ON ARF RINGS

A THESIS
SUBMITTED TO THE DEPARTMENT OF MATHEMATICS
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Sefa Feza Arslan
September, 1994

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Sinan Sertöz(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Hurşit Önsiper

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Varol Akman.

Assoc. Prof. Dr. Varol Akman

Approved for the Institute of Engineering and Sciences:

M. Baray

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

ON ARF RINGS

Sefa Feza Arslan
M.S. in Mathematics
Advisor: Asst. Prof. Dr. Sinan Sertöz
September, 1994

In this thesis, we worked with curves which have cusp type singularities. We described the Arf theory, which solves the problem of understanding and finding the multiplicity sequence of a curve branch algebraically. We proposed an algorithm for finding the Arf characters of a given curve branch. We also faced the problem of Frobenius, and proposed an algorithm for the solution of problem of Frobenius in the most general case.

Keywords : Curve branch, singularity, blow up, multiplicity sequence, Arf ring, Arf semigroup, Arf closure, Arf characters, Frobenius.

ÖZET

ARF HALKALARI

Sefa Feza Arslan
Matematik Bölümü Yüksek Lisans
Danışman: Asst. Prof. Dr. Sinan Sertöz
Eylül, 1994

Bu tezde, köşe noktası biçiminde tekillikleri olan eğrilerle ilgilendik. Bir eğri kolunun çokkatlılık dizisinin anlaşılması ve bulunması sorununu cebirsel olarak çözen Arf kuramını tanıttık. Verilen bir eğri kolunun Arf karakterlerini bulan bir algoritma önerdik. Ayrıca, Frobenius problemi ile de karşılaştık ve en genel durumdaki çözümü için bir algoritma önerdik.

Anahtar Kelimeler : Eğri kolu, tekillik, tekilliğin çözülmesi, çokkatlılık dizisi, Arf halkası, Arf yarıgrubu, Arf kapanışı, Arf karakterleri, Frobenius.

ACKNOWLEDGMENTS

I would like to thank to Asst. Prof. Dr. Sinan Sertöz for his supervision, for his continued guidance, for his readiness to help at all times, for his critical comments while reading the written work, and for his encouragement through the development of this thesis.

I would like to thank Berna, who gave me hope with her love and support when I found myself in times of trouble.

I would also like to thank Göksen for his great help in computer graphics, and for his friendship.

I would like to thank my family for their love and support.

It is a pleasure to express my thanks to all my friends, with whom I shared everything; good times, bad times, agonies, hopes, and utopias.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

The link between algebra and geometry makes it possible to predict algebraically the result of a geometric process. In this way, a geometric problem can be solved by algebraic methods and computations. Also, invariants of a geometric object can be found algebraically; these are very important for classification.

We will deal with curves, which have cusp type singularities. Singular curves can be classified by finding nonsingular curves, which are birationally equivalent to these singular curves. This can be done by a blow up process. For cusp type singularities, one blow up may not be sufficient to remove the singularity. Hence, successive blow ups must be applied to obtain a nonsingular curve. The multiplicity sequence constructed by taking the multiplicity of the singularity before each blow up is a fundamental invariant of the singularity. Arf shows that the completion of the local ring at the singularity of the branch carries all the information necessary to obtain the multiplicity sequence [2].

Arf passes from geometry to algebra by using the completion of the local ring. He constructs the canonical closure of this ring, later called the Arf closure. The orders of the elements of this ring form a sub-semigroup of the natural numbers. In this way, Arf passes from algebra to arithmetic. From this semigroup, Arf obtains some numbers by a process to be described later, and then he determines the multiplicity sequence of the curve branch by applying the modified Jaccobian algorithm [8, pp. 108-109] to these numbers. These numbers are called the Arf characters of that curve branch.

The purpose of this thesis is to describe the work of Arf and to propose a computer algorithm for finding the Arf characters of a given curve branch.

Because of the the sub-semigroup mentioned above, we face the problem of finding the largest integer which is not included in this semigroup, if the generators of this semigroup are relatively prime. This is the famous problem of Frobenius. We also propose an algorithm for the solution of the problem of Frobenius in the most general case.

In chapter 2, we give the necessary preliminaries for understanding the Arf theory. We describe the category in which we will be working, by defining its objects and morphisms. Then we describe curves, singularities, and resolution of singularities, and give examples.

In chapter 3, we present a history of the problem of obtaining the multiplicity sequence of a singular curve branch without applying succesive blow ups to the curve. Then we describe the work of Arf and his solution to this problem.

In chapter 4, we offer a literature review of the problem of Frobenius. This involves looking for the largest integer that is not included in the semigroup generated by relatively prime integers. Then we propose an algorithm for the solution of the problem of in the most general case.

In chapter 5, to find the Arf characters of a curve branch we propose an algorithm applicable to computer', by depending on the work of Arf. In this algorithm, given a parameterization of a curve branch as input, its Arf characters are obtained as output.

We use the following notation throughout:

$\mathbb{R}$ = Real numbers

$\mathbb{N}$ = {0,1,2,...}

Chapter 2

CURVES, SINGULARITIES, AND RESOLUTION OF SINGULARITIES

2.1 Background

We will be working in the category of algebraic varieties and birational maps over an algebraically closed field. Our main construction is a blowing up of an algebraic variety, which is the main example of a birational map. We recall some definitions.

Let k be an algebraically closed field.

Definition 2.1.1. An *affine n-space* over k is defined to be the set of all n-tuples, components of which are from k. An affine n-space over k is denoted by $\mathbb{A}_k^n$, and by $\mathbb{A}^n$ if no confusion about the field arises.

It looks as if $\mathbb{A}^n$ and k^n are the same but there is a major difference. k^n has an origin and a vector space structure, but $\mathbb{A}^n$ is just a set of points. In k^n the origin is a distinguished point, however in $\mathbb{A}^n$ all points are considered with equal attention and no point is distinguished.

Let $A = k[x_1, ..., x_n]$ be the polynomial ring in n variables over k. If f is a polynomial in A and $a = (a_1, ..., a_n)$ is an element of the affine space $\mathbb{A}^n$, then $f(a)$ is defined as $f(a_1, ..., a_n)$. The zeros of a polynomial $f \in A$ is the set of all $a \in \mathbf{A}^n$ satisfying $f(a) = 0$.

If S is a collection of polynomials from A, then

$$Z(S) = \{a \in \mathbb{A}^n \mid f(a) = 0 \text{ for all } f \in S\},$$

is the set of all simultaneous solutions of the polynomials in S.

If I is the ideal generated by S, then one can show that $Z(S) = Z(I)$. It follows from *Hilbert's basis theorem* that $A = k[x_1, ..., x_n]$ is a Noetherian ring. (See for example [3, p. 81].) Since A is a Noetherian ring, every ideal of it is finitely generated. Hence, it is possible to express $Z(S)$ as the common zeros of a finite number of polynomials $f_1, ..., f_m$.

Also, for any subset $X \subseteq \mathbb{A}^n$, the *ideal* of X in A can be defined by

$$I(X) = \{f \in A \mid f(P) = 0 \text{ for all } P \in X\}.$$

Definition 2.1.2. A subset V of $\mathbb{A}^n$ is an *algebraic set* if there is a subset $S \subseteq A$ which satisfies $V = Z(S)$.

By using the algebraic sets, we can define a topology on $\mathbb{A}^n$. This is done by taking closed sets as algebraic sets and open sets as the complements of these algebraic sets.

Definition 2.1.3. The topology defined on $\mathbb{A}^n$ by taking closed sets as algebraic sets and open sets as the complement of these algebraic sets is called the *Zariski topology* on $\mathbb{A}^n$.

Let us prove that this is indeed a topology.

We must show that the following three propositions are satisfied. *(i)* Finite intersections of open sets are open, *(ii)* arbitrary (finite or infinite) unions of open sets are open, and *(iii)* the empty set and whole space are open. In order to prove *(i)*, it is sufficient to show that the union of two closed sets is closed. Let X_1, X_2 be two closed sets. Since X_1 and X_2 are algebraic, they can be written as $Z(S_1)$ and $Z(S_2)$ for some subsets S_1, S_2 of A. Then $X_1 \cup X_2 = Z(S_1S_2)$, where S_1S_2 is the set of all products of elements of S_1 by S_2. In fact, if $P \in X_1 \cup X_2$, then $P \in Z(S_1)$ or $P \in Z(S_2)$. Hence, P is a zero of every polynomial in S_1S_2 and $P \in Z(S_1S_2)$. This proves that $X_1 \cup X_2 \subset Z(S_1S_2)$. Conversely, if $P \in Z(S_1S_2)$ and P is not an element of X_1, then there is $f_1 \in S_1$ such that $f_1(P) \neq 0$. But for any $f_2 \in S_2$, $(f_1 f_2)(P) = 0$ since $P \in Z(S_1 S_2)$, so $f_2(P)$ must be zero. Hence $P \in X_2 \subset X_1 \cup X_2$. To show *(ii)*, it is sufficient to prove that arbitrary intersections of algebraic sets are algebraic. If $X_n = Z(S_n)$ is any family of algebraic sets and $P \in \cap X_n$, then for every n, all the polynomials in S_n are zero at P. This shows that $P \in Z(\cup S_n)$. Hence $\cap X_n \subset Z(\cup S_n)$. Conversely, if $P \in Z(\cup S_n)$, then for every n, $P \in Z(S_n)$. This shows that $P \in \cap Z(S_n) = \cap X_n$. Thus $Z(\cup S_n) \subset \cap X_n$. This proves that $\cap X_n = Z(\cup S_n)$. Finally to show *(iii)* note that, since $\emptyset = Z(1)$, it is an algebraic set and its complement $\mathbb{A}^n$ is open. In the same way, $\mathbb{A}^n = Z(0)$ is an algebraic set, and its complement $\emptyset$ is open.

