EFFECTIVE ALGORITHM FOR COMPUTING QUOTIENTS OF SEMI-ALGEBRAIC EQUIVALENCE RELATIONS

NATHANAEL COX

Abstract. 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.

Contents

1. Introduction

1.1. **Background.** 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, the important problem becomes figuring out 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 [Kirwan, 1986]. 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 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: [Mumford et al., 1994]).

An example of some interest is the case of semi-algebraic sets in R^n. Given a semi-algebraic set X and a semi-algebraic equivalence relation E, i.e. $E \subset X \times X$ is a semi-algebraic set, one would like to know under what circumstances the space X/E is realizable as a semi-algebraic set. Semi-algebraic sets are an example of a more general class of objects: o-minimal structures (see below for some basic information on such structures. For a more detailed introduction see [Coste, 2000] or [van den Dries, 1998].) The question on semi-algebraic sets leads to the same question on the more general setting of o-minimal structures. Given a definable set X and definable equivalence relation E on X, under what conditions is the resulting quotient space, X/E, realizable as a definable set. It was shown by van den Dries in the nineties in his text on o-minimal structures (see [van den Dries, 1998]) that X/E is not just realizable as a definable set,but also as a definably proper set, if and only if E is a definably proper equivalence relation on X. E is a definably proper equivalence relation if and only if the projection map $p : E \to X$ is a definably proper map, i.e. for every definable set $K \subset X$, if K is closed and bounded in the ambient space of X, then $p^{-1}(K) \subset E$ is closed and bounded in the ambient space of E.

A quotient space of particular interest is the following: Given a topological space X and a continuous function $f : X \to \mathrm{R}$, we define an equivalence relation $E_{\mathrm{Reeb}} \subset X \times X$ on X by setting $(x_1, x_2) \in E$ if and only if $f(x_1) = f(x_2)$ and x_1 and x_2 are in the same connected component of $f^{-1}(f(x_1)) = f^{-1}(f(x_2))$. We call X/E the Reeb graph of f and denote it $\mathrm{Reeb}(f)$. Reeb graphs were first introduced by Georges Reeb in [Reeb, 1946] as a tool in Morse Theory. They were further generalized to allow the case where $f : X \to Y$, for a general topological space Y. In this case, we refer to $\mathrm{Reeb}(f)$ as the Reeb space of f. In [Burlet and de Rham, 1974], Burlet and de Rham introduced the Reeb space of f as the Stein Factorization of a map f, where f was restricted to bivariate, generic, smooth maps.

Reeb spaces are of particular interest in semi-algebraic geometry. It was shown in [Basu et al., 2020] that if X is a closed and bounded definable set and $f : X \to Y$ is a definable map, then the equivalence relation E_{Reeb} is definably proper. Moreover, this means that $\mathrm{Reeb}(f)$ exists as a definably proper quotient of X. In other words $\mathrm{Reeb}(f)$ is definably homeomorphic to a definable set, so this equivalence relation preserves definability.

1.2. **Effective Algorithms and Complexity.** We prove our main result with an effective algorithm that outputs the space we are looking for. In [Basu et al., 2006a], Basu, Pollack, and Roy define an algorithm as "a computational procedure that takes an input and after a finite number of allowed operations produces and output." In our case, we are considering formulas defined by polynomial equalities and inequalities over R so our allowed operations are addition, multiplication, division by nonzero elements, and comparing elements with the natural ordering $<$ on R. The complexity of an algorithm is the number of operations performed by the algorithm in terms of the size of the input. The size of the input is a vector of integers. In our case, this vector contains the numbers of polynomials necessary to define our formulas, the number of variables in those polynomials, and their maximal degree bound.

In our algorithm, we use several well known algorithms in algorithmic algebraic geometry. We describe effective quantifier elimination and semi-algebraic triangulation in more detail below. However, the complexity of these algorithms are singly exponential and doubly exponential, respectively, in the number of 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. Cyclindrical Decomposition is 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.

1.3. **Main Result.** Our main result makes effective in the semi-algebraic category the following result, due to van den Dries:

Theorem 1 (Theorem 10.2.15 [van den Dries, 1998])**.** *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.*

Our main result is the following theorem on semi-algebraic sets (which are definable).

Theorem 2. *Let $\mathcal{P}_1 \subset \mathrm{R}[X_1, \ldots, X_m]$ and $\mathcal{P}_2 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$. 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 $P \in \mathcal{P}_2$. If $k_2 \approx k_1^{2^{\mathcal{O}(m)}}$ and $d_2 \approx d_1^{2^{\mathcal{O}(m))}}$, this algorithm has complexity

$$2^{2^{\mathcal{O}(m^2)} m^{\mathcal{O}(m)}} m^{m^{\mathcal{O}(m)}} (kd)^{2^{\mathcal{O}(m^3)} m^{\mathcal{O}(m^2)}}$$

where $k \approx k_2 \approx k_1^{2^{\mathcal{O}(m)}}$ and $d \approx d_2 \approx d_1^{2^{\mathcal{O}(m))}}$.

If $k_2 >> k_1^{2^{\mathcal{O}(m)}}$ and $d_2 >> d_1^{2^{\mathcal{O}(m))}}$, the algorithm has complexity:

$$2^{2^{\mathcal{O}(m^2)} m^{\mathcal{O}(m)}} m^{m^{\mathcal{O}(m)}} (k_2 d_2)^{2^{\mathcal{O}(m^3)} m^{\mathcal{O}(m^2)}}.$$

Finally, if $k_1^{2^{\mathcal{O}(m)}} >> k_2$ and $d_1^{2^{\mathcal{O}(m))}} >> d_2$, the algorithm has complexity:

$$2^{2^{\mathcal{O}(m^2)} m^{\mathcal{O}(m)}} m^{m^{\mathcal{O}(m)}} (k_1 d_1)^{2^{\mathcal{O}(m^4)} m^{\mathcal{O}(m^2)}}.$$

2. Preliminary Results

2.1. **Notation and Definitions.** Before we begin proving our main result, we need some background definitions. We begin with o-minimal structures, a set of "nice" subsets of real closed fields.

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

(i) $\mathcal{S}_m$ *is a boolean algebra of subsets of* R^m*.*
(ii) If $A \in \mathcal{S}_m$*, then* $A \times \mathrm{R}, \mathrm{R} \times A \in \mathcal{S}_{m+1}$*.*
(iii) The set $\{(x_1, \ldots, x_m) \in \mathrm{R}^m | x_1 = x_m\}$ *is in* $\mathcal{S}_m$*.*
(iv) If $A \in \mathcal{S}_{m+1}$ *and* $\pi : \mathrm{R}^{m+1} \to \mathrm{R}^m$ *is the projection onto the first* m *coordinates, then* $\pi(A) \in \mathcal{S}_m$*.*
(v) The set $\{(x, y) \in \mathrm{R}^2 | x < y\} \in \mathcal{S}_2$*.*
(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 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 formulas in the language of the reals. We go into more detail about such formulas 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 $S_1 \subset \mathrm{R}^{m_1}, \ldots, S_k \subset \mathrm{R}^{m_k}$ is a tuple $(h_1, \ldots, h_k, T)$ consisting of a definable set $T \subset \mathrm{R}^n$, for some n, and definable maps $h_i : S_i \to T$ such that :*

(i) h_i is a homeomorphism onto $h_i(S_i)$ and $h_i(S_i)$ is open in T, for $i = 1, \ldots, k$
(ii) T is the disjoint union of the sets $h_1(S_1), \ldots, h_k(S_k)$

Let $n = 1 + \max\{m_i | 1 \leq i \leq k\}$ and $h_i : S_i \to \mathrm{R}^n$ by $h_i(x) = (x, i, \ldots, i)$. Then $(h_1, \ldots, h_k, \cup_i h_i(S_i))$ 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 a concept of a disjoint sum in the definable category, we would like to construct quotients on these sums. We take definable sets $X \subset \mathrm{R}^m$ and $Y \subset \mathrm{R}^n$, 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 $S \subset \mathrm{R}^m$ is a pair (h, S') consisting of a closed and bounded definable set $S' \subset \mathrm{R}^n$ (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 $S \subset \mathrm{R}^m$ and $T \subset \mathrm{R}^n$, a* **completion of** $f : S \to T$ *is a triple consisting of a completion $i : S \to S'$ of S and $j : T \to T'$ and a definable continuous map $f' : S' \to T'$ such that $f' \circ i = j \circ f$. In other words, we obtain the following commutative diagram, which we call a* **completion diagram of** $f : S \to T$*:*

$$\begin{tikzcd} S \arrow[r, "i"] \arrow[d, "f"] & S' \arrow[d, "f'"] \\ T \arrow[r, "j"] & T' \end{tikzcd}$$

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 $a_0, \dots, a_k \in \mathrm{R}^n$ be an affine indepedent tuple. We say*

$$(a_0, \dots, a_k) = \left\{ \sum t_i a_i | t_i \geq 0, \sum t_i = 1 \right\} \subset \mathrm{R}^n$$

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

Definition 9. *A **simplicial complex** K in R^n is a finite collection K of simplicies in R^n such that for all simplicies $\sigma = (a_0, \dots, a_k)$ and $\tau = (b_0, \dots, b_l)$ 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 $X \subset \mathrm{R}^m$ be a definable set. A **triangulation** in R^n of X is a pair (Φ, K) consisting of a complex K in R^n and a definable homeomorphism $\Phi : X \to |K|$, where $|K|$ is the union of the simplices of K in 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 R^n if A is is a union of elements of $\Phi^{-1}(K)$.

