}{\sqrt{231}}(b,c) + \frac{6}{\sqrt{231}}(c,d) - \frac{5}{\sqrt{231}}(d,\mathrm{e})\}), \\
&(13, \infty, \operatorname{span}\{\frac{11}{\sqrt{840}}(a,b) + \frac{1}{\sqrt{840}}(a,c) - \frac{8}{\sqrt{840}}(a,d) - \frac{4}{\sqrt{840}}(a,\mathrm{e}) \\
&- \frac{10}{\sqrt{840}}(b,c) + \frac{21}{\sqrt{840}}(b,d) - \frac{9}{\sqrt{840}}(c,d) + \frac{4}{\sqrt{840}}(d,\mathrm{e})\}), \\
&(14, \infty, \operatorname{span}\{-\frac{3}{\sqrt{120}}(a,b) - \frac{1}{\sqrt{120}}(a,c) + \frac{4}{\sqrt{120}}(a,\mathrm{e}) + \frac{2}{\sqrt{120}}(b,c) \\
&+ \frac{3}{\sqrt{120}}(b,d) - \frac{8}{\sqrt{120}}(b,\mathrm{e}) + \frac{1}{\sqrt{120}}(c,d) + \frac{4}{\sqrt{120}}(d,\mathrm{e})\}), \\
&(15, \infty, \operatorname{span}\{\frac{1}{\sqrt{15}}(a,c) - \frac{1}{\sqrt{15}}(a,\mathrm{e}) + \frac{1}{\sqrt{15}}(b,c) - \frac{1}{\sqrt{15}}(b,\mathrm{e}) - \frac{1}{\sqrt{15}}(c,d) \\
&+ \frac{3}{\sqrt{15}}(c,\mathrm{e}) - \frac{1}{\sqrt{15}}(d,\mathrm{e})\}\}
\end{aligned}
$$

We won't list the weight of every element of $\mathfrak{B}_1(\mathcal{F})$, but as one example,

$$
w_p(11, \infty, \mathfrak{P}_1^{s,\infty}(\mathcal{F})) = \left(\frac{3}{\sqrt{55}}, \frac{5}{\sqrt{55}}, 0, \frac{2}{\sqrt{55}}, \frac{3}{\sqrt{55}}, 0, 0, \frac{2}{\sqrt{55}}, 0, \frac{2}{\sqrt{55}}\right).
$$

3.3.2 Towards Proving Stability

A desirable quality that we would like to have is that the barcodes/persistent diagrams of these persistent harmonic chain groups are stable under small perturbations. A traditional distance used to measure stability between persistence diagrams is the bottleneck distance.

Given a filtration $\mathcal{F}$ on a simplicial complex K, for each $p \geq 0$ we shall denote by $Dgm_p(\mathcal{F})$ the multiset of points in $\mathbb{R}^2 \cup \{\infty\}$ containing the points (s,t) with multiplicity

$\mu_p^{s,t}(\mathcal{F})$ for every $(s, t, \mu_p^{s,t}) \in \mathcal{B}_p(\mathcal{F})$ union the set of points $\{(s, s)\}_{s\in\mathbb{R}}$ with infinite multiplicity.

Definition 21 (Bottleneck Distance)**.** *Let $\mathcal{F}$ and $\mathcal{F}'$ be two filtrations on a simplicial complex K. Let $X = Dgm_p(\mathcal{F})$ and $Y = Dgm_p(\mathcal{F}')$. The* **bottleneck distance***, denoted W_∞, between X and Y is:*

$$W_\infty(X, Y) = \inf_{\eta : X \to Y} \sup_{x \in X} |x - \eta(x)|_\infty$$

where η is taken over all possible bijections between X and Y and $|\cdot|_\infty$ represents the L_∞-norm.

Edelsbruner and Harer prove that persistence diagrams are stable with respect to the bottleneck distance in the sense described in the theorem below.

Theorem 3.3.1 (Bottleneck Stability)**.** *[17] Let K be a simplicial complex and $f, g : K \to \mathbb{R}$ be two functions that induce filtrations on K. For each dimension p, we have*

$$W_\infty(Dgm_p(\mathcal{F}_f), Dgm_p(\mathcal{F}_g)) \leq |f - g|_\infty.$$

We prove a stability result on harmonic chain groups using persistence functions (definition 23). To define these functions we need the Grassmannian of a vector space.