This establishes the Zariski topology on $\mathbb{A}^n$. From now on whenever a topology is referred to, we will mean Zariski topology, unless otherwise stated.

Definition 2.1.4. A nonempty subset X of a topological space T is irreducible if it cannot be expressed as the union of two proper subsets which are closed in Y. Hence, a set $V \subset \mathbb{A}^n$ is reducible if $V = V_1 \cup V_2$, where V_1, V_2 are closed in V, satisfying $V_1 \neq V$ and $V_2 \neq V$.

Remark 2.1.5. Consider the Zariski topology on $\mathbb{A}^n$. Let $S \subset k[x_1, ..., x_n]$ be a collection of polynomials f_i. Then $Z(S)$ is a closed set and $\mathbb{A}^n - Z(S)$ is an open set. Since $Z(S) = \cap Z(f_i)$,

$$\mathbb{A}^n - Z(S) = \mathbb{A}^n - \cap Z(f_i) = \cup(\mathbb{A}^n - Z(f_i)),$$

which shows that every open set can be written as the union of $(\mathbb{A}^n - Z(f_i))$'s for some f_i. Hence, a base of open sets can be given by these sets. For arbitrary $f, g \neq 0$,

$$(\mathbb{A}^n - Z(f)) \cap (\mathbb{A}^n - Z(g)) = \mathbb{A}^n - Z(fg)$$

which is nonempty because $f, g \neq 0$, and $Z(fg) \neq \mathbb{A}^n$. This shows that every intersection of nonempty open sets is nonempty. Thus, Zariski topology is not Hausdorff.

Proposition 2.1.6. Any nonempty open subset X of an irreducible space Y is irreducible and dense.

Proof: Assume that an open subset X of Y is not dense. Then its closure $\overline{X} = Y_1$ is a closed proper subset in Y. Since X is open, $Y_2 = Y - X$ is a closed proper subset in Y. Y can be expressed as the union $Y = Y_1 \cup Y_2$ of two proper subsets, each one of which is closed in Y. But this contradicts with the irreducibility of Y. Hence, our assumption is wrong and any open subset of an irreducible space is dense. Now, assume that an open subset X of Y is not irreducible. Then, X can be expressed as the union $X = X_1 \cup X_2$ of two proper subsets, each one of which is closed in X. Since X is dense, $\overline{X} = Y = \overline{X_1} \cup \overline{X_2}$. $\overline{X_1} \neq Y$ and $\overline{X_2} \neq Y$, because for example, if $\overline{X_1} = Y$, then $X_1 = X \cap \overline{X_1} = X$ but this is a contradiction. So $Y = \overline{X_1} \cup \overline{X_2}$ and $\overline{X_2}$, $\overline{X_2}$ are proper subsets of Y. Again, this contradicts with the irreducibility of Y. Hence, any open subset of an irreducible space is irreducible. □

Remark 2.1.7. To construct a link between geometry and algebra, we explore the correspondence between ideals and algebraic sets. This link is important because it gives us the opportunity to translate any statement about algebraic sets into a statement about ideals and conversely.

From an algebraic set $X \subset \mathbf{A}^n$, we pass to the ideal of polynomials vanishing at X,

$$I(X) = \{f \in k[x_1, ..., x_n] \mid f(P) = 0 \text{ for all } P \in X\}.$$

If we pass to the zero set of this ideal $Z(I(X))$, then $X \subseteq Z(I(X))$ from the definition of zero set, since every polynomial in the ideal is zero at every point of X. As X is an algebraic set, it can be written as $X = Z(f_1, ..., f_s)$. Then the ideal generated by $f_1, ..., f_s$, which we show by $< f_1, ..., f_s >$, is in $I(X)$. If any polynomial is added to a collection of polynomials, the added polynomial may not vanish at some point where all the other polynomials are zero. This makes the zero set of the new collection of polynomials smaller. Now $I(X)$ contains $< f_1, ..., f_s >$, but may contain some other polynomials as well. Hence $Z(I(X)) \subset Z(< f_1, ..., f_s >) = X$. This shows that $Z(I(X)) = X$.

In general, for an ideal $a \subset A$, $Z(a)$ is its zero set and $I(Z(a))$ is the ideal of the polynomials vanishing at $Z(a)$. $I(Z(a))$ obviously contains I but it may have more elements. Hence $a \subseteq I(Z(a))$.

Example 2.1.8. Consider the ideal in $k[x_1, x_2]$ generated by x_1^2 and x_2^2. $Z((x_1^2, x_2^2))$ is (0,0). But $I((0,0))$ is the ideal generated by x_1 and x_2, and $(x_1, x_2) \supset (x_1^2, x_2^2)$, but $(x_1, x_2) \neq (x_1^2, x_2^2)$.

Definition 2.1.9. Let $a \subset k[x_1, ..., x_n] = A$ be an ideal. *The radical of* a, denoted $\sqrt{a}$, is defined as,

$$\sqrt{a} = \{f \in A \mid f^m \in a \text{ for some integer } m \geq 1\}.$$

An ideal a is a *radical ideal* if it is equal to its radical.

Theorem 2.1.10. (*Hilbert's Nullstellensatz*). Let k be an algebraically closed field, let a be an ideal in $A = k[x_1, ..., x_n]$, and let $f \in A$ be a polynomial which vanishes at all points of $Z(a)$. Then $f^m \in a$ for some integer $r > 0$.

Proof: See [13, p. 374] or [7, pp. 168-173]. □

Now it follows from *Hilbert's Nullstellensatz* that $I(Z(a)) = \sqrt{a}$ for an ideal a. Thus if a is a radical ideal, then $I(Z(a)) = a$.

In this way, we found a correspondence between radical ideals and algebraic sets. This link between algebra and geometry is the main theme.

Irreducibility is not only important geometrically, but it also corresponds to special ideals in the algebraic category.

Proposition 2.1.11. An algebraic set is irreducible if and only if its ideal is a prime ideal.

Proof: First, let us show that if X is irreducible then $I(X)$ is a prime ideal. If $f_1 f_2 \in I(X)$, then $Z(f_1 f_2) \supset Z(I(X))$. We showed in Remark 2.1.7 that $Z(I(X)) = X$, so $X \subseteq Z(f_1 f_2) = Z(f_1) \cup Z(f_2)$. X can be expressed as the union of two closed sets such that $X = (Z(f_1) \cap X) \cup (Z(f_2) \cap X)$. Since X is irreducible, either $X = X \cap Z(f_1)$ or $X = X \cap Z(f_2)$. Hence, $X \subseteq Z(f_1)$ or $X \subseteq Z(f_2)$, that is $f_1 \in I(X)$ or $f_2 \in I(X)$.

Conversely, for a prime ideal p, assume that $Z(p) = X_1 \cup X_2$. Then, $I(Z(p)) = I(X_1 \cup X_2)$. Since p is a prime ideal and since every prime ideal is a radical ideal, $I(Z(p)) = p$. The polynomials that vanish at every point of both X_1 and X_2 form the intersection of the set of polynomials vanishing on X_1 and X_2. Thus we can write $p = I(X_1) \cap I(X_2)$. Then we have either $p = I(X_1)$ or $p = I(X_2)$. Assume this is not true, that is p is a proper subset of $I(X_1)$ and $I(X_2)$. Then there are polynomials satisfying $f \in I(X_1)$, $f \notin I(X_2)$, $g \in I(X_2)$, and $g \notin I(X_1)$. In particular this implies that $f \notin p$ and $g \notin p$. Now, fg is an element of both $I(X_1)$ and $I(X_2)$. So $fg \in p$ and since p is a prime ideal, either f or g must be an element of p. But this is a contradiction, since neither f nor g is in p. So our assumption is false, and hence, $p = I(X_1)$ or $p = I(X_2)$, which shows that $Z(p) = Z(I(X_1) = X_1$ or $Z(p) = Z(I(X_2) = X_2$. So $Z(p)$ is irreducible. □

Since every maximal ideal is prime, it is clear that a maximal ideal m of $A = k[x_1, ..., x_n]$ corresponds to a minimal irreducible closed subset of $\mathbb{A}^n$; which is a point. Then every maximal ideal of A can be expressed as $m = (x_1 - a_1, ..., x_n - a_n)$, for some $a_1, ..., a_n \in k$.

The objects of our category can now be defined as follows.

Definition 2.1.12. An *affine variety* is an irreducible closed subset of $\mathbb{A}^n$. An open subset of an affine variety is called a *quasi-affine variety.* If X is a quasi-affine variety, an irreducible locally closed subset of X is called a *subvariety of X.*

Example 2.1.13. We have seen above that any point of P of $\mathbb{A}^n$ is a minimal irreducible closed subset. Hence, any point P is an example of an affine variety. For any irreducible polynomial $f \in k[x_1, ..., x_n]$, the zero set $Z(f)$ is also an example of affine variety. In $\mathbb{A}^2$, with coordinates x_1 and x_2, $Z(x_1^3 - x_2^2)$ describes the *cuspidal cubic curve.* Similarly in $\mathbb{A}^3$ with coordinates x_1, x_2, and x_3, the zero set $Z(x_2 - x_1^2, x_3 - x_1^3)$ is another affine variety known as the *twisted cubic curve.*

Before we define the morphisms of our category, we define some fundamental concepts.