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

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

- *$P = 0$ and $P \neq 0$, for $P \in \mathrm{R}[X_1, \dots, X_m]$ are formulas with set free variables $Free(P = 0) = Free(P \neq 0) = \{X_1, \dots, X_m\}$.*
- *If Φ_1 and Φ_2 are formulas, then $\Phi_1 \wedge \Phi_2$ and $\Phi_1 \vee \Phi_2$ are formulas 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 formulas 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 $Free(\Phi) = \{Y_1, \dots, Y_m\}$ is the set $Reali(\Phi) = \{y \in \mathrm{R}^m | \Phi(y) \text{ is true}\}$.*
*Two formulas 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*

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

where $Q_i \in \{\exists, \forall\}$ and Ψ is a quantifer free formula. In other words, all the quantified variables are written first in the formula. All formulas are equivalent to a formula written in prenex normal form.

For a more in-depth explanation of formulas and logic, see Introduction to Mathematical Logic by Elliott Mendelson [Mendelson, 2015].

With the definitions out of the way, we finish this section by introducing some common formulas 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 $\Phi_{\Delta_i}(\lambda_0, \ldots, \lambda_i) \leftarrow \sum \lambda_j = 1 \wedge \bigwedge_{j=0}^{i} \lambda_j > 0$ a formula whose realization is the standard i-simplex.

(3) Given formulas 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 $Q_i \in \{\exists, \forall\}$.

2.2. **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 3 (Theorem 2.77 [Basu et al., 2006a])**.** *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* R. *Then there is a quantifier free formula $\Psi(Y)$ with coefficients in D such that for every $y \in \mathrm{R}^k$, 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 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 [Basu et al., 2006a]. The result is described in the following theorem.

Theorem 4 (Theorem 14.16 [Basu et al., 2006a])**.** *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 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 $X_{[i]}$ has size n_i for $1 \le i \le \omega$. Given $\Phi(Y)$, a $(\mathcal{P},\Pi)$-formula, there exists a quantifier free formula*

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

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

$$I \le s^{(n_\omega+1)\cdots(n_1+1)(m+1)} d^{\mathcal{O}(n_\omega)\cdots\mathcal{O}(n_1)\mathcal{O}(m)},$$
$$J_i \le s^{(n_\omega+1)\cdots(n_1+1)(m+1)} d^{\mathcal{O}(n_\omega)\cdots\mathcal{O}(n_1)},$$
$$K_{ij} \le d^{\mathcal{O}(n_\omega)\cdots\mathcal{O}(n_1)},$$

and the degrees of the polynomials $P_{ijk}(y)$ 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.

2.3. **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 5 (Theorem 4.5 [Basu, 2007])**.** *Let $S \subset \mathrm{R}^m$ 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 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 $\mathcal{P} \subset \mathrm{R}[X_1,\ldots,X_m]$ containing k polynomials bounded by degree at most d, the the semi-algebraic triangulation (K,h) can be computed in time $(kd)^{2^{\mathcal{O}(m)}}$.

As can been, semi-algebraic triangulation has doubly exponential complexity. This is because the general algorithm for computing a semi-algebraic triangulation relies on computing a cylindrical decomposition, which requires doubly exponential computations. Determining a general algorithm to compute a semi-algebraic triangulation in singly exponential complexity is an open problem in algorithmic semi-algebraic geometry that is of great interest. There exist special cases where a triangulation can been found in singly exponential computations.

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

3. Proof of the Main Theorem

3.1. **Outline of the Proof.** We prove theorem 2 with an algorithm that follows the proof of theorem 1. We present here a brief summary of the steps in the proof of theorem 1. We reference the corresponding algorithms that line up with each step.

Proof. The effective steps of the proof of theorem 1 is contained in the General Quotient Algorithm 3.3.3 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 3.3.3, this set, call it X', is the realization of the formula $\Phi_{X'}$, described ere]. 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 3.3.3 again, hown 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 $\text{cl}(S_D)$ defined ere], to Y' with theorem 10.2.12 in [van den Dries, 1998] (which in our case is the Second Gluing Quotient Algorithm). The Second Gluing Quotient Algorithm 3.3.2 relies on both the First Gluing Quotient Algorithm 3.3.1 and the Completion Algorithm 3.2.4. Both of these rely on the Extension Algorithm 3.2.3, which uses the Semi-Algebraic Path Algorithm 3.2.2, which in turn relies on the Partition of Unity Algorithm 3.2.1. 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 3.3.3 define this new map with domain X and we are done, see ere]. $\square$

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.

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

3.2.1. *Partition of Unity Algorithm.*
Input($\mathcal{P} \subset \mathrm{R}[X_1, \dots, X_m]$, Φ_B a $\mathcal{P}$-formula describing a semi-algebraic set B, a family of $\mathcal{P}$ open formulas $\{\Phi_{U_i}\}_{i=1}^n$ describing a family of semi-algebraic subsets $\{U_i\}$ of B that cover B)

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

Procedure:

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

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

(b) Set

$$\widetilde{\Phi_{V_i}}(X) \leftarrow \exists X_1, X_2, t_1, t_2 \forall X_3, X_4 (\Phi_B(X)$$
$$\wedge M_{d_{A_0,B}}(X, t_1, X_1, X_3) \wedge M_{d_{A_1,B}}(X, t_2, X_2, X_4) \wedge t_2 < t_1).$$

(c) Let $\mathcal{Q}_0 = \emptyset$. Apply theorem 4 with inputs

$$(\mathcal{P} \cup \mathcal{Q}_{i-1} \subset \mathrm{R}[X_1, \ldots, X_m, X_1', \ldots, X_{4m}', t_1, t_2],$$
$$\Pi = [(X_1, \ldots, X_m), (X_1', \ldots, X_{2m}', t_1, t_2), (X_{2m+1}', \ldots, X_{4m}')], \widetilde{\Phi_{V_i}})$$

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

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

(e) Set

$$\widetilde{\Phi_{g_i}}(X, t) \leftarrow \exists X_1, X_2, \exists t' > 1, \forall X_3, X_4$$
$$[(t \leq 1 \wedge M_{d_{V_i^C,B}}(X, t, X_1, X_2)) \vee (t = 1 \wedge M_{d_{V_i^C,B}}(X, t', X_2, X_4))].$$

(f) Apply theorem 4 with inputs

$$(\mathcal{P} \cup \mathcal{Q}_i \subset \mathrm{R}[X_1, \ldots, X_{m+1}, X_1', \ldots, X_{4m}', t'],$$
$$\Pi = [(X_1, \ldots, X_{m+1}), (X_1', X_{2m}', t'), (X_{2m+1}', \ldots, X_{4m}')], \widetilde{\Phi_{g_i}})$$

to obtain a set of polynomials $\mathcal{Q}_i' \subset \mathrm{R}[X_1, \ldots, X_{m+1}]$ and an equivalent quantifier free $\mathcal{Q}_i'$ formula Φ_{g_i}.

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

(3) For $i = 1$ to n, set

$$\widetilde{\Phi_{f_i}}(X, t) \leftarrow \exists t_1, \ldots, t_n \left(\bigwedge_{j=1}^{n} \Phi_{g_j}(X, t_j) \wedge t \cdot \left(\sum_j t_j \right) = t_i \right).$$

(4) For $i = 1$ to n, apply theorem 4 with inputs

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

to obtain a set of polynomials $\mathcal{T}_i \subset \mathrm{R}[X_1, \ldots, X_{m+1}]$ and an equivalent quantifier free $\mathcal{T}_i$ formula Φ_{f_i}.