Definition 22 (Grassmannian)**.** *Let V be a p-dimensional vector space over a field $\mathbb{F}$. The* **k-dimensional Grassmannian of** *V is the space of all k-dimensional subspaces of V. The doubly infinite Grassmannian, $Gr(\infty, \infty)$, is the disjoint union of all d dimensional subspaces of a p-dimensional vector space over $\mathbb{F}$, for every $d \leq p \in \mathbb{N}$.*

For our purposes, we take $V = C_p(K)$ and $\mathbb{F} = \mathbb{R}$. we can now define persistence functions.

Definition 23 (Persistence Function)**.** *Let $f : K \to \mathbb{R}$ be a function that induces a simplex-wise filtration $\mathcal{F}_f$ on K. For each dimension p and each time t, we obtain a harmonic chain group $\mathfrak{h}_p(K_t)$, where $K_t = f^{-1}(-\infty, t]$ (note that K_t will be equal to a subcomplex of K in* *the filtration $\mathcal{F}_f$ for each t). This allows us to define the following new functions induced by f on $\mathbb{R}$ for each p:*

$$F_p : \mathbb{R} \to \amalg_d Gr(d, C_p(K))$$
$$s \stackrel{F_p}{\mapsto} \mathfrak{h}_p(K_s)$$

We call F_p a **persistence function of dimension** p.

Remark 5. Let F_p be a persistence function of dimension p. Then F_p has the following properties:

1. F_p is piecewise constant

2. F_p has finitely many discontinuities at times $\{t_1, \ldots, t_N\}$

3. $|\dim(F_p(t_{\mathrm{i}})) - \dim(F_p(t_{\mathrm{i}+1}))| \leq 1$ for $\mathrm{i} = 1, \ldots, N-1$

4. $F_p(t) = 0$ for all $t < t_1$

Any function $F : \mathbb{R} \to Gr(\infty, \infty)$ with these properties we will also call a persistence function.

Remark 6. Notice that F_p contains all the information of persistent homology. Once F_p is known, for $s \leq t$ we have $\mathrm{proj}_{F_p(t)}(F_p(s)) = \mathfrak{h}_p^{s,t}(K) \cong H_p^{s,t}(K)$. With this in mind we can translate definition 3.3.1 as:

$$\begin{aligned}
\mathfrak{L}_p^s(F_p) &= F_p(s) \cap \bigcup_{s'<s} \left(\mathrm{proj}_{F_p(s)}(F_p(s'))\right)^{\perp} \\
\mathfrak{M}_p^{s,t}(F_p) &= \bigcup_{s'<s} (\mathfrak{i}_p^{s,t})^{-1}(\mathrm{proj}_{F_p(t)}(F_p(s')) \\
\mathfrak{N}_p^{s,t}(F_p) &= \bigcup_{s'<s\leq t'<t} (\mathfrak{i}_p^{s,t'})^{-1}(\mathrm{proj}_{F_p(t)}(F_p(s')) \\
\mathfrak{P}^{s,t}(F_p) &= \mathfrak{M}_p^{s,t}(F_p) \cap (\mathfrak{N}_p^{s,t}(F_p)^{\perp} \\
\mathfrak{P}^{s,\infty}(F_p) &= F_p(s) \cap \bigcap_{s\leq t} \mathfrak{M}_p^{s,t}(F_p)^{\perp}
\end{aligned}$$

To determine the stability of the harmonic spaces, we need a notion of distance on the Grassmannian. From Lim and Ye [35], we having the following metric on $Gr(\infty,\infty)$ for $\mathbf{A} \in Gr(k,n), \mathbf{B} \in Gr(l,n)$ with $k,l \leq n$:

$$d_{Gr(\infty,\infty)}(\mathbf{A},\mathbf{B}) = \left(|k-l|\frac{\pi^2}{4} + \sum_{\mathrm{i}=1}^{\min\{k,l\}} \theta_{\mathrm{i}}^2\right)^{1/2}$$

where θ_{i} is the i-th principal angle between $\mathbf{A}$ and $\mathbf{B}$. To determine θ_{i}, let A and B be matrices of the appropriate dimension whose columns are orthonormal bases of $\mathbf{A}$ and $\mathbf{B}$. We may use singular value decomposition to write $A^T B = U\Sigma V^T$, where $\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{k\times l}$. Here $\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_{\min\{k,l\}})$, and we have that $\theta_{\mathrm{i}} = \cos^{-1}(\sigma_{\mathrm{i}})$ for $\mathrm{i} = 1, \ldots, \min\{k,l\}$. Note that because singular value decomposition is obtainable algorithmically, determining the distance between subspaces is straightforward to compute.