Definition 2.1.14. The *affine coordinate ring $A(X)$* of a variety $X \subseteq \mathbb{A}^n$ is defined to be $A/I(X)$, that is $k[x_1, ..., x_n]/I(X)$. The elements of the affine coordinate ring are the polynomial functions on our variety. Two polynomials that are equal at each point of the variety are the same function on this variety, and polynomials which are zero at every point of the variety correspond to the zero function on our variety.

A maximal ideal m_P of $A(X)$ for a point $P \in X$ is the set of polynomials vanishing at P, viz. $m_P = \{f \in A(X) \mid f(P) = 0\}$.

The polynomials of $k[x_1, ..., x_n]$ can be considered as functions on $\mathbb{A}^n$. We allow other functions that can be written as the quotient of two polynomials locally, where the denominator polynomial is not zero.

Definition 2.1.15. A function is *regular at a point of a variety X*, if it can be expressed as the quotient of two polynomials on an open neighborhood of the point, where the polynomial in the denominator does not vanish on this neighborhood.

Namely, for a variety X, a function $f : X \to k$ is *regular at a point* $x \in X$ if there is an open neighborhood U with $x \in U \subseteq X$, and polynomials $f_1, f_2 \in k[x_1, ..., x_n]$, satisfying $f = f_1/f_2$ on U and f_2 is nowhere zero on U [10, p. 15].

If f is regular at every point of X, then f is *regular* on X.

Proposition 2.1.16. Functions that are regular at every point of a variety are polynomials. Hence, the ring of functions that are regular at every point of a variety X, denoted by *O(X)*, will be isomorphic to the affine coordinate ring *A(X)* of X.

Proof: See [10. p. 17]. □

We have defined the ring of all regular functions on a variety X. Let us now define the *local ring of P on X*, where $P \in X$.

Definition 2.1.17. The *germ of a regular function on Y near P* is a pair $< U, f >$ where U is an open subset of X containing P, and f is a regular function on U. Two pairs $< U, f >$ and $< V, g >$ are equivalent if $f = g$ on $U \cap V$. The *local ring of P on X*, $O_{P,X}$ is the ring of these germs. This is a local ring and its maximal ideal is the set of germs of regular functions which vanish at P.

Proposition 2.1.18. $O_{P,X}$ is isomorphic to the localization of the affine coordinate ring at its maximal ideal m_P corresponding to P.

Proof: See [10, p. 17]. □

Definition 2.1.19. The *function field K(X)* of X consists of the elements $< U, f >$ where U is a nonempty open subset of X, and f is a regular function on U. $< U, f >$ is equivalent to $< V, g >$ if $f = g$ on $U \cap V$. It is obvious that *K(Y)* is isomorphic to the quotient field of *A(Y)*. The elements of the function field are called *rational functions.*

Finally we can define the morphisms of our category.

We define a morphism between two varieties in such a way that information about regular functions are transferred from one variety to the other, in a manner made precise in the following definition.

Definition 2.1.20. For two varieties X and Y, a *morphism* $\varphi : X \longrightarrow Y$ is defined to be a continuous map such that for every open set $V \subseteq Y$, and for every regular function $f : V \longrightarrow k$, the function $f \circ \varphi : \varphi^{-1}(V) \longrightarrow k$ is regular. In particular, if f is regular on Y, $f \circ \varphi$ is regular on X.

Definition 2.1.21. A *biregular morphism* $\varphi : X \longrightarrow Y$ of two varieties is a morphism which admits an inverse morphism $\psi : Y \longrightarrow X$ with $\psi \circ \varphi = id_X$ and $\varphi \circ \psi = id_Y$. The biregular morphism is the isomorphism in this category.

This completes the definition of the category of varieties and morphisms. This category is known as the category for the *biregular theory.* The main theme in this category is that two objects (varieties) are considered the same if their coordinate rings are isomorphic. However, experience shows that we have to 'enlarge' our category, if we focus our attention on the function fields instead of coordinate rings.

Definition 2.1.22. For two varieties X and Y, a *rational map* $\varphi : X \longrightarrow Y$ is an equivalence class of pairs $< U, \varphi_U >$ where U is a nonempty open subset of X, φ_U is a morphism from U to Y, and $< U, \varphi_U >$ and $< V, \varphi_V >$ are equivalent if $\varphi_U = \varphi_V$ on $U \cap V$. For some $< U, \varphi_U >$, if the image φ_U is dense in Y, then φ is called *dominant.*

Definition 2.1.23. A *birational map* is a rational map that has an inverse, i.e., there exits a rational map $\psi : Y \longrightarrow X$ satisfying $\psi \circ \varphi = id_X$ and $\varphi \circ \psi = id_Y$.

Birational isomorphism is satisfactory and in fact useful because of the following fact.

Proposition 2.1.24. Two varieties are birationally equivalent, if and only if their function fields are isomorphic.

Proof: See [10, p. 26]. □

We will see in section 2.4 that since *blowing up* is a rational map, a variety X and its blowing up is *birationally equivalent*, and their function fields are isomorphic.

The last concept we will mention briefly in this section is the *projective space.*

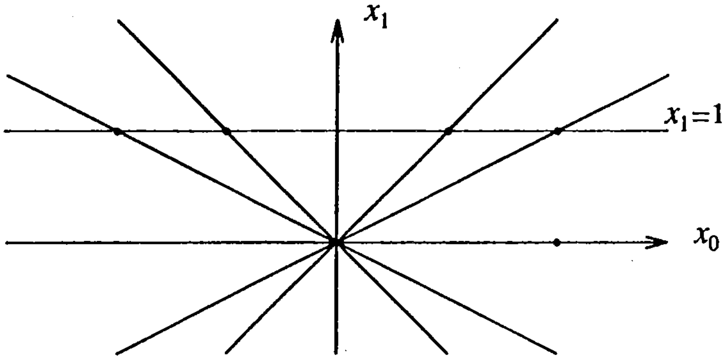

Figure 2.1: $\mathbb{P}^1_{\mathbb{R}}$

Definition 2.1.25. Projective space may be considered as a space whose points are the lines through the origin of some vector space. Hence, formally the projective space $\mathbb{P}^n$ is the quotient of the set $k^{n+1} - \{(0,...,0)\}$ under the equivalence relation given by $(a_0,...,a_n) \sim (\alpha a_0,...,\alpha a_n)$ for all $\alpha \in k, \alpha \neq 0$.

Namely, points lying on the same line through the origin all denote the same point in the projective space. The equivalence class of the point $(a_0,...,a_n)$ in k^{n+1} is denoted by $[a_0 : ... : a_n]$ in $\mathbb{P}^n$.

As an example, consider $\mathbb{P}^1_{\mathbb{R}}$, the set of all lines in $\mathbb{R}^2$ with axes x_0 and x_1. All the lines through the origin except the line $x_1 = 0$ can be parameterized by taking their intersections with the line $x_1 = 1$. With this parameterization, the line $x_1 = 0$ will denote the point at infinity. As a set,

$$\mathbb{P}^1_{\mathbb{R}} = \{(x,1) \mid x \in \mathbb{R}\} \cup \{(1,0)\},$$

where $(1,0)$ is referred as "the point at infinity", see figure 2.1.

In $\mathbb{P}^n_k$ (the set of all lines through the origin in k^{n+1} with axes $x_0,...,x_n$), consider the n-space given by $x_n = 1$. It parameterizes all lines through the origin in k^{n+1} except those that lie in $x_n = 0$. The lines in $x_n = 0$ are the points of $\mathbb{P}^{n-1}$. Hence, $\mathbb{P}^n = \mathbb{A}^n \cup \mathbb{P}^{n-1}$.

We expect $\mathbb{P}^n$ to be locally like $\mathbb{A}^n$ even at infinity, and our expectation is fulfilled when we prove the following.

Proposition 2.1.26. $\mathbb{P}^n$ is a union of $\mathbb{A}^n$'s.

Proof: Some special subsets U_i of $\mathbb{P}^n$ are defined as

$$U_i = \{[x_0 : ... : x_n] \in \mathbb{P}^n \mid x_i \neq 0\}.$$

Hence, if $p = [x_0 : ... : x_n] \in U_i$, then we can write $p = [\frac{x_0}{x_i} : ... : 1 : ... : \frac{x_n}{x_n}]$. We can define a function $\varphi_i : U_i \to k^n$ as

$$\varphi_i([x_0, ..., x_n]) = (\frac{x_0}{x_i}, ..., \frac{x_{i-1}}{x_i}, \frac{x_{i+1}}{x_i}, ..., \frac{x_n}{x_i}).$$

It is clear that φ_i is onto and one to one. It has an inverse

$$\varphi_i^{-1}(a_1, ..., a_n) = [a_1 : ... : a_{i-1} : 1 : a_{i+1} : ... : a_n] \in U_i$$

(Note that φ_i and φ_i^{-1} are acceptable morphisms in our category.)

Thus, U_i is isomorphic to $\mathbb{A}^n$, and $\mathbb{P}^n$ is a union of $\mathbb{A}^n$'s. Hence, locally $\mathbb{P}^n$ looks like $\mathbb{A}^n$. □

Polynomials are not well defined on $\mathbb{P}^n$. However, if f is a homogeneous polynomial and is zero for some $(x_0, ..., x_n) \in k^{n+1} - \{(0, ..., 0)\}$, then f is zero at all points on the line $(\lambda x_0, ..., \lambda x_n)$. Hence, if $P = [x_0 : ... : x_n]$, then we can say that $f(P) = 0$ or $f(P) \neq 0$. Now, we can talk about the zero sets of homogeneous polynomials and define them as algebraic sets. We can define a topology on $\mathbb{P}^n$ by taking the algebraic sets as closed sets. A *projective variety* is then defined to be an irreducible algebraic set in $\mathbb{P}^n$.