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

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

Complexity Analysis for Partition of Unity Algorithm:

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

(1) In step (1) we loop from $i = 1$ to n to obtain formulas Φ_{V_i} and Φ_{g_i}. Each iteration applies theorem 4 twice, returning two sets of polynomials $\mathcal{Q}_i$ and $\mathcal{Q}'_i$. Each formula and set of polynomials depends on the iteration prior, so we estabilish a reccurence relation. Let $q_i = |\mathcal{Q}_i|$ and $q'_i = |\mathcal{Q}'_i|$. Let $\mathfrak{d}_i$ and $\mathfrak{d}'_i$ bound the degrees of the polynomials of $\mathcal{Q}_i$ and $\mathcal{Q}'_i$, respectively. From the complexity analysis in [Basu et al., 2006a] of Algorithm 14.5, $q_i(k,d) \sim [q_{i-1}(k,d)\mathfrak{d}_{i-1}(k,d)]^{m^{\mathcal{O}(c)}}$. Similarly $\mathfrak{d}_i(k,d) = \mathfrak{d}_{i-1}(k,d)^{m^{\mathcal{O}(c)}}$. 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 $q_i(k,d) = (kd)^{m^{\mathcal{O}(ci)}}$ and $\mathfrak{d}_i(k,d) = d^{m^{\mathcal{O}(ci)}}$. With this, we see that $q'_i(k,d) = (q_i(k,d)\mathfrak{d}_i(k,d))^{(m+1)^{\mathcal{O}(c)}} = (kd)^{m^{\mathcal{O}(ci)}}$ and $\mathfrak{d}'_i(k,d) = \mathfrak{d}_i(k,d)^{(m+1)^{\mathcal{O}(c)}} = d^{m^{\mathcal{O}(ci)}}$. The computational complexity of step (1) of the algorithm is $(kd)^{\mathcal{O}(m^{cn})}$.
(2) In step (4), we apply theorem 4 n times. Each application inputs the set of polynomials $\mathcal{Q}' \subset \mathrm{R}[X_1,\ldots,X_{m+1}]$ and contains on the order of $q'_n(k,d) = (kd)^{m^{\mathcal{O}(cn)}}$ polynomials whose degrees are bounded by $\mathfrak{d}'_i(k,d) = d^{m^{\mathcal{O}(cn)}}$. 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
$$n(kd)^{m^{\mathcal{O}(cn)}}.$$

Proof of Correctness for Partition of Unity Algorithm: The correctness follows from Lemma 6.3.7 of [van den Dries, 1998] and from the correctness of Algorithm 14.5 from [Basu et al., 2006a].

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 formulas 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.

3.2.2. *Semi-Algebraic Path Algorithm.*
Input($\mathcal{P} \subset \mathrm{R}[X_1,\ldots,X_m]$, Φ_B a $\mathcal{P}$-formula describing a semi-algebraic set B, $\mathcal{P}$-closed formulas Φ_{A_0} and Φ_{A_1} describing disjoint closed semi-algebraic sets $A_0, A_1 \subset B$)

Output($\mathcal{Q} \subset \mathrm{R}[X_1,\ldots,X_{m+1}]$, 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 4 with inputs
$$(\mathcal{P} \subset \mathrm{R}[X_1, \ldots, X_m, X_1', \ldots, X_{4m}', t_1, t_2],$$
$$\Pi = [(X_1, \ldots, X_m), (X_1', \ldots, X_{2m}', t_1, t_2), (X_{2m+1}', \ldots, X_{4m}')], \widetilde{\Phi_{U_0}})$$
and
$$(\mathcal{P} \subset \mathrm{R}[X_1, \ldots, X_m, X_1', \ldots, X_{4m}', t_1, t_2],$$
$$\Pi = [(X_1, \ldots, X_m), (X_1', \ldots, X_{2m}', t_1, t_2), (X_{2m+1}', \ldots, X_{4m}')], \widetilde{\Phi_{U_1}})$$
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 $\mathcal{Q}' = \mathcal{Q}_1 \cup \mathcal{Q}_2 \subset \mathrm{R}[X_1, \ldots, X_m]$.
(4) Apply the Partition of Unity Algorithm 3.2.1 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 $\mathcal{Q}_3 \subset \mathrm{R}[X_1, \ldots, X_{m+1}]$ and $\mathcal{Q}_3$-formulas $\Phi_{f_0}, \Phi_{f_1}, \Phi_{f_2}$.
(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 4 with inputs
$$(\mathcal{P} \cup \mathcal{Q}_3 \subset \mathrm{R}[X_1, \ldots, X_{m+1}, X_1', \ldots, X_{4m}', t_1, t_2, t_3, t_4, t_5],$$
$$\Pi = [(X_1, \ldots, X_{m+1}), (X_1', \ldots, X_{2m}', t_1, t_2, t_3, t_4, t_5),$$
$$(X_{2m+1}', \ldots, X_{4m}')], \widetilde{\Phi_f})$$
to obtain a set of polynomials $\mathcal{Q} \subset \mathrm{R}[X_1, \ldots, X_{m+1}]$ 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 $\mathcal{P} \subset \mathrm{R}[X_1, \ldots, X_m]$ of degree at most d.

(1) In step (3) we apply quantifier elimination (theorem 4) 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 $\mathcal{Q}' = \mathcal{Q}_1 \cup \mathcal{Q}_2 \subset \mathrm{R}[X_1, \ldots, X_m]$ 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 3.2.1 with $n = 3$ and set of polynomials $\mathcal{P} \cup \mathcal{Q}'$. This has complexity $3(kd)^{m^{\mathcal{O}(c)}}$. This returns a set of polynomials $\mathcal{Q}'' \subset \mathrm{R}[X_1, \ldots, X_m]$ of size $3(kd)^{m^{\mathcal{O}(c)}}$ 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 $\mathcal{Q} \subset \mathrm{R}[X_1, \ldots, X_m]$ of size $(3kd)^{m^{\mathcal{O}(c)}}$ whose degrees are bounded by $d^{m^{\mathcal{O}(c)}}$.

(4) Therefore the total complexity of the algorithm is
$$(3kd)^{m^{\mathcal{O}(c)}}.$$

Proof of Correctness of Semi-Algebraic Path Algorithm: The correctness of this algorithm follows from the proof of Lemma 6.3.8 in [van den Dries, 1998], and from the correctness of Algorithm 14.5 in [Basu et al., 2006a] and the correctness of the Partition of Unity Algorithm 3.2.1.

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 $\mathcal{P}_1$ formulas 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 [van den Dries, 1998] that we are making a effective. We then provide a brief summary of the proofs, linking to the appropriate corresponding lines in the algorithm.

Theorem 6 (Theorem 8.3.3 [van den Dries, 1998]). *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 7 (Theorem 8.3.9 [van den Dries, 1998]). *Let A be a definable closed subset of the definable set $B \subset \mathrm{R}^m$. 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 8 (Theorem 8.3.10 [van den Dries, 1998]). *Let A be a definable closed subset of the definable set $B \subset \mathrm{R}^m$, for some m. Let $f : A \to C$ be a definable continuous map into a definable set $C \subset \mathrm{R}^n$, 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 lines [2]. We also need to define several functions. First, for each vertex $e \in \|(K)$, we define $\lambda_e : |K| \to [0,1]$ as follows: for $x \in (e_0, \ldots, e_k)$, where $(e_0, \ldots, e_k)$ a k-simplex of K set $\lambda_e(x) = \left\{ \begin{array}{ll} 0 & e \notin \{e_0, \ldots, e_k\} \\ \lambda_i & e = e_1 \end{array} \right.$, 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 convience, 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: $\omega_\sigma(x) = \left\{ \begin{array}{ll} 1 & \sigma \in L \\ 0 & \sigma \in K - L \end{array} \right.$. 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 8.3.3: let $r(x) : st_{K'}(|L|) \to |L|$ by $r(x) = \dfrac{\sum_{\substack{\sigma \subset cl(K) \\ \sigma \text{simplex}}} \mu_\sigma(x) \cdot b(\sigma)}{\sum_{\substack{\sigma \subset cl(K) \\ \sigma \text{ simplex}}} \mu_\sigma(x)}$ which we define in line [8]. This retraction has all the properties stated in the conclusion of theorem 8.3.3. 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 8.3.9, we restrict the retraction function from above to $cl(U) \cap K$, [14]. To obtain the function from 8.3.10 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]

3.2.3. *Extension Algorithm.*
Input($\mathcal{P}_1 \subset \mathrm{R}[X_1, \ldots, X_m]$, 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, $\mathcal{P}_2 \subset \mathrm{R}[X_1, \ldots, X_{2m+1}]$, Φ_φ a $\mathcal{P}_2$ formula representing the graph of a semi-algebraic contraction φ, $\mathcal{P}_3 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$, Φ_f a $\mathcal{P}_3$ formula representing the graph of a semi-algebraic map f from A to a set Y)

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

(1) Apply theorem 5 (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 $\mathcal{Q}_\sigma$ formulas $\Phi_{K,\sigma}$, for each simplex σ of K.
    By restricting to A, we also obtain formulas $\Phi_{L,\sigma}$ for each $\sigma \in L$. Let $\mathcal{Q}_1 = \bigcup_\sigma \mathcal{Q}_\sigma \subset \mathrm{R}[X_1, \ldots, X_{2m+1}]$.