From remark 6 since F_p contains all the information on persistence harmonic subspaces, we can determine the distance between the set of harmonic subspaces by defining a distance between two persistence functions of dimension p F_p and G_p induced by functions f and g, respectively. We define this distance as

$$d_{\mathfrak{h}}(F_p, G_p) = \int_{\mathbb{R}} d_{Gr(\infty,\infty)}(F_p(t), G_p(t))dt. \tag{3.3.4}$$

The fact that $d_{\mathfrak{h}}$ is a metric follows from the fact that $d_{Gr(\infty,\infty)}$ is a metric. There are several metrics on $Gr(\infty,\infty)$ listed in [35]. The following results will hold for each metric, except the Martin metric, just with a different constant c in the case of Theorems 15 and 17. The Martin metric fails because subspaces of different dimensions are defined to have infinite distance between each other, and our arguments don't hold in that case.

Theorem 15. *Let K be a finite simplicial complex. For each $p \geq 0$, there exists a c depending only on K such that such that if F, G are persistence functions induced f, g, respectively, then $d_{\mathfrak{h}}(F,G) \leq c \cdot |f - g|$, where $|\cdot|$ is the L_∞ norm.*

Proof. Let m be the number of p dimensional simplices of K. There are only finitely many disjoint intervals $\{[x_{\mathrm{i}}, y_{\mathrm{i}})\}_{\mathrm{i}=1}^{M}$, for some $M \leq m$, where $F(t)$ and $G(t)$ are constant and different.This is because F and G differ at time t if and only if $\mathfrak{h}_p(K_t^f) \neq \mathfrak{h}_p(K_t^g)$. This happens when the p dimensional simplices are added in different order and/or at different times by f and g. There are only finitely many p dimensional simplices to add, and all such simplices are eventually added, so F and G can only differ on a set of finitely many disjoint intervals. Secondly, we can obtain an upper bound on $\max_{\mathrm{i}}\{y_{\mathrm{i}} - x_{\mathrm{i}}\}$ as follows. Because $|f - g| < \varepsilon$, $|f(\sigma) - g(\sigma)| < \varepsilon$ for all p-dimensional simplices $\sigma \in K$. The images of F and G are determined by the values f and g take on each simplex. Hence in total, F and G can only ever differ over an interval of length at most $m \cdot \varepsilon$. Therefore we have that

$$d_{\mathfrak{h}}(F,G) = \sum_{\mathrm{i}=1}^{M} \int_{x_{\mathrm{j}}}^{y_{\mathrm{j}}} d_{Gr(\infty,\infty)}(F(t),G(t))dt = \sum_{\mathrm{i}=1}^{M} (y_{\mathrm{i}} - x_{\mathrm{i}}) d_{Gr(\infty,\infty)}(F(x_{\mathrm{i}}),G(x_{\mathrm{i}}))$$

because F and G are piecewise constant. From here, using the definition of $d_{Gr(\infty,\infty)}$, we see that

$$d_{\mathfrak{h}}(F,G) \leq \sum_{\mathrm{i}=1}^{M} (m \cdot \varepsilon) \left(|\dim(F(x_{\mathrm{i}})) - \dim(G(x_{\mathrm{i}}))| \frac{\pi^2}{4} + \sum_{\mathrm{j}=1}^{\min\{\dim(F(x_{\mathrm{i}})),\dim(G(x_{\mathrm{i}}))\}} \theta_{\mathrm{j}}^2 \right)^{\frac{1}{2}}$$

Because θ_{j} is a principal angle, $0 \leq \theta_{\mathrm{j}} \leq \frac{\pi}{2}$ for all j. We assume there are m p-dimensional simplices, so $|\dim(F(x_{\mathrm{i}})) - \dim(G(x_{\mathrm{i}}))| \leq m$. Therefore:

$$d_{\mathfrak{h}}(F,G) \leq \sum_{\mathrm{i}=1}^{M} (m \cdot \varepsilon) \left(\frac{m\pi^2}{4} + \sum_{\mathrm{j}=1}^{m} \left(\frac{\pi}{2}\right)^2 \right)^{\frac{1}{2}}$$

$$= \sum_{\mathrm{i}=1}^{M} (m \cdot \varepsilon) \left(\frac{m\pi^2}{4} + \frac{m\pi^2}{4} \right)^{\frac{1}{2}} \leq \left(\frac{m^2\sqrt{m}\pi}{\sqrt{2}} \right) \cdot \varepsilon$$

Setting $c = \frac{m^2\sqrt{m}\pi}{\sqrt{2}}$, we have proven the claim, since c depends only on the number of p dimensional simplices of K. □

The above theorem proves that the harmonic distance between two persistence functions is stable with respect to the L_∞-norm of the functions that induce the persistence functions. The next theorem shows that persistence function of f contains all the information of $\mathfrak{B}_p(\mathcal{F}_f)$.