Proposition 2.1.27. A projective variety X is covered by the open sets $X \cap U_i$, which will be homeomorphic to affine varieties by the mapping φ_i.

Proof: See [9, p. 5]. □

Example 2.1.28. Consider the projective variety X, which is the zero set of the homogeneous polynomial $x_0 x_2^2 - x_1^3$. x_0 can be taken to be 1 on the intersection of X with the open set U_0. Then the intersection will be the zero set of the cuspidal cubic curve $x_2^2 = x_1^3$.

2.2 Curves

Curves are special varieties. Let us define dimension first in order to define a curve. A *chain of an ideal I* is a sequence of ideals satisfying $I_0 \subset ... \subset I_k = I$ and the *length* of this chain is k. In a ring R, the *height* of a prime ideal p is the maximum n such that there exists a chain $p_0 \subset p_1 \subset \ldots \subset p_n = p$ of distinct prime ideals.

Definition 2.2.1. The *Krull dimension* of the ring R is defined as the maximum of the heights of all prime ideals, i.e., the maximum length of chains of prime ideals in R.

Definition 2.2.2. A variety is called a *curve*, if the Krull dimension of its affine coordinate ring is 1.

If X is an affine variety, then the dimension of X is equal to the dimension of its affine coordinate ring $A(X)$. A variety X in $\mathbb{A}^n$ has dimension $n-1$ if and only if it is the zero set of a single nonconstant irreducible polynomial, see [10, p. 7]. A plane curve is then the set of all points whose coordinates satisfy an equation $f(x,y) = 0$, where f is a polynomial with certain coefficients from the ground field.

Example 2.2.3. The cuspidal cubic which is defined by $x_2^2 - x_1^3 = 0$ is an example of a plane curve in $\mathbb{A}^2$.

Example 2.2.4 The nodal cubic which is defined by $x_2^2 - x_1^2 - x_1^3 = 0$ is another example of a plane curve in $\mathbb{A}^2$.[1]

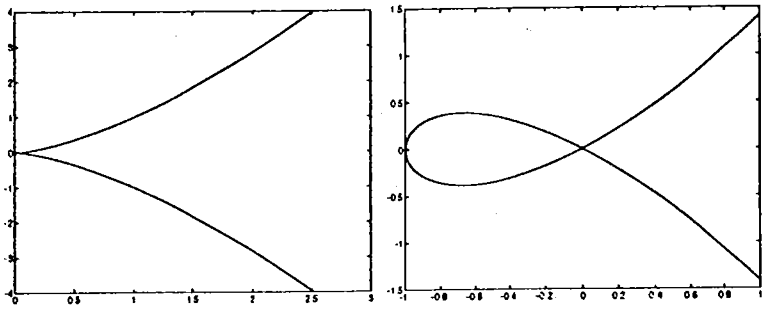

Figure 2.2: The cuspidal cubic curve and the nodal cubic curve

Example 2.2.5. Four-leaved rose which is defined by $(x_1^2+x_2^2)^3 - 4x_1^2x_2^2 = 0$ is another plane curve [7, p. 146].

[1] The cuspidal cubic curve and nodal cubic curve were known by Greeks as the "cissoid of Diocles" and "conchoid of Nicomedes" and used for solving the problem of doubling the cube and trisecting the angle. Diocles showed that $\sqrt[3]{2}$ can be constructed by using ruler, compass and cissoid. These were the classical problems of antiquity. Later, it was proved by Galois theory that they can not be solved by only ruler and compass construction. For more information, see [6, pp. 9-16].

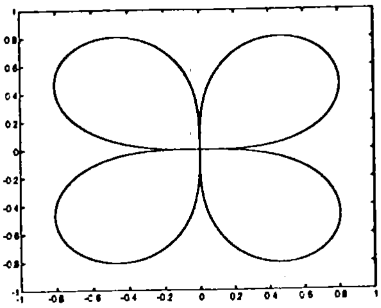

Figure 2.3: Four-leaved rose

Example 2.2.6. The twisted cubic which is defined by $x_2 - x_1^2 = 0$ and $x_3 - x_1^3 = 0$ is a space curve in $\mathbb{A}^3$.

We will deal with the *branch of a curve at a point.* Before giving its definition, we prefer to give an example to be familiar with the notion.

Example 2.2.7. The nodal cubic curve $x_2^2 - x_1^3 - x_1^2 = 0$ has locally two components around (0,0). To see this rewrite the equation of the curve as, $x_2^2 = x_1^2(1 + x_1)$, and observe that,

$$(1 + x_1)^{1/2} = \pm(1 + \tfrac{1}{2}x_1 - \tfrac{1}{8}x_1^2 + ...).$$

This leads to the equations $x_2 = x_1 + \frac{1}{2}x_1^2 - \frac{1}{8}x_1^3 + ...$ and $x_2 = -(x_1 + \frac{1}{2}x_1^2 - \frac{1}{8}x_1^3 + ...)$. They are called the *branches* of the curve at (0,0).

We will give a precise definition of a *branch* by using parameterization.

Let $k[[t]]$ be the formal power series ring. An element φ of $k[[t]]$ is of the form, $\varphi = a_0 + a_1t + ... + a_nt^n + ...$ where $a_i \in k$. *Order* of φ is the degree of the smallest degree term present. Namely, smallest i satisfying $a_i \neq 0$.

Proposition 2.2.8. A variety C of $\mathbb{A}^n$ has a parameterization at any one of its points $(a_1, ..., a_n)$ in the form;

$$\begin{aligned} x_1 &= \varphi_1(t) \\ &\vdots \\ x_n &= \varphi_n(t) \end{aligned}$$

where $\varphi_1(t), ..., \varphi_n(t)$ are power series in t and $(\varphi_1(0), ..., \varphi_n(0)) = (a_1, ..., a_n)$, if and only if C is a curve.

Proof: For the part of the proof beginning with "if C is a curve", see [1, pp. 62-63].

Conversely, let $\alpha : k[x_1, ..., x_n] \to k[\varphi_1(t), ..., \varphi_n(t)] \subset k[[t]]$ be the map such that $\alpha(x_i) = \varphi_i(t)$. It is obvious that that the kernel of this map is *I(X)*. Hence, $k[x_1, ..., x_n]/I(X) \cong k[\varphi_1(t), ..., \varphi_n(t)]$. Since $k[\varphi_1(t), ..., \varphi_n(t)]$ has dimension 1, $k[x_1, ..., x_n]/I(X)$ has dimension 1, too. Hence, X is a curve. □

This parameterization $x_1 = \varphi_1(t), ..., x_2 = \varphi_n(t)$ of a curve C at a point $(a_1, ..., a_n)$ corresponds to a *branch* of C at this point. A parameterization $x_1 = \varphi_1^*(t), ..., x_n = \varphi_n^*(t)$ is *redundant*, if it can be obtained from some other parameterization by substituting for t some power series in t of order > 1. A parameterization is called *irredundant* if it is not redundant.

Definition 2.2.9. A *branch* at a point is an equivalence class of irredundant parameterizations at that point; two parameterizations are equivalent, if one can be obtained from the other by substituting a power series of order 1, see [1, pp. 63-64].

Example 2.2.10. For the nodal cubic curve, by using the equations $x_2 = x_1 + \frac{1}{2}x_1^2 - \frac{1}{8}x_1^3 + ...$ and $x_2 = -(x_1 + \frac{1}{2}x_1^2 - \frac{1}{8}x_1^3 + ...)$ obtained in Example 2.2.7, we will have two parameterizations of the curve at (0,0). The first parameterization at (0,0) is

$$x_1 = t,\ x_2 = t + \tfrac{1}{2}t^2 - \tfrac{1}{8}t^3 + ...,$$

and the second one is

$$x_1 = t,\ x_2 = -(t + \tfrac{1}{2}t^2 - \tfrac{1}{8}t^3 + ...).$$

From the definition, these are the two branches of the curve at (0,0).

At $(-1, 0)$, the same curve has parameterizations,

$$x_1 = t^2 - 1,\ x_2 = t(t^2 - 1) \text{ and } x_1 = t^2 - 1,\ x_2 = -t(t^2 - 1).$$

They correspond to the same branch, because the second parameterization can be obtained from the first one by substituting $-t$ which has order 1. Hence at $(-1, 0)$, there is only one branch of the curve.

The cuspidal cubic curve has one branch at (0,0). It has the parameterization

$$x_1 = t^2, \quad x_2 = t^3$$

The twisted cubic curve has also one branch at (0,0) and it has the parameterization

$$x_1 = t, \quad x_2 = t^2, \quad x_3 = t^3$$

We shall deal with curves which have polynomial parameterizations. These curves can be defined by $x_1 = \varphi_1(t), ..., x_n = \varphi_n(t)$ where $\varphi_1(t), ..., \varphi_n(t)$ are polynomials in t. Not every curve does have a polynomial parameterization. But for every curve that has a polynomial parameterization, it is possible to find the defining equations, i.e., an implicit representation. (There are algorithms for doing this by using *Groebner bases.* For more information, see [7, pp. 126-132].)

2.3 Singularity

In general, singularity within a totality may be defined as a place of uniqueness, of specialty, of degeneration, of indeterminacy or infinity, [6, p. 82]. Smoothness means no sudden and unexpected changes. Singularity may also be defined as the place where smoothness is violated.

For varieties, singularity can be defined both geometrically and algebraically. The geometric definition, which is historically the first, depends on the formal derivatives of the generators defining the ideal of that variety.

Definition 2.3.1. For a variety X of dimension r in A^n, let $f_1, ..., f_k$ be the generators defining its ideal. X is *nonsingular at a point* $P \in X$ if the rank of the matrix $J = (\partial f_i / \partial x_j(P))$ is $n - r$. This matrix is called the *Jacobian matrix* at P.