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

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

$$\bigvee_{j_q=q}^{m} \bigvee_{\substack{\sigma_0 \subset \cdots \subset \sigma_q \in K \\ \sigma_i = (u_0, \ldots, u_{j_i}) \\ u_k \in \mathrm{Vert}(K) \\ j_0 \le j_1 \le \cdots \le j_q}} \Phi_{K,\sigma_q}(t_0, \ldots, t_{j_q}, X) \wedge$$

$$\bigwedge_{i=0}^{q} r_i = b(\sigma_i) \wedge s_0 r_0 + \cdots + s_q r_q = t_0 u_0 + \cdots + t_{j_q} u_{j_q}$$

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

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

$$\bigvee_{j_q=q}^{m} \bigvee_{\substack{\sigma_0 \subset \cdots \subset \sigma_q \in L \\ \sigma_i = (u_0, \ldots, u_{j_i}) \\ u_k \in \mathrm{Vert}(L) \\ j_0 \le j_1 \le \cdots \le j_q}} \Phi_{L,\sigma_q}(t_0, \ldots, t_{j_q}, X) \wedge$$

$$\bigwedge_{i=0}^{q} r_i = b(\sigma_i) \wedge s_0 r_0 + \cdots + s_q r_q = t_0 u_0 + \cdots + t_{j_q} u_{j_q}.$$

(3) Apply theorem 4 for each $\sigma \in b(K)$ and each $\sigma \in b(L)$ with inputs:

$$\big(\mathcal{Q}_1 \subset \mathrm{R}[X_1, \ldots, X_{2m+1}, X'_1, \ldots, X'_{m+1}],$$
$$\Pi = [(X_1, \ldots, X_{2m+1}), (X'_1, \ldots, X'_{m+1})], \widetilde{\Phi_{b(K),\sigma}}\big)$$

and

$$\big(\mathcal{Q}_1 \subset \mathrm{R}[X_1, \ldots, X_{2m+1}, X'_1, \ldots, X'_{m+1}],$$
$$\Pi = [(X_1, \ldots, X_{2m+1}), (X'_1, \ldots, X'_{m+1})], \widetilde{\Phi_{b(L),\sigma}}\big)$$

to obtain several sets of polynomials. Let $\mathcal{Q}_2$ denote the union of all sets of polynomials obtained for each simplex of $b(K)$, and $\mathcal{Q}_3$ for each simplex in $b(L)$. Then $\mathcal{Q}_2, \mathcal{Q}_3 \subset \mathrm{R}[X_1, \ldots, X_{2m+1}]$ and for each $\sigma \in b(K)$ we obtain equivalent quantifier free $\mathcal{Q}_2$ formulas $\Phi_{b(K),\sigma}$. Similarly for each $\sigma \in b(L)$, we obtain equivalent quantifer free $\mathcal{Q}_3$ formulas $\Phi_{b(L),\sigma}$.

(4) Set

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

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

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

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

(5) Apply theorem 4 with inputs

$$\big(\mathcal{Q}_1 \cup \mathcal{Q}_2 \subset \mathrm{R}[X_1, \ldots, X_m, X'_1, \ldots, X'_{4m+3}],$$
$$\Pi = [(X_1, \ldots, X_m), (X'_1, \ldots, X'_{4m+3})], \widetilde{\Phi_{st_{b(K)}(L)}}\big)$$

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

(6) For $\sigma \in K - L$, set $\Phi_{\mu_{\sigma,K,L}}(X,t) = (t=0)$.
For $\sigma = (v_0, \dots, v_p) \in L$, set

$$\Phi_{\mu_{\sigma,K,L}}(X,t) = \exists t_0, \dots, t_m \bigvee_{i=0}^{m} \bigvee_{\substack{\sigma' \in b(L) \\ \dim(\sigma')=i \\ \sigma'=(v'_0,\dots,v'_i)}} \Phi_{b(L),\sigma'}(t_0, \dots, t_i, X)$$

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

(7) For $\sigma \in L$, apply theorem 4 with inputs

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

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

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

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

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

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

(9) Apply theorem 4 with inputs

$$\big(\mathcal{Q}_5 \subset \mathrm{R}[X_1, \dots, X_{m+1}, X'_1, \dots, X'_{2m+2}],$$

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

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

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

(11) Set

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

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

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

(12) Apply theorem 4 with inputs

$$(\mathcal{P}_1 \cup \mathcal{Q}_1 \cup \mathcal{Q}_4 \subset \mathrm{R}[X_1, \dots, X_m, X'_1, \dots, X'_{4m}, t, t'],$$

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

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

(13) Apply the Semi-Algebraic Path Algorithm 3.2.2 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 $\mathcal{Q}_8 \subset \mathrm{R}[X_1, \ldots, X_{m+1}]$ 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

$$\begin{gathered}\widetilde{\Phi_{f'}}(X,Y) \leftarrow \exists Z_1, Z_2, t \forall \varepsilon > 0 \exists X' \\ [(M_r(X, Z_1, \varepsilon, X') \wedge \Phi_g(X,t) \wedge \Phi_f(Z_1, Z_2) \wedge \Phi_\varphi(Z_2, t, Y)) \\ \vee(\Phi_X(X) \wedge \neg \Phi_U(X) \wedge Y = 0)].\end{gathered}$$

(16) Apply theorem 4 with inputs

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

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

to obtain a set of polynomials $\mathcal{Q}_9 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ 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: $\mathcal{P}_1 \subset \mathrm{R}[X_1, \ldots, X_m]$ of size k_1 whose degrees are bounded by d_1, $\mathcal{P}_2 \subset \mathrm{R}[X_1, \ldots, X_{2m+1}]$ of size k_2 whose degrees are bounded by d_2, and $\mathcal{P}_3 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ of size k_3 whose degrees are bounded by d_3.

(1) In step (1) we apply the triangulation algorithm, theorem 5, once. This has complexity $(k_1 d_1)^{2^{\mathcal{O}(m)}}$. From this, we obtain a set of polynomials $\mathcal{Q}_1 \subset \mathrm{R}[X_1, \ldots, X_m]$ of size $(k_1 d_1)^{2^{\mathcal{O}(m)}}$ whose degrees are bounded by $d_1^{2^{\mathcal{O}(m)}}$.

(2) In step (3) we apply quantifier elimination to $\widetilde{\Phi_{b(K),\tau}}$ and $\widetilde{\Phi_{b(L),\tau}}$ for each $\tau \in K$ and L, respectively. Each application has complexity $(k_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}$. The number of simplices of K is bounded above by $(k_1 d_1)^{2^{\mathcal{O}(m)}}$. Each simplex of K and L, contributes up to $m^{\mathcal{O}(m)}$ simplices to $b(K)$ and $b(L)$. There fore $b(K)$ and $b(L)$ contain at most $(k_1 d_1)^{2^{\mathcal{O}(m)}} m^{\mathcal{O}(m)}$ simplices. Since we apply quantifier elimination for each simplex of $b(K)$ and $b(L)$, this step has complexity

$$m^{\mathcal{O}(m)} (k_1 d_1)^{2^{\mathcal{O}(m)}} (k_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}} = m^{\mathcal{O}(m)} (k_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}.$$

Let $\mathcal{Q}_2, \mathcal{Q}_3 \subset \mathrm{R}[X_1, \ldots, X_{2m+1}]$ denote the unions of the polynomials obtained from each $\tau \in b(K)$ and $b(L)$, respectively. Therefore $\mathcal{Q}_2$ and $\mathcal{Q}_3$ contain at most $m^{\mathcal{O}(m)} (k_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}$ polynomials with degrees bounded by $(d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}$.

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

$$m^{\mathcal{O}(m) m^{\mathcal{O}(c)}} (k_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}} = m^{m^{\mathcal{O}(c)}} (k_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}.$$

We return a set of polynomials $\mathcal{Q}_4 \subset \mathrm{R}[X_1, \ldots, X_m]$ of size $m^{m^{\mathcal{O}(c)}} (k_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}$ whose degrees are bounded by $(d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}$.

(4) In step (7), we apply quantifier elimination for each $\sigma \in L$. For each iteration, this has complexity $m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$. The number of simplices in L is bounded above by the number of simplices in K, so we perform this step at most $(k_1d_1)^{2^{\mathcal{O}(m)}}$ times. Hence the total complexity of this step is $m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$. Each iteration returns a set of polynomials. Let $\mathcal{Q}_5 \subset \mathrm{R}[X_1, \ldots, X_{m+1}]$ denote the union of these polynomials. Then $\mathcal{Q}_5$ contains at most $m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$ polynomials of degree at most $(d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$.

(5) In step (9) we apply quantifier elimination to $\widetilde{\Phi_{r'}}$. This has complexity
$$m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}.$$
We obtain a set of polynomials $\mathcal{Q}_6 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ containing at most $m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$.