Theorem 16. *Each persistence function of dimension p can be associated with a distinct p-dimensional barcode.*

Proof. Let F be a persistence function of dimension p with discontinuities $\{t_1, \ldots, t_N\}$. Each point t_i where $\dim(F(t_\mathrm{i})) > \dim(F(t_{\mathrm{i}-1})$ corresponds to the birth of a new harmonic chain at time t_i. Each point t_j where $\dim(F(t_\mathrm{j})) < \dim(F(t_{\mathrm{j}-1}))$ corresponds to the death of a harmonic chain at time t_j. Pair each death with the most recent birth (i.e. if there is a death at t_5 and births at t_2 and t_4, pair t_5 with t_4). Note that conditions (3) and (4) in remark 5 guarantee that for each death there is always a prior birth to pair with. If there are unpaired births, pair them with $+\infty$. For each pairing $(t_\mathrm{i}, t_\mathrm{j})$, where t_j may be infinity, create a tuple $(t_\mathrm{i}, t_\mathrm{j}, \mathfrak{P}_p^{t_\mathrm{i}, t_\mathrm{j}}(F))$, where $\mathfrak{P}_p^{t_\mathrm{i}, t_\mathrm{j}}(F)$ is as defined in remark 6. It is clear that the set of all such tuples is a p-dimensional harmonic barcode. □

Remark 7. We denote by $\mathfrak{B}_p(F)$ the barcode obtained from a p-dimensional persistence function F. If F is induced by a function $f : K \to \mathbb{R}$, then $\mathfrak{B}_p(\mathcal{F}_f) = \mathfrak{B}_p(F)$. This is because the discontinuities of F are determined by the filtration $\mathcal{F}_f$ induced by f. Hence the birth and death times of the harmonic chains from $\mathcal{F}_f$ will be the same as the birth and death times from Theorem 16 and $\mathfrak{P}_p^{s,t}(F) = \mathfrak{P}_p^{s,t}(\mathcal{F}_f)$.

We connect persistence functions to the traditional bottleneck distance by proving that the harmonic distance between persistence functions is stable with respect to the traditional bottleneck distance between persistence diagrams. As before, to define the bottleneck distance, we add the diagonal to the barcodes with infinite multiplicity and denote this by $\mathrm{Dgm}_p(F)$.

Theorem 17. *Let K be a finite simplicial complex. For each $p \geq 0$, there exists a $c = c(K)$ depending only on K such that if F and G are persistence functions of dimension p then $d_\mathfrak{h}(F, G) \leq c \cdot W_\infty(\mathrm{Dgm}_p(F), \mathrm{Dgm}_p(G))$.*

Proof. Let m be the number of p simplices of K. Let $\{s_1, \ldots, s_n\}$ and $\{t_1, \ldots, t_n\}$ be the points of discontinuity of F and G, respectively. For every birth-death pair $(s_{\mathrm{i}_1}, s_{\mathrm{i}_2})$ in $Dgm_p(F)$, one of two things happen. Either there exists a birth-death pair $(t_{\mathrm{j}_1}, t_{\mathrm{j}_2})$ in $Dgm_p(G)$ such that $\max\{|s_{\mathrm{i}_1} - t_{\mathrm{j}_1}|, |s_{\mathrm{i}_2} - t_{\mathrm{j}_2}|\} < \varepsilon$, or $|s_{\mathrm{i}_2} - s_{\mathrm{i}_1}| < \varepsilon$. In either case, $F(t)$

differs from $G(t)$ on an interval of length at most ε. Since there are only m p simplices, the total length of intervals where F and G can differ is bounded above by $m \cdot \varepsilon$. Let $\{[x_{\mathrm{i}}, y_{\mathrm{i}})\}_{\mathrm{i}=1}^{M}$ be the intervals where F and G differ. We proceed similarly to the proof of theorem 15.