From this definition, we can deduce that singular points of a curve C are those at which the curve has more than one tangent (counting multiplicity). In other words, a point of a curve is said to be singular if every line through this point has intersection multiplicity greater than 1 with the curve there. *Multiplicity of a curve at a point* P is d, if every line through P has intersection multiplicity at least d with the curve there.

Hence, for a plane curve C of degree n, if the lines through P meet C

outside P in at most $n-d$ points, then multiplicity of P at C is d. By using the Jacobian, a singular point of a plane curve defined by $f(x_1,x_2)=0$ is a point (a,b) with $f(a,b)=0, f_{x_1}(a,b)=0, f_{x_2}(a,b)=0$.

Example 2.3.2. Both the cuspidal cubic curve and the nodal cubic curve defined above has singularity at (0,0). These two examples are highly illustrative for understanding the types of singularities. The cuspidal cubic curve which is defined by $x_2^2-x_1^3=0$ has cusp type of singularity at (0,0), i.e., it has the same tangent with multiplicity greater than 1. The nodal cubic curve which is defined by $x_2^2-x_1^2-x_1^3=0$ has a node at (0,0), where it has two distinct tangents.

All singular curves have either one of these two types of singularities or combinations of them.

Example 2.3.3. The four leaved rose defined by $(x_1^2+x_2^2)^3-4x_1^2x_2^2=0$ is an example having both types of singularities.

By using algebraic concepts, an equivalent definition for singularity can be found.

Definition 2.3.4. A local ring A with maximal ideal m is a *regular local ring* if dim m/m^2 = dim A.

We now give an algebraic definition for singularity;

Definition 2.3.5. Let X be a variety and $P \in X$ be a point. X is singular at P if and only if the local ring $O_{P,X}$ is not a regular local ring.

Proposition 2.3.6. The geometric and algebraic definitions are equivalent. That is, for a variety X of $\mathbb{A}^n$ and $P \in X$, the local ring $O_{P,X}$ is a regular local ring if and only if the rank of the Jacobian matrix at P for the generators of the ideal X is $n-r$ where r is the dimension of X.

Proof. Let $P \in X$ be $(a_1,...,a_n)$. Considering P as a point of $\mathbb{A}^n$, the corresponding maximal ideal is $a_P=(x_1-a_1,...,x_n-a_n)$. A linear map α from $A=k[x_1,...,x_n]$ to k^n is defined by

$$\alpha(f)=(\tfrac{\partial f}{\partial x_1}(P),...,\tfrac{\partial f}{\partial x_n}(P))$$

Since $\alpha(x_i-a_i)=(0,...,1,...,0)$, these form a basis for k^n. Hence, the map $\alpha : a_P \to k^n$ is surjective. Since $\alpha((x_i-a_i)(x_j-a_j))=(0,...,0)$, it is obvious that $\alpha(a_P^2) = 0$. Then, it follows immediately that $\alpha' : a_P/a_P^2 \rightarrow k^n$ is an isomorphism. Let $I(X)$ be the ideal of X such that $I(X) = (f_1, ..., f_m)$. From the definition of Jacobian matrix and α, maximum number of the independent vectors $\alpha(f_i)$ for i from 1 to m is the rank of the Jacobian matrix. Hence, $\alpha(I(X))$ as a subspace of k^n has a dimension equal to the rank of the Jacobian matrix. $\alpha'((I(X) + a_P^2)/a_P^2)$ is equal to $\alpha'(I(X))$, so $(I(X) + a_P^2)/a_P^2$ as a subspace of a_P/a_P^2 has a dimension equal to the rank of the Jacobian matrix, too. Let m_P be the maximal ideal of $O_{P,X}$. Then $m_P/m_P^2 \cong a_P/(I(X) + a_P^2)$. Since $a_P \supset I(X) + a_P^2 \supset a_P^2$, $(a_P/a_P^2)/((a_P + I(X))/a_P^2)$ is isomorphic to $a_P/(I(X) + a_P^2)$, see [13, p. 83]. Hence, dim $a_P/(I(X) + a_P^2)$ + dim $(I(X) + a_P^2)/a_P^2$ = dim $a_P/a_P^2 = n$, and dim m_P/m_P^2 + rank $J = n$. If dim $X = r$, then the local ring $O_{P,X}$ has dimension r, too. Hence, $O_{P,X}$ is regular if and only if dim $m_P/m_P^2 = r$, which is the same thing as saying that the rank of the Jacobian matrix is $n - r$. This shows the equivalence of the algebraic and geometric definitions. □

2.4 Resolution of Singularities

We can classify the singular curves by finding nonsingular curves which are birationally equivalent to these singular curves. This can be done by the resolution of singularities. For curves, the construction of the *blow up* of the curve at a point is the main tool in the resolution of singularities of this curve.

The construction consists of simply removing the singular point and replacing it by a projective line, the points of which will correspond to the tangent directions at that point. Now, let us construct the blowing up of $\mathbb{A}^2$ at the point $O = (0,0)$. The product $\mathbb{A}^2 \times \mathbb{P}^1$ is considered with x_1, x_2 affine coordinates of $\mathbb{A}^2$ and y_1, y_2 homogeneous coordinates of $\mathbb{P}^1$. The closed subsets of $\mathbb{A}^2 \times \mathbb{P}^1$ are defined by the polynomials in x_1, x_2, y_1, y_2 which are homogeneous with respect to y_1, y_2. Now, blowing up of $\mathbb{A}^2$ at the point O is defined to be the closed subset X of $\mathbb{A}^2 \times \mathbb{P}^1$ defined by the equation $x_1y_2 = x_2y_1$.

We have a projection map $\pi : X \rightarrow \mathbb{A}^2$.

$\pi^{-1}(O)$ consists of all points of the form $(0,0) \times [y_1, y_2]$ with $[y_1, y_2] \in \mathbb{P}^1$. Hence, $\pi^{-1}(0)$ is isomorphic to $\mathbb{P}^1$. If we draw a picture of this blow up, it looks like a spiral staircase (Figure 2.4).

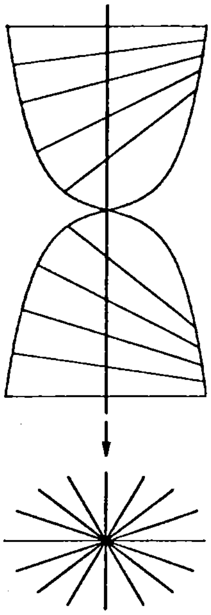

Figure 2.4: The blow up X of $\mathbb{A}^2$

We can generalize this and construct the blowing up of $\mathbb{A}^n$ at the point O=(0,...,0). Now, the product $\mathbb{A}^n \times \mathbb{P}^{n-1}$ is considered with $x_1, ..., x_n$ affine coordinates of $\mathbb{A}^n$ and $y_1, ..., y_n$ homogeneous coordinates of $\mathbb{P}^{n-1}$. The blowing up of $\mathbb{A}^n$ at the point O is defined to be the closed subset X of $\mathbb{A}^n \times \mathbb{P}^{n-1}$ defined by the equations $\{x_iy_j = x_jy_i \mid i, j = 1, ..., n\}$. Again, we have π, the projection map from X to $\mathbb{A}^n$. $\pi^{-1}(O)$ consists of all points $(0, ..., 0) \times [y_1, ..., y_n]$ where $[y_1, ..., y_n] \in \mathbb{P}^{n-1}$. In order to find the blow up at any point, we use change of linear coordinates which sends this arbitrary point to $O = (0, ..., 0)$.

If C is a curve of $\mathbb{A}^n$ passing through O, then blowing up of C at the point O is defined to be the closure $\overline{C}$ of $\pi^{-1}(C - O)$ in $\mathbb{A}^n \times \mathbb{P}^{n-1}$ with respect to Zariski topology where $\pi : X \longrightarrow \mathbb{A}^n$ is the blowing up of $\mathbb{A}^n$ at the point O. The map $\pi : \overline{C} \longrightarrow C$ is a birational morphism.

We will give illustrative examples of blow up.

Example 2.4.1. Let us find the blowing up of the nodal cubic curve $C_1 : x_2^2 = x_1^3 + x_1^2$ at O. The blow up X of $\mathbb{A}^2$ at O is a closed subset of $\mathbb{A}^2 \times \mathbb{P}^1$ satisfying the equation $x_1y_2 = x_2y_1$, where y_1, y_2 are homogeneous coordinates. The inverse image of C_1 in X will be obtained by considering this equation with the curve equation $x_2^2 = x_1^3 + x_1^2$. If the open set of $\mathbb{P}^1$ with $y_1 \neq 0$ is considered, y_1 can be set to be 1. Then, $x_2^2 = x_1^2(x_1 + 1)$ and $x_2 = x_1y_2$. By substituting, we have $x_1^2y_2^2 = x_1^2(x_1 + 1)$, from which we obtain two irreducible components, first of which is $x_1 = 0, x_2 = 0$ and y_2 arbitrary. This is the *exceptional curve* of the blow up. The other is $y_2^2 = x_1 + 1$ and $x_2 = x_1y_2$, which is the closure of $\pi^{-1}(C_1 - O)$ and meets the exceptional curve at $y_2 = \pm 1$, which correspond to the slopes of the two branches of C_1 at O. On the other open set, $y_2 \neq 0$, y_2 may be considered to be 1. Then, $x_2^2 = x_1^2(x_1 + 1)$ and $x_1 = x_2y_1$. By substituting, we have $x_2^2 = x_2^3y_1^3 + x_2^2y_1^2$, from which we obtain two irreducible components, first of which is $x_1 = 0, x_2 = 0$ and y_1 arbitrary. The other is $y_1^2 + y_1^3x_2 = 1$ and $x_1 = x_2y_1$, which meets the exceptional curve at ± 1. In this way, we obtained a nonsingular curve. We observe that the effect of blowing up is to separate out branches of curves passing through the singular point according to their slopes. Hence, it is clear that node type of singularities can be resolved by only one blowing up but for cusp type of singularities, this is not always the case as the following arguments show.