(6) In step (12) we apply quantifier elimination to $\widetilde{\Phi_U}$. This has complexity $m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$ and returns a set of polynomials $\mathcal{Q}_7 \subset \mathrm{R}[X_1, \ldots, X_m]$ containing at most $m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$.

(7) In step (13), we apply the Semi-Algebraic Path Algorithm 3.2.2 with $\mathcal{P}_1 \cup \mathcal{Q}_7$ as the set of polynomials. This has complexity $m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$ and returns a set of polynomials $\mathcal{Q}_8 \subset \mathrm{R}[X_1, \ldots, X_{m+1}]$ containing at most $m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$.

(8) In step (16) we apply quantifier elimination to $\widetilde{\Phi_{f'}}$. This has complexity
$$\left[\left(k_2 + k_3 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + d_3 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$$
. It returns a set of polynomials $\mathcal{Q}_9 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ containing at most
$$\left[\left(k_2 + k_3 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + d_3 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$$
polynomials whose degrees are bounded by $(d_2 + d_3 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}})^{m^{\mathcal{O}(c)}}$.

(9) The last step dominates the complexity of the other steps, so the total complexity of Algorithm 8.3 is bounded above by
$$\left[\left(k_2 + k_3 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + d_3 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}.$$

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 [van den Dries, 1998], and from the correctness of Algorithm 14.5 [Basu et al., 2006a], the triangulation algorithm, and the Semi-Algebraic Path 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 $\mathcal{P}_1$ formulas, 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, $j : Y \to \mathrm{cl}(Y)$

is a completion of Y. Using this completion, we are able to obtain a completion diagram of f:

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

These sets are outputted by our algorithm as formulas $\Phi_{f'}$ 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.

3.2.4. *Completion Algorithm.*
Input($\mathcal{P}_1 \subset \mathrm{R}[X_1, \dots, X_m]$, 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 $\mathcal{P}_1$ bounded formula describing a semi-algebraic set Y, $\mathcal{P}_2 \subset \mathrm{R}[X_1, \dots, X_{2m}]$, Φ_f a $\mathcal{P}_2$ formula representing the graph of a semi-algebraic map from A to Y)

Output($\mathcal{Q} \subset \mathrm{R}[X_1, \dots, X_{2m}]$, $\Phi_{X'}$ a $\mathcal{Q}$ formula describing a completion X' of X, $\Phi_{A'}$ a $\mathcal{Q}$ formula describing a completion A' of A, $\mathcal{Q}' \subset \mathrm{R}[X_1, \dots, X_{3m}]$, $\Phi_{f'}$ 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 [Basu et al., 2006a] with input

$$(\mathcal{P}_1 \subset \mathrm{R}[X_1, \dots, X_m, X'_1, \dots, X'_{2m}, \varepsilon],$$
$$\Pi = [(X_1, \dots, X_m), (X'_1, \dots, X'_m), (\varepsilon), (X'_{m+1}, \dots, X'_{2m})],$$
$$\forall Y \exists \varepsilon > 0 \forall X (\Phi_Y(X) \wedge ||X - Y||^2 < \varepsilon^2 \rightarrow \Phi_{B(r)}(Y)))$$

for increasing values of $r = 1, 2, \dots$ 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 3.2.3 with inputs

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

to obtain a set of polynomials $\mathcal{Q}_1 \subset \mathrm{R}[X_1, \dots, X_{2m}]$ and a $\mathcal{Q}_1$ formula Φ_{f^*} representing the graph of a semi-algebraic map from X to $B(n)$.
(5) Set $\Phi_\mu(X_1, \dots, X_m, Y_1, \dots, Y_m) \leftarrow \bigwedge_{i=1}^{m} \left((2X_iY_i + 1)^2 = 1 + 4X_i^2\right)$, a formula representing the graph of a semi-algebraic homeomorphism from $\mathrm{R}^m \rightarrow (-1, 1)^m$.
(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 4 with inputs

$$(\mathcal{Q}_1 \cup \mathcal{P}_1 \subset \mathrm{R}[X_1, \dots, X_{2m}, X'_1, \dots, X'_{3m}\varepsilon],$$
$$\Pi = [(X_1, \dots, X_{2m}), (X'_1, \dots, X'_{3m}), (\varepsilon)], \widetilde{\Phi_{X'}})$$

to obtain a set of polynomials $\mathcal{Q}_2 \subset \mathrm{R}[X_1,\dots,X_{2m}]$ 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 4 with inputs

$$(\mathcal{P}_1 \cup \mathcal{Q}_1 \cup \mathcal{Q}_2 \subset \mathrm{R}[X_1,\dots,X_{2m},\varepsilon,X_1',\dots,X_{2m}'],$$

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

to obtain a set of polynomials $\mathcal{Q}_3 \subset \mathrm{R}[X_1,\dots,X_{2m}]$ 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 4 with inputs

$$(\mathcal{Q}_1 \subset \mathrm{R}[X_1,\dots,X_{2m},X_1',\dots,X_m'],$$

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

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

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

Complexity Analysis for Completion Algorithm: We input two families of polynomials: $\mathcal{P}_1 \subset \mathrm{R}[X_1,\dots,X_m]$ of size k_1 whose degrees are bounded by d_1, and $\mathcal{P}_2 \subset \mathrm{R}[X_1,\dots,X_{2m}]$ 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 $n(k_1d_1)^{m^{\mathcal{O}(c)}}$.

(2) In step (4) we apply the Extension Algorithm 3.2.3. This step has complexity

$$\left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}.$$

This returns a set of polynomials $\mathcal{Q}_1 \subset \mathrm{R}[X_1,\dots,X_{2m}]$ containing at most $\left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_2 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}})^{m^{\mathcal{O}(c)}}$.

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

$$\left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$$

and returns a set of polynomials $\mathcal{Q}_2 \subset \mathrm{R}[X_1,\dots,X_{2m}]$ containing at most $\left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_2 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}})^{m^{\mathcal{O}(c)}}$.

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

$$\left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$$

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

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

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

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

(6) Therefore the entire algorithm has complexity

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

Proof of Correctness of Completion Algorithm: The correctness of this algorithm follows from Lemma 10.2.7 in [van den Dries, 1998], and from the correctness of Algorithms 14.3 and 14.5 in [Basu et al., 2006a] and the Extension Algorithm.

3.3. **Quotient Algorithms.** 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 $\mathcal{P}_1$ formulas, 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 9 (Theorem 10.2.11 [van den Dries, 1998])**.** *Let $X \subset \mathrm{R}^m$ and $Y \subset \mathrm{R}^n$ be definable sets. Let $A \subset X$ be definable, closed, and bounded in the ambient space 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 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 R^M, line [6]. Let $\tilde{f} : X \to \mathrm{R}^M$ be a definable continuous extension of $f : A \to Y$, line [5,7] and let $d_A : \mathrm{R}^M \to \mathrm{R}$ be the distance function on A. Finally, we define a map $p : X \amalg Y \to \mathrm{R}^{2M+1}$ 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 $E(f)$. □

3.3.1. *First Gluing Quotient Algorithm.*
Input($\mathcal{P}_1 \subset \mathrm{R}[X_1, \ldots, X_m]$, 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 $\mathcal{P}_1$ formula

describing a semi-algebraic set Y, $\mathcal{P}_2 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$, Φ_f a $\mathcal{P}_2$ formula representing a semi-algebraic map from A to Y)

Output($\mathcal{Q} \subset \mathrm{R}[X_1, \ldots, X_{2m+3}]$, Φ_Z a $\mathcal{Q}_1$ formula describing the quotient space $X \amalg_f Y$, $\mathcal{Q}' \subset \mathrm{R}[X_1, \ldots, X_{3m+4}]$, 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 [Basu et al., 2006a] with inputs
$$(\mathcal{P}_1 \subset \mathrm{R}[X_1, \ldots, X_{2m}], \Pi = [(X_1, \ldots, X_m), (X_{m+1}, \ldots, X_{2m})], \exists X \Phi_A(X))$$
to determine if A is empty or not.

(2) If $A = \emptyset$, return
$$(\mathcal{P}_1, \Phi_X \vee \Phi_Y, \mathcal{P}_1(X_1, \ldots, X_m) \cup \mathcal{P}_1(X_{m+1}, \ldots, X_{2m}) \subset \mathrm{R}[X_1, \ldots, X_{2m}],$$
$$\Phi_{\mathbf{id}}(X, Y) \leftarrow (X = Y)).$$

(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 3.2.3 with inputs
$$(\mathcal{P}_1, \Phi_X, \Phi_A, \mathcal{P}_1(X) \cup \mathcal{P}_1(Y) \subset \mathrm{R}[X_1, \ldots, X_m, Y_1, \ldots, Y_m, t], \Phi_\varphi, \mathcal{P}_2, \Phi_f)$$
to obtain a set of polynomials $\mathcal{Q}_1 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ and a $\mathcal{Q}_1$ formula $\Phi_{f'}$ representing the graph of a semi-algebraic map f' from X to R^m.