$$d_{\mathfrak{h}}(F,G) = \sum_{\mathrm{i}=1}^{M} \int_{x_{\mathrm{j}}}^{y_{\mathrm{j}}} d_{Gr(\infty,\infty)}(F(t), G(t))dt = \sum_{\mathrm{i}=0}^{M} (y_{\mathrm{i}} - x_{\mathrm{i}}) d_{Gr(\infty,\infty)}(F(x_{\mathrm{i}}), G(x_{\mathrm{i}}))$$

$$\leq \sum_{\mathrm{i}=1}^{M} (m \cdot \varepsilon) \left(|\dim(F(x_{\mathrm{i}})) - \dim(G(x_{\mathrm{i}}))| \frac{\pi^2}{4} + \sum_{\mathrm{j}=1}^{\min\{\dim(F(x_{\mathrm{i}})), \dim(G(x_{\mathrm{i}}))\}} \theta_{\mathrm{j}}^2 \right)^{\frac{1}{2}}$$

$$\leq \sum_{\mathrm{i}=1}^{M} (m \cdot \varepsilon) \left(\frac{m\pi^2}{2} \right)^{\frac{1}{2}} \leq \left(\frac{m^2\sqrt{m}\pi}{\sqrt{2}} \right) \cdot \varepsilon$$

Setting $c = \frac{m^2\sqrt{m}\pi}{\sqrt{2}}$, we have proven the claim, since c depends only on the number of p dimensional simplices of K. $\square$

3.4 Conclusion

In this chapter, we examined the problem of determining representative cycles of homology classes. We explored previous results in this area, before suggesting using harmonic chains as a means to solve the problem. Harmonic chains have a natural inner product that works well for determining a chain to represent the homology class. By codifying the information of persistent harmonic chain groups into persistence functions, we are able to prove a stability of these representations. This stability implies that harmonic chains are a natural choice as representatives of the homology classes. We have mainly focused on developing the theory of harmonic persistent homology subspaces as representative cycles. It is useful in applications to have representative cycles to represent persistent homology classes. Future work in this area should be applying the techniques described here to real world data sets.

REFERENCES

[1] L. van den Dries, *Tame Topology and O-minimal Structures*, ser. London Mathematical Society Lecture Note Series. Cambridge: Cambridge University Press, 1998, vol. 248, pp. x+180, ISBN: 0-521-59838-9.

[2] B. Eckmann, "Harmonische funktionen und randwertaufgaben in einem komplex", *Commentarii Mathematici Helvetici*, vol. 17, no. 1, pp. 240–255, 1944.

[3] E. Munch, "A user's guide to topological data analysis", *Journal of Learning Analytics*, vol. 4, no. 2, pp. 47–61, 2017.

[4] F. Kirwan, "Rational intersection cohomology of quotient varieties", *Inventiones mathematicae*, vol. 86, no. 3, pp. 471–505, Oct. 1986.

[5] D. Mumford, F. Kirwan, and J. Fogarty, *Geometric Invariant Theory*, 3rd. Berlin: Springer-Verlag, 1994.

[6] S. Basu, N. Cox, and S. Percival, *On the reeb spaces of definable maps*, 2020.

[7] G. Reeb, "Sur les points singuliers d'une forme de pfaff complètement intégrable ou d'une fonction numérique", *Comptes Rendus de l'Académie des Sciences*, vol. 222, pp. 847–849, 1946.

[8] G. Singh, F. Mémoli, and G. Carlsson, "Topological methods for the analysis of high dimensional data sets and 3d object recognition", *Eurographics Symposium of Point-Based Graphics*, 2007.

[9] E. Munch and B. Wang, "Convergence between categorical representations of Reeb space and Mapper", in *32nd International Symposium on Computational Geometry (SoCG 2016)*, S. Fekete and A. Lubiw, Eds., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 51, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 53:1–53:16, ISBN: 978-3-95977-009-5. DOI: 10.4230/LIPIcs.SoCG.2016.53. arXiv: 1512.04108.

[10] X. Ge, I. Safa, M. Belkin, and Y. Wang, "Data skeletonization via reeb graphs", in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., vol. 24, 2011.

[11] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology matching for fully automatic similarity estimation of 3d shapes", in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01, New York, NY, USA: Association for Computing Machinery, 2001, pp. 203–212.

[12] Z. Wood, H. Hoppe, M. Desbrun, and P. Schröder, "Removing excess topology from isosurfaces", *ACM Trans. Graph.*, vol. 23, no. 2, pp. 190–208, 2004.

[13] T. K. Dey, F. Mémoli, and Y. Wang, "Topological analysis of nerves, reeb spaces, mappers, and multiscale mappers", *Proceedings of International Symposium of Computational Geometry*, 2017.