Example 2.4.2. Consider the cuspidal cubic curve $C_2 : x_1^3 = x_2^2$ at O. X, which is the blow up of $\mathbb{A}^2$ at O, is a closed subset of $\mathbb{A}^2 \times \mathbb{P}^1$ satisfying the equation $x_1y_2 = x_2y_1$, where y_1, y_2 are homogeneous coordinates. The inverse image of C_2 in X will be obtained by considering this equation with the curve equation $x_2^2 = x_1^3$. Again if we consider the open set of $\mathbb{P}^1$ with $y_1 \neq 0, y_1$ can be set to be 1. Then, $x_2^2 = x_1^3$ and $x_2 = x_1y_2$. Substitution will give $(x_1y_2)^2 = x_1^3$ from which we can obtain two irreducible components first of which is $x_1 = 0, x_2 = 0$ and y_2 arbitrary. Second one, which is defined by $y_2^2 = x_1$ and $x_2 = x_1y_2$ is the closure of $\pi^{-1}(C_2 - O)$. If the other open set is considered y_2 may be set to 1. Then $x_1 = x_2y_1$. By substituting, we have $x_2^2 = x_2^3y_1^3$. We obtain again two irreducible components. $x_1 = 0, x_2 = 0$ and y_1 arbitrary. The second one is $x_2y_1^3 = 1$ and $x_1 = x_2y_1$. We again obtained a nonsingular curve.

Example 2.4.3. The last example is again a curve which has a cusp type of singularity. It is the curve $C_3 : x_2^2 = x_1^5$. If the same procedure is followed considering the open set $y_1 \neq 0$ of $\mathbb{P}^2$, then the closure of $\pi^{-1}(C_3 - O)$ will be defined by the equations $y_2^2 = x_1^3$ and $y_2x_1 = x_2$. It is obvious that the curve obtained has cusp type of singularity too. Hence, for cusp type of singularities, one blowing up may not be enough to remove the singularity. (In this example another blow up will resolve the singularity.)

It will be useful to write the equations defining the blow up in local coordinates. Recalling that the blow up of X of $\mathbb{A}^n$ at (0,...,0) is,

$$\{((x_1, \ldots, x_n), [y_1 : \ldots : y_n]) \mid x_iy_j - y_ix_j = 0\}.$$

Consider $U_1 = \{y_1 \neq 0\} \subset X$. From the defining equations of the blow up we have, $x_2 = \frac{y_2}{y_1}x_1, ..., x_n = \frac{y_n}{y_1}x_1$. For $x_1 \neq 0, \frac{y_2}{y_1} = \frac{x_2}{x_1}, ..., \frac{y_n}{y_1} = \frac{x_n}{x_1}$. Define a one to one and onto map $\varphi_1 : U_1 \to \mathbf{A}^n$ as,

$$\varphi_1((x_1, ..., x_n) \times [y_1 : ... : y_n]) = (x_1, \frac{y_2}{y_1}, ..., \frac{y_n}{y_1}) = (X_1, ..., X_n)$$

with inverse $\varphi_1^{-1} : \mathbf{A}^n \to U_1$ as,

$$\varphi_1^{-1}(X_1, ..., X_n) = (X_1, X_1X_2, ..., X_1X_n) \times [1 : X_2 : ... : X_n].$$

Hence, the blown up space will be considered as having local coordinates, $X_1 = x_1, X_2 = \frac{y_2}{y_1}, ..., X_n = \frac{y_n}{y_1}$. If $(X_1, ..., X_n) = (0, ..., 0)$ then, $x_1 = y_2 = ... = y_n = 0$, and this corresponds to the point ((0,...,0),[1:0:...:0]) in the blown up space.

This can be done for every open set U_i.

Now, by changing to Euclidean coordinates, let us show what happens to the defining equations of the curves after the blow up in the above examples.

Example 2.4.4. Consider the nodal cubic curve defined by $x_2^2 = x_1^3 + x_1^2$. On the open set U_1, where local coordinates are $X_1 = x_1, X_2 = \frac{x_2}{x_1}$, the blown up curve has the equation $X_1 + 1 = X_2^2$.

Example 2.4.5. For the twisted cubic curve defined by $x_2^2 = x_1^3$, on the open set U_1, where local coordinates are $X_1 = x_1, X_2 = \frac{x_2}{x_1}$, the blown up curve has the equation $X_1 = X_2^2$.

Example 2.4.6. For the last curve defined by $x_2^2 = x_1^5$, on the open set U_1, where local coordinates are $X_1 = x_1, X_2 = \frac{x_2}{x_1}$, the blown up curve has the equation $X_1^3 = X_2^2$. From this equation, the blown up curve is still singular.

Hence, if we have the parameterization of a curve C as

$$x_1 = \varphi_1(t), ..., x_n = \varphi_n(t),$$

then the its blow up on U_1 will have parameterization in local coordinates as

$$X_1 = \varphi_1(t), X_2 = \frac{\varphi_2(t)}{\varphi_1(t)}, ..., X_n = \frac{\varphi_n(t)}{\varphi_1(t)}.$$

In Proposition 2.2.8, we have shown that

$$k[x_1, x_2, ..., x_n]/I(C) \cong k[\varphi_1(t), \varphi_2(t), ..., \varphi_n(t)].$$

Then the local ring of the curve C at $(0,0,...,0)$ is

$$(k[x_1, x_2, ..., x_n]/I(C))_{(x_1,x_2,...,x_n)} \cong k[\varphi_1(t), \varphi_2(t), ..., \varphi_n(t)]_{(t)}.$$

Recall that the completion of a local ring A, denoted by $\hat{A}$, can be defined as the inverse limit $\varprojlim A/m^n$ where m is the maximal ideal of A (see [3, pp. 100-103]). Then the completion of the local ring of the curve at $(0,0,...,0)$ is

$$k[[\varphi_1(t), \varphi_2(t), ..., \varphi_n(t)]].$$

We have seen above that on U_1 the blown up branch curve has parameterization in local coordinates as $X_1 = \varphi_1(t), X_2 = \frac{\varphi_2(t)}{\varphi_1(t)}, ..., X_n = \frac{\varphi_n(t)}{\varphi_1(t)}$. Now, if the singularity is translated to $(0,0,...,0)$, we have in local coordinates

$$X_1' = \varphi_1(t) - c_1, X_2' = \frac{\varphi_2(t)}{\varphi_1(t)} - c_2, ..., X_n' = \frac{\varphi_n(t)}{\varphi_1(t)} - c_n$$

where $c_1, c_2, ..., c_n$ are the constant terms of $X_1, X_2, ..., X_n$. The local ring at $(0,0,...,0)$ is now

$$k[\varphi_1(t) - c_1, \frac{\varphi_2(t)}{\varphi_1(t)} - c_2, ..., \frac{\varphi_n(t)}{\varphi_1(t)} - c_n]_{(t)}.$$

The completion of this local ring is

$$k[[\varphi_1(t) - c_1, \frac{\varphi_2(t)}{\varphi_1(t)} - c_2, ..., \frac{\varphi_n(t)}{\varphi_1(t)} - c_n]].$$

Chapter 3

ARF CHARACTERS OF SINGULAR BRANCHES

In section 2.4, we have seen that for curves which have cusp type singularities, one blow up may not be enough to remove the singularity. Hence, successive resolution processes must be applied. The sequence constructed by taking the multiplicity of the singular point before each successive resolution process is called the *multiplicity sequence.*[*] The problem is to obtain the multiplicity sequence from the local ring of the curve at the singularity, without applying successive blow ups to the curve. In other words, the problem is to predict algebraically the result of a geometric process.

3.1 Historical Development

The multiplicity sequence of a singular plane curve branch can be found by using *Euclidean algorithm*, which is the process of finding the greatest common divisor of two numbers. The branch is characterized by a number of pairs of numbers μ_i, ν_i ($i = 1, ..., k$) such that for $i = 2, .., k$ ν_i is the greatest common divisor of μ_{i-1} and ν_{i-1}, and μ_i is not divisible by ν_i for $i = 1, ..., k$. Euclidean algorithm is applied to these pairs characterizing the branch. The divisors are the multiplicities of the points of the branch and the corresponding quotients are the number of consecutive points of such multiplicity given by the divisor [8, pp. 107-108].

These pairs can be determined from the below parameterization of the plane curve C, known as *Puiseux expansion* [6, pp. 512-518],

$$x_1 = t^u, \quad x_2 = \ldots + b_{m_1}t^{m_1} + \ldots + b_{m_2}t^{m_2} + \ldots + b_{m_3}t^{m_3} + \ldots + b_{m_k}t^{m_k} + \ldots$$

In x_2, m_1 is the smallest exponent not divisible by u whose coefficient b_{m_1} is nonzero, m_2 is the smallest exponent not divisible by greatest common divisor of u and m_1 whose coefficient b_{m_2} is nonzero, m_3 is the smallest exponent not divisible by greatest common divisor of u, m_1, m_2 whose coefficient b_{m_3} is nonzero, ..., and m_k is the smallest integer for which greatest common divisor of u, m_1, m_2, ..., m_k is 1 whose coefficient b_{m_k} is nonzero [1, p. 166]. These terms $b_{m_1}t^{m_1}$, $b_{m_2}t^{m_2}$, ..., $b_{m_k}t^{m_k}$ are the *characteristic terms* of the expansion. Now, we can determine pairs μ_i, ν_i from the degrees of these characteristic terms, and degree of x_1, which is u. First $\mu_1 = m_1$ and $\nu_1 = u$. Then for $i = 2, ..., k$, $\mu_i = m_i - m_{i-1}$ and $\nu_i =$ greatest common divisor of μ_{i-1} and ν_{i-1}.