(6) We need to identify X, Y, and A with their images in $X \amalg Y$. To this end we define formulas whose realizations are subsets of R^M as follows:
   (a) $\Phi_{X'}(X') \leftarrow \exists X \Phi_X(X) \wedge X' = (X, 1)$
   (b) $\Phi_{Y'}(Y') \leftarrow \exists Y \Phi_Y(Y) \wedge Y' = (Y, 2)$
   (c) $\Phi_{A'}(A') \leftarrow \exists A \Phi_A(A) \wedge A' = (A, 1)$

(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 4 with inputs
$$(\mathcal{P}_1 \cup \mathcal{Q}_1 \subset \mathrm{R}[(X_1, \ldots, X_{2m}, X_1', \ldots, X_{10m+4}'],$$
$$\Pi = [(X_1, \ldots, X_{2m}), (X_1', \ldots, X_{m+1}'), (X_{m+2}', \ldots, X_{10m+4}')], \widetilde{\Phi_p})$$
to obtain a set of polynomials $\mathcal{Q}_2 \subset \mathrm{R}[X_1, \ldots, X_{3M+1}]$ 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 4 with inputs

$$(\mathcal{Q}_2 \subset \mathrm{R}[X_1, \ldots, X_{3M+1}, X'_1, \ldots, X'_{m+1}],$$
$$\Pi = [(X_1, \ldots, X_{3M+1}), (X'_1, \ldots, X'_{m+1})], \widetilde{\Phi_Z})$$

to obtain a set of polynomials $\mathcal{Q}_3 \subset \mathrm{R}[X_1, \ldots, X_{2M+1}]$ 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 $\mathcal{P}_1 \subset \mathrm{R}[X_1, \ldots, X_m]$ of size k_1 whose degrees are bounded by d_1, and we input a set of polynomials $\mathcal{P}_2 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ 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 3.2.3 in step (5). This has complexity

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

and returns a set of polynomials $\mathcal{Q}_1 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ containing at most $\left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_2 + (d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}})^{m^{\mathcal{O}(c)}}$.

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

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

and returns a set of polynomials $\mathcal{Q}_2 \subset \mathrm{R}[X_1, \ldots, X_{3m+4}]$ containing at most $\left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_2 + (d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}})^{m^{\mathcal{O}(c)}}$.

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

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

and returns a set of polynomials $\mathcal{Q}_3 \subset \mathrm{R}[X_1, \ldots, X_{2m+3}]$ containing at most $\left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1 d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $(d_2 + (d_1)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}})^{m^{\mathcal{O}(c)}}$.

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

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

Proof of Correctness of the First Gluing Quotient Algorithm: The correctness of the algorithm follows from Lemma 10.2.11 in [van den Dries, 1998], and from the correctness of Algorithms 14.3 and 14.5 in [Basu et al., 2006a] and the Extension Algorithm 3.2.3.

By taking the above algorithm and applying the Completion Algorithm 3.2.4, 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 $\mathcal{P}_1$ formulas 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 formulas 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 10 (Theorem 10.2.12 [van den Dries, 1998])**.** *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 [van den Dries, 1998] 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 $E(f)$.

3.3.2. *Second Gluing Quotient Algorithm:*
Input($\mathcal{P}_1 \subset \mathrm{R}[X_1, \ldots, X_m]$, 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, $\mathcal{P}_2 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$, Φ_f a $\mathcal{P}_2$ formula representing the graph of a semi-algebraic map f from A to Y)

Output($\mathcal{Q}_1 \subset \mathrm{R}[X_1, \ldots, X_{4m+3}]$, Φ_Z a $\mathcal{Q}_1$ formula describing a semi-algebraic quotient Z of $X \amalg Y$, $\mathcal{Q}_2 \subset \mathrm{R}[X_1, \ldots, X_{6m+4}]$, 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 $\Phi_\mu(X_1, \ldots, X_m, Y_1, \ldots, Y_m) \leftarrow \bigwedge_{i=1}^{m} \left((2X_iY_i + 1)^2 = 1 + 4X_i^2\right)$, a formula representing the graph of a semi-algebraic homeomorphism from $\mathrm{R}^m \to (-1, 1)^m$.
(2) Set $\Phi_{Y_{bounded}}(Y) \leftarrow \exists X(\Phi_Y(X) \wedge \Phi_\mu(X, Y))$.
(3) Apply theorem 4 with inputs

$$(\mathcal{P}_1 \subset \mathrm{R}[X_1, \ldots, X_m, X'_1, \ldots, X'_m],$$
$$\Pi = [(X_1, \ldots, X_m), (X'_1, \ldots, X'_m)], \Phi_{Y_{bounded}})$$

and

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

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

(4) Apply the Completion Algorithm 3.2.4 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 $\mathcal{Q}_2 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ and $\mathcal{Q}_2$ formulas $\Phi_{X_{new}}$, $\Phi_{X'}$ and $\Phi_{A'}$, and a set of polynomials $\mathcal{Q}_3 \subset \mathrm{R}[X_1, \ldots, X_{3m}]$ 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 3.3.1 with inputs

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

to obtain a set of polynomials $\mathcal{Q}_4 \subset \mathrm{R}[X_1, \ldots, X_{4m+3}]$ and a $\mathcal{Q}_4$ formula $\Phi_{Z'}$ and a set of polynomials $\mathcal{Q}_5 \subset \mathrm{R}[X_1, \ldots, X_{6m+4}]$ 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

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

(7) Apply theorem 4 with inputs

$$(\mathcal{Q}_1 \cup \mathcal{Q}_2 \cup \mathcal{Q}_5 \subset \mathrm{R}[X_1, \ldots, X_{6m+4}, X'_1, \ldots, X'_{3m}],$$
$$\Pi = [(X_1, \ldots, X_{6m+4}), (X'_1, \ldots, X'_{3m})], \widetilde{\Phi_Z})$$

to obtain a set of polynomials $\mathcal{Q}_6 \subset \mathrm{R}[X_1, \ldots, X_{4m+3}]$ 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 4 with inputs

$$(\mathcal{Q}_1 \cup \mathcal{Q}_2 \cup \mathcal{Q}_5 \subset \mathrm{R}[X_1, \ldots, X_{6m+4}, X'_1, \ldots, X'_{3m}],$$
$$\Pi = [(X_1, \ldots, X_m), (X'_1, \ldots, X'_{3m})], \widetilde{\Phi_p})$$

to obtain a set of polynomials $\mathcal{Q}_7 \subset \mathrm{R}[X_1, \ldots, X_{6m+4}]$ 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 $\mathcal{P}_1 \subset \mathrm{R}[X_1, \ldots, X_m]$ of size k_1 whose degrees are bounded by d_1, and we input a set of polynomials $\mathcal{P}_2 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ 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 $\Phi_{cl(\widetilde{Y_{bounded}})}$. Each application adds complexity $(k_1d_1)^{m^{\mathcal{O}(c)}}$ and returns a set of polynomials. We let $\mathcal{Q}_1 \subset \mathrm{R}[X_1, \dots, X_m]$ be the union of the two returned sets of polynomials. $\mathcal{Q}_1$ contains at most $(k_1d_1)^{m^{\mathcal{O}(c)}}$ polynomials whose degrees are bounded by $d_1^{m^{\mathcal{O}(c)}}$.

(2) In step (4) we apply the Completion Algorithm 3.2.4. This adds complexity

$$\left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$$

and returns two set of polynomials. First $\mathcal{Q}_2 \subset \mathrm{R}[X_1, \dots, X_{2m}]$ containing at most $\left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$ polynomials of degree at most $(d_2+(d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}})^{m^{\mathcal{O}(c)}}$. Second $\mathcal{Q}_3 \subset \mathrm{R}[X_1, \dots, X_{3m}]$ with the same cardinality and degree bounds as $\mathcal{Q}_2$.

(3) In step (5) we apply the First Gluing Quotient Algorithm 3.3.1. 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

$$k_1^* = k_2^* = \left[\left(k_2 + \left(m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right)\left(d_2 + (d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{m^{\mathcal{O}(c)}}$$

and let $d_1^* = d_2^* = (d_2 + (d_1)^{2^{\mathcal{O}(m)}(k_1d_1)^{2^{\mathcal{O}(m)}}m^{\mathcal{O}(c)}})^{m^{\mathcal{O}(c)}}$. Applying this algorithm has complexity

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

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

$$m^{m^{\mathcal{O}(c)}}\left[\left(k_2 + m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\left(d_2 + d_1^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}.$$

This returns two sets of polynomials: $\mathcal{Q}_4 \subset \mathrm{R}[X_1, \dots, X_{4m+3}]$ and $\mathcal{Q}_5 \subset \mathrm{R}[X_1, \dots, X_{6m+4}]$. Both sets of polynomials have cardinality on the order $m^{m^{\mathcal{O}(c)}}\left[\left(k_2 + m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\left(d_2 + d_1^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$ with degrees bounded by $\left(d_2 + d_1^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$.