[14] N. Cox, "A bound on the higher dimensional betti numbers of the reeb space", preprint.

[15] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, 2nd. Springer, 2006, vol. 10.

[16] N. Cox, "Effective semi-algebraic quotient algorithm", Effective Methods in Algebraic Geometry (MEGA), 2021.

[17] H. Edelsbrunner and J. L. Harer, *Computational topology.* American Mathematical Society, Providence, RI, 2010, pp. xii+241, An introduction, ISBN: 978-0-8218-4925-5.

[18] K. Emmett, B. Schweinhart, and R. Rabadan, "Multiscale topology of chromatin folding", in *Proceedings of the 9th EAI international conference on bio-inspired information and communications technologies*, ICST, 2016, pp. 177–180.

[19] P. Wu, C. Chen, Y. Wang, S. Zhang, C. Yuan, Z. Qian, D. Metaxas, and L. Axel, "Optimal topological cycles and their application in cardiac trabeculae restoration", in *International Conference on Information Processing in Medical Imaging*, Springer, Ed., 2017, pp. 80–92.

[20] S. Basu and N. Cox, "Harmonic persistent homology", preprint.

[21] M. Coste, *An introduction to o-minimal geometry.* Pisa: Istituti Editoriali e Poligrafici Internazionali, 2000, Dip. Mat. Univ. Pisa, Dottorato di Ricerca in Matematica.

[22] E. Mendelson, *Introduction to Mathematical Logic*, Sixth. CRC Press, 2015.

[23] S. Basu, *Algorithmic semi-algebraic geometry and topology - recent progress and open problems*, Sep. 2007.

[24] S. Basu, R. Pollack, and M.-F. Roy, "Computing the dimension of a semi-algebraic set", *Journal of Mathematical Sciences*, vol. 134, no. 5, pp. 2346–2353, 2006.

[25] A. Gabrielov, N. Vorobjov, and T. Zell, "Betti numbers of semialgebraic and sub-Pfaffian sets", *Journal of the London Mathematical Society*, vol. 69, no. 1, pp. 27–43, 2004.

[26] S. Basu, “On bounding the betti numbers and computing the euler characteristic of semi-algebraic sets”, *Discrete Computational Geometry*, vol. 22, no. 1, pp. 1–18, 1999.

[27] S. Basu and N. Karisani, *Efficient simplicial replacement of semi-algebraic sets and applications*, arXiv, 2020.

[28] N. Otter, M. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, “A roadmap for the computation of persistent homology”, *EPJ Data Science*, vol. 6, no. 17, 2017.

[29] I. Obayashi, “Volume-optimal cycle: Tight representative cycle of a generator in persistent homology”, *SIAM Journal on Applied Algebra and Geometry*, vol. 2, no. 4, pp. 508–534, 2018.

[30] S. Basu, F. Utro, and L. Parida, “Essential simplices in persistent homology and subtle admixture detection”, in *18th International Workshop on Algorithms in Bioinformatics*, L. Parida and E. Ukkonen, Eds., ser. Leibniz International Proceedings in Informatics, vol. 113, Schloss Dagstuhl, 2018, 14:1–14:10.

[31] T. K. Dey, T. Hou, and S. Mandal, “Persistent 1-cycles: Definition, computation, and its application”, *CoRR*, 2018.

[32] T. K. Dey, T. Hou, and S. Mandal, “Computing minimal persistent cycles: Polynomial and hard cases”, in *Proceedings ACM-SIAM Sympos. Discrete Algorithms (SODA20)*, 2019.

[33] J. Gamble, H. Chintakunta, and H. Krim, “Adaptive tracking of representative cycles in regular and zigzag persistent homology”, *CoRR*, 2014.

[34] R. Bott and L. W. Tu, *Differential Forms in Algebraic Topology*, ser. Graduate Texts in Mathematics. Springer, New York, 1982.

[35] L.-H. Lim and K. Ye, “Schubert varieties and distances between subspaces of different dimensions”, *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 3, pp. 1176–1197, 2016.

VITA

My academic journey began at Penn High School in Mishawaka, IN. After I graduated in 2010, I moved to St. Olaf College in Northfield, MN. At St. Olaf, I double majored in Math and Physics. It was my time at St. Olaf that kindled my interested in mathematics and inspired me to pursue a doctorate degree. My academic journey as a student ends at Purdue University, but the skills and knowledge I have learned here will continue to guide me throughout my life.