Example 3.1.1. Consider the branch given by,

$$\begin{array}{rcl} x_1 & = & t^{100} \\ x_2 & = & t^{250} + t^{375} + t^{410} + t^{417} \end{array}$$

in which only characteristic terms appear. By using the method given above, this branch is characterized by the pairs (250,100), (125,50), (35,25) and (7,5). Applying the Euclidean algorithm to these pairs,

$$\begin{array}{rcl}
250 & = & 2 \times \mathbf{100} + 50 \\
100 & = & 2 \times \mathbf{50} \\
\hline
125 & = & 2 \times \mathbf{50} + 25 \\
50 & = & 2 \times \mathbf{25} \\
\hline
35 & = & 1 \times \mathbf{25} + 10 \\
25 & = & 2 \times \mathbf{10} + 5 \\
10 & = & 2 \times \mathbf{5} \\
\hline
7 & = & 1 \times \mathbf{5} \\
5 & = & 2 \times \mathbf{2} + 1 \\
2 & = & 2 \times \mathbf{1}
\end{array}$$

The numbers in boldface are the multiplicities, and the numbers multiplied by them in the table denote the number of times each multiplicity is repeated in the sequence. Then the multiplicity sequence of the curve branch is:

(100,100 50,50,50,50 25,25,25 10,10 5,5,5 2,2 1,1).

As expected, for a general curve in n-space, this method does not work. In fact the theory of algebraic curve branches in plane was complete even in 1938, when Semple wrote "Singularities of Space Algebraic Curves", see [17]. In his paper, Semple analyzed the geometry of successive resolution processes on a singular curve branch in 3-space. Semple defined *proximity* in order to determine a relation between the successive blow ups. A plane $e_1^{(1)}$ is obtained when a point E_1 of a space curve branch is resolved. Any point E_2 of $e_1^{(1)}$ is defined to be proximate to E_1. By resolving E_2 again, a plane $e_2^{(2)}$ is obtained. Also, a surface $e_1^{(2)}$ represents $E_1 - E_2$. These two, $e_1^{(2)}$ and $e_2^{(2)}$, meet in a line. Then from the point of view of linear equivalence, $E_1 \equiv e_1^{(2)} + e_2^{(2)}$. If E_3 is chosen freely in $e_2^{(2)}$, it will be proximate only to its immediate ancestor E_2, but if it is chosen in the intersection line, then it will be proximate to both E_1 and E_2. Suppose, E_3 is chosen from the intersection line and resolved. Then, a plane $e_3^{(3)}$ representing E_3, a surface $e_2^{(3)}$ representing $E_2 - E_3$, and a surface $e_1^{(3)}$ representing $E_1 - E_2 - E_3$. These intersect in curves, which are concurrent in a point. E_4 can be chosen freely in $e_3^{(3)}$ proximate only to E_3, or in the intersection of $e_3^{(3)}$ and $e_1^{(3)}$, proximate to E_1 and E_3, or in the intersection of $e_3^{(3)}$ and $e_2^{(3)}$, proximate to E_2 and E_3, or as the point where the curves are concurrent, proximate to E_1, E_2, and E_3.

We can summarize these by explaining the set up after every blow up:

(0) We have a point E_1 (which is to be resolved).

(1) A plane $e_1^{(1)}$ is obtained. A point E_2 of $e_1^{(1)}$ is chosen. We have now E_2 and $e_1^{(1)} - E_2 \equiv E_1 - E_2$. ($E_2$ is to be resolved.)

(2) We obtain a plane $e_2^{(2)}$ representing E_2, and a surface $e_1^{(2)}$ representing $E_1 - E_2$. These intersect in a line. E_3 is chosen on the line. Now, we have E_3, $e_2^{(2)} - E_3 \equiv E_2 - E_3$ and $e_1^{(2)} - E_3 \equiv E_1 - E_2 - E_3$. ($E_3$ is to be resolved.)

(3) We obtain a plane $e_3^{(3)}$ representing E_3, a surface $e_2^{(3)}$ representing $E_2 - E_3$, and a surface $e_1^{(3)}$ representing $E_1 - E_2 - E_3$. These intersect in curves concurrent in a point.

By generalizing this, Semple defined a point E_j as proximate to E_i, if E_j is chosen from the resolution of E_i or if E_j is any point of any *diminished neighborhood of* E_i, which is obtained by subtracting from it a set of points of itself. As above, $E_1 - E_2 - E_3$ is a diminished neighborhood of E_1. Hence, E_4 which is chosen from this neighborhood is proximate to E_1. $E_2 - E_3$ is a diminished neighborhood of E_2, and E_4 chosen from this neighborhood is proximate to E_2.

From this definition, Du Val in 1942 deduced the following three results for the n-space [8, p. 109].

(1) If a point is proximate to two others, then one of these is proximate to the other.

(2) In n dimensions, no point can be proximate to more than n others.

(3) The multiplicity of a curve in any point is the sum of its multiplicities in points proximate to that.

Du Val defined the *restriction of a point* to be the number of its predecessors to which it is proximate. This led to the definition of a *leading point of the branch* as one whose restriction is less than that of its successor. The sum of the multiplicities of the first n-points is called the *multiplicity sum corresponding to the n-th point.* Du Val called the multiplicity sum corresponding to a leading point as a *character of the branch.* His main contribution to the problem of finding the multiplicity sequence of an arbitrary curve branch is that, if the characters of the branch are known, then the multiplicity sequence of the branch can be found by applying the *modified Jacobian algorithm* to these characters.

The modified Jacobian algorithm [8, p. 108-109] is as follows:

We begin with the characters of an algebraic branch. They form the first row. The least of these is chosen as the divisor. The least of the remaining ones is chosen and divided by the divisor. The product (quotient × divisor for this least element) is subtracted from all the numbers except the divisor itself. This algorithm differs from the Jacobian algorithm by the fact that in Jacobian algorithm all the numbers in the row except the divisor are divided by the divisor so that all the remainders are less than the divisor. In modified Jacobian algorithm the remainders may be greater than the divisor. The divisor and the remainders obtained form the second row. If any remainder is zero, it is omitted. The algorithm stops, when we reach a row consisting of only one element. The divisors are the multiplicities of the points of the branch, and the quotient corresponding to each divisor is the number of points of the corresponding multiplicity.

Example 3.1.2. We will apply the modified Jacobian algorithm to two algebraic curve branches. The first one has the characters 100, 250, 425, 485, 512. The second one is an example of Du Val which has the characters 2087, 4610, 6068, 6384.

100	250	425	485	512	2	**2087**	4610	6068	6384	2
	$\underline{200}$	$\underline{200}$	$\underline{200}$	$\underline{200}$			$\underline{4174}$	$\underline{4174}$	$\underline{4174}$	
100	**50**	225	285	312	2	2087	**436**	1894	2210	4
$\underline{\underline{100}}$		$\underline{100}$	$\underline{100}$	$\underline{100}$		$\underline{1744}$		$\underline{1744}$	$\underline{1744}$	
	50	125	185	212	2	343	436	**150**	466	2
		$\underline{100}$	$\underline{100}$	$\underline{100}$		$\underline{300}$	$\underline{300}$		$\underline{300}$	
	50	**25**	85	112	2	**43**	136	150	166	3
	$\underline{\underline{50}}$		$\underline{50}$	$\underline{50}$			$\underline{129}$	$\underline{129}$	$\underline{129}$	
		25	35	62	1	43	**7**	21	37	3
			$\underline{25}$	$\underline{25}$		$\underline{21}$		$\underline{\underline{21}}$	$\underline{21}$	
		25	**10**	37	2	22	**7**		16	2
		$\underline{20}$		$\underline{20}$		$\underline{14}$			$\underline{14}$	
		5	10	17	2	8	7		**2**	3
			$\underline{\underline{10}}$	$\underline{10}$		$\underline{6}$	$\underline{\underline{6}}$			
		5		7	1	2	**1**		2	2
				$\underline{5}$		$\underline{2}$			$\underline{2}$	
		5		**2**	2		1			
		$\underline{4}$								
		1		2	2					
				$\underline{\underline{2}}$						
		1								

The multiplicity sequence of the curve branch, which has the characters 100, 250, 425, 485, 512 is:

(100,100 50,50,50,50 25,25,25 10,10 5,5,5 2,2 1,1).

The multiplicity sequence of the curve branch, which has the characters 2087, 4610, 6068, 6384 is:

(2087,2087 436,436,436,436 150,150 43,43,43 7,7,7,7,7 2,2,2 1,1)

Every time a column enters the algorithm, a leading point is passed so restriction rises by unity. Every time a column becomes zero, restriction falls by unity. We can observe from the table that, for the first curve branch, the restriction can not be greater than 2. Thus, the branch is capable of existing in two dimensions. This is the expected result because the example given previously as a plane algebraic curve has the same multiplicity sequence. For the second curve, the restriction rises to 3. Hence, the branch is capable of existing in three dimensions, but not in a plane.

This method is valid for any algebraic curve branch with known characters. More than this, it gives us the opportunity to know the minimum dimension in which the branch is capable of existing. What Du Val asked was, how we should obtain the characters of the curve branch. The relation between these and the expansion of branch in power series was unsolved, until Arf wrote his paper in 1949 [2].