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

$$m^{m^{\mathcal{O}(c)}}\left[\left(k_2 + m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\left(d_2 + d_1^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$$

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

$$m^{m^{\mathcal{O}(c)}}\left[\left(k_2 + m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\left(d_2 + d_1^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$$

polynomials whose degrees are bounded by $(d_1^*)^{2^{\mathcal{O}(m)}(k_1^*d_1^*)^{2^{\mathcal{O}(m)}}m^{\mathcal{O}(c)}}$.

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

$$m^{m^{\mathcal{O}(c)}}\left[\left(k_2+m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\left(d_2+d_1^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$$

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

$$m^{m^{\mathcal{O}(c)}}\left[\left(k_2+m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\left(d_2+d_1^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}$$

polynomials whose degrees are bounded by $(d_1^*)^{2^{\mathcal{O}(m)}(k_1^*d_1^*)^{2^{\mathcal{O}(m)}}m^{\mathcal{O}(c)}}$.

(6) Therefore the entire algorithm has complexity

$$m^{m^{\mathcal{O}(c)}}\left[\left(k_2+m^{m^{\mathcal{O}(c)}}(k_1d_1)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\left(d_2+d_1^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}.$$

Proof of Correctness of the Second Gluing Quotient Algorithm: The correctness of the algorithm follows from Proposition 10.2.12 in [van den Dries, 1998], and from the correctness of Algorithm 14.5 [Basu et al., 2006a], the Completion Algorithm 3.2.4, and the First Gluing Quotient Algorithm 3.3.1.

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 1.

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

Output($\mathcal{Q}_1 \subset \mathrm{R}[X_1,\ldots,]$, Φ_f a $\mathcal{Q}_1$ formula describing the graph of the map from X to the semi-algebraic realization of X/E, $\mathcal{Q}_2 \subset \mathrm{R}[X_1,\ldots,]$, $\Phi_{X/E}$ 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 [Basu et al., 2006b] 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 4 with inputs

$$(\mathcal{P}_2 \subset \mathrm{R}[X_1,\ldots,X_{2m},X_1',\ldots,X_m'],$$
$$\Pi = [(X_1,\ldots,X_{2m}),(X_1',\ldots,X_m')],\widetilde{\Phi_f})$$

   to obtain a set of polynomials $\mathcal{Q}_1 \subset \mathrm{R}[X_1,\ldots,X_{2m}]$ 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 4 with inputs

$$(\mathcal{Q}_1 \subset \mathrm{R}[X_1,\dots,X_m,X'_1,\dots,X'_m],$$
$$\Pi = [(X_1,\dots,X_m),(X'_1,\dots,X'_m)], \widetilde{\Phi_{(X/E)}})$$

to obtain a set of polynomials $\mathcal{Q}_2 \subset \mathrm{R}[X_1,\dots,X_m]$ 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 5 with inputs $(\mathcal{P}_1, \Phi_X)$ to obtain a triangulation (h,K) of X. We obtain sets of polynomials $\mathcal{Q}_\sigma$ and $\mathcal{Q}_\sigma$ formulas $\Phi_{K,\sigma}$, for each simplex σ of K. Let $\mathcal{Q}_2 = \bigcup_\sigma \mathcal{Q}_\sigma \subset \mathrm{R}[X_1,\dots,X_{2m+1}]$.
(b) Set $\Phi_S(X) \leftarrow \forall Z(\Phi_E(X,Z) \Rightarrow X \le Z)$.
(c) Set $\widetilde{\Phi_{S_D}}(X) \leftarrow \exists t_0,\dots,t_D \forall Z \bigvee_{\substack{\sigma\in K\\ \dim(\sigma)=D}} \Phi_{K,\sigma}(t_0,\dots,t_D,X) \wedge M_S(X,Z)$.
(d) Apply theorem 4 with inputs

$$(\mathcal{P}_2 \cup \mathcal{Q}_2 \subset \mathrm{R}[X_1,\dots,X_{2m},X'_1,\dots,X'_{m+D+1}],$$
$$\Pi = [(X_1,\dots,X_{2m}),(X'_1,\dots,X'_m),(X'_{m+1},\dots,X'_{m+D+1})], \widetilde{\Phi_{S_D}})$$

to obtain a set of polynomials $\mathcal{Q}_3 \subset \mathrm{R}[X_1,\dots,X_m]$ 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 4 with inputs

$$(\mathcal{P}_2 \cup \mathcal{Q}_3 \subset \mathrm{R}[X_1,\dots,X_{2m},X'_1,\dots,X'_m,\varepsilon],$$
$$\Pi = [(X_1,\dots,X_{2m}),(X'_1,\dots,X'_m),(\varepsilon)], \widetilde{\Phi_{X'}})$$

to obtain a set of polynomials $\mathcal{Q}_4 \subset \mathrm{R}[X_1,\dots,X_m]$ 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 3.3.3 with inputs

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

to obtain two sets of polynomials $\mathcal{Q}_5 \subset \mathrm{R}[X_1,\dots,X_{4^{D-1}m+4^{D-1}-1}]$ and $\mathcal{Q}_6 \subset \mathrm{R}[X_1,\dots,X_{(4^{D-1}+1)m+4^{D-1}-1}]$ 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 4 with inputs

$$(\mathcal{P}_1 \cup \mathcal{Q}_3 \subset \mathrm{R}[X_1,\dots,X_m,X'_1,\dots,X'_m,\varepsilon],$$
$$\Pi = [(X_1,\dots,X_m),(X'_1,\dots,X'_m),(\varepsilon)],$$
$$\forall \varepsilon > 0 \exists X' M_{\mathrm{cl}(S_D)}(X,X',\varepsilon) \wedge \Phi_X(X)]$$

to obtain a set of polynomials $\mathcal{Q}_7 \subset \mathrm{R}[X_1,\dots,X_m]$ 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 3.3.2 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 $\mathcal{Q}_8 \subset \mathrm{R}[X_1, \ldots, X_{4^D m + 4^D - 1}]$ and $\mathcal{Q}_9 \subset \mathrm{R}[X_1, \ldots, X_{2(4^D m + 4^D - 1)}]$ 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 4 with inputs

$$(\mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{Q}_6 \cup \mathcal{Q}_7 \cup \mathcal{Q}_8 \cup \mathcal{Q}_9$$
$$\subset \mathrm{R}[X_1, \ldots, X_{2(4^D m + 4^D - 1)}, X'_1, \ldots, X'_{(4^D+1)m + 4^D - 1}],$$
$$\Pi = [(X_1, \ldots, X_{2(4^D m + 4^D - 1)}), (X'_1, \ldots, X'_{(4^D+1)m + 4^D - 1})], \widetilde{\Phi_f})$$

to obtain a set of polynomials $\mathcal{Q}_{10} \subset \mathrm{R}[X_1, \ldots, X_{(4^D+1)m + 4^D - 1}]$ 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 $\dim(X) = 0$, 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 $\mathcal{Q}_1 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ 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 quantifer elimination to $\widetilde{\Phi_{X/E}}$. This step has complexity $(k_2 d_2)^{m^{\mathcal{O}(c)}}$ and returns a set of polynomials $\mathcal{Q}_2 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ 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 $\mathcal{Q}_2 \subset \mathrm{R}[X_1, \ldots, X_{2m+1}]$ 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 $\mathcal{Q}_3 \subset \mathrm{R}[X_1, \ldots, X_m]$ containing at most $\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})\right]^{m^{\mathcal{O}(c)}}$ 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 $\mathcal{Q}_4 \subset \mathrm{R}[X_1, \ldots, X_m]$ 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 $\mathcal{Q}_4 \subset \mathrm{R}[X_1, \ldots, X_m]$ 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 $\mathcal{P}_2 \cup \mathcal{Q}_4 \subset \mathrm{R}[X_1, \ldots, X_{2m}]$ 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:

$$\begin{aligned}&\left[(k_2^* + (k_1^*)^{2^{\mathcal{O}(m)}})(d_2^* + (d_1^*)^{2^{\mathcal{O}(m)}})\right]^{m^{\mathcal{O}(c)}} \\ &= [(k_1^*)^{2^{\mathcal{O}(m)}}(d_1^*)^{2^{\mathcal{O}(m)}}]^{m^{\mathcal{O}(c)}} = (k_1^* d_1^*)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}}.\end{aligned}$$

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

$$\begin{aligned}&\left(\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})(d_2 + d_1^{2^{\mathcal{O}(m)}})^{m^{\mathcal{O}(c)}}\right]^{m^{\mathcal{O}(c)}}\right)^{2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}} \\ &= (k_2 + k_1^{2^{\mathcal{O}(m)}})^{m^{\mathcal{O}(2c)} 2^{\mathcal{O}(m)}} (d_2 + d_1^{2^{\mathcal{O}(m)}})^{2m^{\mathcal{O}(2c)} 2^{\mathcal{O}(m)}}.\end{aligned}$$

From this analysis, we see that with each iteration through the first part of the algorithm we end up with another copy of $2^{\mathcal{O}(m)} m^{\mathcal{O}(c)}$ in the exponent and we increase the the coefficient of the exponent of the degree term by 1 each time. Hence, after up to D iterations of items (4-6) the complexity will be

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

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

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

polynomials with degrees bounded by

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

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:

$$\begin{aligned}
&m^{m^{\mathcal{O}(c)}}\left[\left(k_2^* + m^{m^{\mathcal{O}(c)}}(k_1^*d_1^*)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\left(d_2^* + (d_1^*)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}} \\
&= m^{m^{\mathcal{O}(c)}}\left[\left(m^{m^{\mathcal{O}(c)}}(k_1^*d_1^*)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\left((d_1^*)^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}}\right)\right]^{2^{\mathcal{O}(m)}m^{\mathcal{O}(c)}} \\
&= m^{m^{\mathcal{O}(2c)}2^{\mathcal{O}(m)}}(k_1^*d_1^*)^{2^{\mathcal{O}(2m)}m^{\mathcal{O}(2c)}}.
\end{aligned}$$

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:

$$\begin{aligned}
&m^{m^{\mathcal{O}(2c)}2^{\mathcal{O}(m)}}\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})^{m^{\mathcal{O}(D)}2^{\mathcal{O}(Dm)}}(d_2 + d_1^{2^{\mathcal{O}(m)}})^{Dm^{\mathcal{O}(D)}2^{\mathcal{O}(Dm)}}(d_2 + d_1^{2^{\mathcal{O}(m)}})^{Dm^{\mathcal{O}(D)}2^{\mathcal{O}(Dm)}}\right]^{2^{\mathcal{O}(2m)}m^{\mathcal{O}(2c)}} \\
&= m^{m^{\mathcal{O}(2c)}2^{\mathcal{O}(m)}}\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})^{m^{\mathcal{O}(2D)}2^{\mathcal{O}(2Dm)}}(d_2 + d_1^{2^{\mathcal{O}(m)}})^{2Dm^{\mathcal{O}(2D)}2^{\mathcal{O}(2Dm)}}\right].
\end{aligned}$$

We then finish the iteration by applying quantifier elimination to $\widetilde{\Phi_f}$. This adds a factor of $m^{\mathcal{O}(c)}$ in the exponents:

$$m^{m^{\mathcal{O}(3c)}2^{\mathcal{O}(m)}}\left[(k_2 + k_1^{2^{\mathcal{O}(m)}})^{m^{\mathcal{O}(3D)}2^{\mathcal{O}(2Dm)}}(d_2 + d_1^{2^{\mathcal{O}(m)}})^{2Dm^{\mathcal{O}(3D)}2^{\mathcal{O}(2Dm)}}\right].$$

From this, we see that each iteration increase the exponents $m^{\mathcal{O}(c)}$ and $2^{\mathcal{O}(m)}$ 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 linearly, while also adding a linear factor on 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 $4^Dm + 4^D - 1 \sim \mathcal{O}(4^Dm)$ variables. Therefore, the total complexity after up to D iterations of reapplying algorithm 10.2.15 will be:

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

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

(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 $\mathcal{Q}_7 \subset \mathrm{R}[X_1, \ldots, X_m]$ 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 $m^{\mathcal{O}(c)}$ and $2^{\mathcal{O}(m)}$, so the complexity of this step is still $\mathcal{C}$. It returns a set of polynomials $\mathcal{Q}_8 \subset \mathrm{R}[X_1, \ldots, X_{4^D m + 4^D - 1}]$ and $\mathcal{Q}_9 \subset \mathrm{R}[X_1, \ldots, X_{2(4^D m + 4^D - 1)}]$ containing on the order of $\mathcal{C}$ polynomials of degree at most $(d_2 + d_1^{2^{\mathcal{O}(m)}})^{D^2 m^{\mathcal{O}(D^2)} 2^{\mathcal{O}(D^2 m)}}$.

(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 $\mathcal{Q}_{10} \subset \mathrm{R}[X_1, \ldots, X_{(4^D+1)m + 4^D - 1}]$ containing at most $\mathcal{C}$ 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

$$\mathcal{O}(4^D m)^{\mathcal{O}(4^D m)^{\mathcal{O}(D)} 2^{\mathcal{O}(Dm)}} \left[(k_2 + k_1^{2^{\mathcal{O}(m)}})^{m^{\mathcal{O}(D^2)} 2^{\mathcal{O}(D^2 m)}} (d_2 + d_1^{2^{\mathcal{O}(m)}})^{D^2 m^{\mathcal{O}(D^2)} 2^{\mathcal{O}(D^2 m)}}\right].$$

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

(13) In the first case, after simplifying, we see that the complexity is

$$2^{2^{\mathcal{O}(m^2)} m^{\mathcal{O}}(m)} m^{m^{\mathcal{O}(m)}} (kd)^{2^{\mathcal{O}(m^3)} m^{\mathcal{O}(m^2)}}$$

where $k \approx k_2 \approx k_1^{2^{\mathcal{O}(m)}}$ and $d \approx d_2 \approx d_1^{2^{\mathcal{O}(m))}}$.

(14) In the second case, the complexity reduces to

$$2^{2^{\mathcal{O}(m^2)} m^{\mathcal{O}}(m)} m^{m^{\mathcal{O}(m)}} (k_2 d_2)^{2^{\mathcal{O}(m^3)} m^{\mathcal{O}(m^2)}}.$$

(15) Finally, in the last case the complexity becomes

$$2^{2^{\mathcal{O}(m^2)} m^{\mathcal{O}}(m)} m^{m^{\mathcal{O}(m)}} (k_1 d_1)^{2^{\mathcal{O}(m^4)} m^{\mathcal{O}(m^2)}}.$$

Proof of Correctness of the General Quotient Algorithm: The correctness of the algorithm follows from the proof Theorem 10.2.15 in [van den Dries, 1998] and from the correctness of Algorithm 4 in [Basu et al., 2006b], Algorithm 14.5 in [Basu et al., 2006a], and the Second Gluing Quotient Algorithm 3.3.2.

4. Conclusion

In this paper, 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.

Doubly exponential complexity is never desirable if it can be avoided, so future work in this area can focus on developing a more effecient algorithm. In particular, Betti numbers of Reeb spaces of definable maps have been shown to have singly exponential upper bounds in terms of the complexity of the proper semi-algebraic map that defines it. It is widely regarded as a "meta" theorem in algorithmic semi-algebraic geometry that upper bounds on topological complexity are closely related to worst case scenario complexity of algorithms that compute the topological invariants of such objects. In light of this, searching for an algorithm to compute the Reeb space of a proper semi-algebraic map that is singly exponential is likely to bear fruit.

References

[Basu, 2007] Basu, S. (2007). Algorithmic semi-algebraic geometry and topology - recent progress and open problems.

[Basu et al., 2020] Basu, S., Cox, N., and Percival, S. (2020). On the reeb spaces of definable maps.

[Basu et al., 2006a] Basu, S., Pollack, R., and Roy, M.-F. (2006a). *Algorithms in Real Algebraic Geometry*, volume 10. Springer, 2nd edition.

[Basu et al., 2006b] Basu, S., Pollack, R., and Roy, M.-F. (2006b). Computing the dimension of a semi-algebraic set. *Journal of Mathematical Sciences*, 134(5):2346–2353.

[Burlet and de Rham, 1974] Burlet, O. and de Rham, G. (1974). Sur certaines applications génériques d'une variété sur certaines applications génériques d'une variété close à 2 dimensions dans le plan. *L'Ensiegnement Mathématique*, 20:275–292.

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

[Kirwan, 1986] Kirwan, F. (1986). Rational intersection cohomology of quotient varieties. *Inventiones mathematicae*, 86(3):471–505.

[Mendelson, 2015] Mendelson, E. (2015). *Introduction to Mathematical Logic*. CRC Press, sixth edition.

[Mumford et al., 1994] Mumford, D., Kirwan, F., and Fogarty, J. (1994). *Geometric Invariant Theory*. Berlin: Springer-Verlag, 3rd edition.

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

[van den Dries, 1998] van den Dries, L. (1998). *Tame Topology and O-minimal Structures*, volume 248 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge.

Department of Mathematics, Purdue University, 150 N. University Street, West Lafayette, IN 47907, U.S.A

Email address: cox175@purdue.edu