3.2 Arf Rings, Closure and Characters

Arf's work depends on the observation that there is an algebra behind these geometric arguments. He shows that the multiplicity sequence of an algebraic branch can be defined by purely algebraic notions. He constructs the link between geometry and algebra by showing that the *completion* of the local ring at the singularity of the branch carries all the information necessary to obtain the multiplicity sequence.

For a curve branch having the parameterization $x_1 = \varphi_1(t), ..., x_n = \varphi_n(t)$, the completion of the local ring is $k[[\varphi_1(t), ..., \varphi_n(t)]]$, i.e., the ring formed by all formal series of the form,

$$\sum_{i_1,...,i_n \geq 0} \alpha_{i_1,...,i_n} \varphi_1^{i_1} \cdots \varphi_n^{i_n}$$

where $\alpha_{i_1,...,i_n} \in k$.

Arf passes from geometry to algebra by using the completion of the local ring. He constructs the *canonical closure* of this ring, later called the *Arf closure*. The orders of the elements of the constructed ring form a semigroup. This is passing from algebra to arithmetic. Now from this semigroup, Arf obtains some numbers by a process to be described below, and then applies the modified Jacobian algorithm to these numbers to obtain the multiplicity sequence.

In this way, he provides an answer to the question of finding the relation that must exist between Du Val's results and the series representation of the branch.

Before we define *Arf ring, Arf semigroup, Arf closure*, and *Arf characters*, we give necessary definitions and theorems. For the proofs of theorems, see [2].[1]

In the formal power series ring $k[[t]]$, where k is any field, consider a subring H. Let

$$W(H) = \{i_0 = 0, i_1, i_2, ..., i_r, i_{r+1}, ...\}$$

be the orders of the elements of H in increasing order. The integers $i_0, i_1, ..., i_r, ...$ form a semi-group of nonnegative integers. If arbitrary $S_0, S_{i_1}, S_{i_2}, ...$ elements of orders $i_0, i_1, i_2, ...$ respectively are chosen from H, then every element of H can be written in the form

$$\sum_{l=0}^{\infty} \alpha_l S_{i_l}, \qquad (\alpha_l \in k).$$

H is assumed to contain all the power series of this form. The subset of H consisting of all the elements whose orders are greater than or equal to h will be denoted by I_h:

$$I_h = \{f \in H \mid ord(f) \geq h\}.$$

I_h is an ideal of H and its elements are of the form

$$\sum_{i_l \geq h} \alpha_l S_{i_l}, \qquad (\alpha_l \in k).$$

Theorem 3.2.1. [2, Aux. Thm. 1, p. 257] If ν is the greatest common divisor of the elements of $W(H)$, then for sufficiently large r,

$$i_{r+1} = i_r + \nu, \quad i_{r+2} = i_r + 2\nu, \quad ..., \quad i_{r+k} = i_r + k\nu, ...$$

and there exists a power series of order 1,

[1] In his paper, Arf uses the names canonical ring for Arf ring, canonical semigroup for Arf semigroup, and canonical closure for Arf closure. We use "Arf" rather than "canonical".

$$\tau = t(1 + \sum_{l=1}^{\infty} \delta_l t^l) \qquad (\delta_l \in k).$$

such that every element of H is of the form $\sum_{j=0}^{\infty} \alpha_j \tau^{j\nu}$.

From the theorem, it follows that if the greatest common divisor of the elements of $W(H)$ is 1, then H contains power series of all orders after some r. Note that in this case, i.e., when $\nu = 1$, finding the largest integer not contained in $W(H)$ is known as the *problem of Frobenius*, which we will discuss in chapter 4.

Theorem 3.2.2. [2, Aux. Thm. 3, p. 259] If the set of quotients of elements of I_h by S_h (an element of order h in H) is denoted by I_h/S_h, and the ring generated in $k[[t]]$ by I_h/S_h is denoted by $[I_h/S_h]$, then the ring $[I_h/S_h]$ does not depend on the choice of S_h.

Thus, we will denote $[I_h/S_h]$ by $[I_h]$.

Example 3.2.3. [2, p. 260] This illustrative example of Arf shows that the semigroup $W([I_{i_h}])$ contains the semigroup generated by the integers

$$i_h - i_h = 0, \quad i_{h+1} - i_h, \quad i_{h+2} - i_h, \ldots$$

which are the orders of the elements of I_{i_h}/S_{i_h}. But $W([I_{i_h}])$ is not necessarily equal to this semigroup. Consider the ring H formed by all the series of the form

$$\sum_{i,j \geq 0} \alpha_{ij} X^i Y^j \qquad (\alpha_{ij} \in k),$$

where $X = t^4, \quad Y = t^{10} + t^{15}$. $W(H)$ contains the integers

0,4,8,10,12,14,16,18,20,22,24,25,26,28,29,30,32,33,34,35,36,37,38,...

Hence, the orders of the elements of I_4/X are

0,4,6,8,10,12,14,16,18,20,21,22,24,25,26,28,29,30,31,32,33,34,...

which generated the semigroup

$$0,4,6,8,10,12,14,16,18,20,21,22,24,25,26,27,28,29,30,31,32,33,34,...$$

but $[I_4]$ contains the element $(Y/X)^2 - X^3 = 2t^{17} + t^{22}$ whose order is 17, which is not an element of this set.

Definition 3.2.4. [2, p. 260] A ring is called an *Arf ring* if $[I_h] = I_h/S_h$ for every $h \in W(H)$.

Remark 3.2.5. [2, p. 260] If H is an Arf ring, then the integers

$$i_h - i_h = 0, \quad i_{h+1} - i_h, \quad i_{h+2} - i_h, \quad ...$$

form a semigroup for every h. A semigroup of nonnegative integers

$$i_0 = 0, \quad i_1, \quad i_2, \quad ..., \quad i_h, \quad ...$$

is called an *Arf semigroup* if the sequence of integers

$$i_h - i_h = 0, \quad i_{h+1} - i_h, \quad i_{h+2} - i_h, \quad ...$$

is a semigroup for every h.

From the definition, $k[[t]]$ is an Arf ring, and $\mathbb{N}$ is an Arf semigroup.

Theorem 3.2.6. [2, Aux. Thm. 4, p. 261] *(i)* Intersection of Arf rings is an Arf ring.

(ii) Intersection of Arf semigroups is an Arf semigroup.

Remark 3.2.7. [2, p. 261] If H is an Arf ring, then so is $[I_{i_h}]$.

Definition 3.2.8. [2, p. 263] Given a ring H, the intersection of all Arf rings containing H is an Arf ring $^*H$, which is called the *Arf closure of H*. Similarly for a semigroup G, the intersection of all semigroups containing G is an Arf semigroup $^*G$, called the *Arf closure of G*.

Theorem 3.2.9. [2, Thm. 1, p. 264] The intersection of all the semigroups g such that $^*g =^* G$ is a semigroup g_χ such that $^*g_\chi =^* G$.

Definition 3.2.10. [2, p. 265] The sub-semigroup g_χ defined in the above theorem is called the *characteristic sub-semigroup* of all g such that ${}^*g =^* G$.

If g is a semigroup, then there is a minimal set of generators, $\chi_1, \chi_2, ..., \chi_h$ such that

$$g = \{\alpha_1\chi_1 + ... + \alpha_h\chi_h \mid \alpha_1, ..., \alpha_h \text{ nonnegative integers}\}.$$

They are defined in the following way: χ_1 is the smallest nonzero integer in g. χ_2 is the smallest of the integers of g which is not of the form $\alpha_1\chi_1$, where α_1 is a nonnegative integer. χ_3 is the smallest of the integers of g which are not of the form $\alpha_1\chi_1 + \alpha_2\chi_2$, where α_1, α_2 are nonnegative integers. $\chi_1, \chi_2, ..., \chi_n$ being defined, χ_{n+1} is the smallest integer in g which is not of the form

$$\alpha_1\chi_1 + \alpha_2\chi_2 + ... + \alpha_n\chi_n$$

where $\alpha_1, \alpha_2, ..., \alpha_n$ are nonnegative integers, i.e., $\chi_1, ..., \chi_n$ is a minimal set of generators of g over $\mathbb{N}$.

Definition 3.2.11. [2, p. 265] g_χ being the characteristic sub-semigroup of g, the minimal set of generators $\chi_1, \chi_2, ..., \chi_h$ of g_χ defined in this way are the *characters of g*.

Theorem 3.2.12. [2, Thm. 2, p. 265] $\gamma_1 < \gamma_2 < ... < \gamma_l$ being the generators of a semigroup g, the set of the characters of g is contained in the set $\{\gamma_1, \gamma_2, ..., \gamma_l\}$.

Theorem 3.2.13. [2, Thm. 3, p. 266] g being the semigroup generated by $0 < \gamma_1 < \gamma_2 < ... < \gamma_l$ over natural numbers, the integers

$$\nu_1, \nu_2, ..., \nu_{N-1}, \nu$$

such that

$${}^*g = \{0, \nu_1, \nu_1 + \nu_2, ..., \nu_1 + \nu_2 + ... + \nu_{N-1} + \mathbb{N}\nu\}$$

are obtained from $0 < \gamma_1 < \gamma_2 < ... < \gamma_l$ by the modified Jacobian algorithm of Du Val. There the integers $\nu_1, \nu_2, ..., \nu_{N-1}, \nu$ appear as divisors, while the quotients represent the number of times each divisor is repeated in the sequence

$\nu_1, \nu_2, \dots, \nu_{N-1}, \nu$. Conversely, if the numbers $\nu_1, \nu_2, \dots, \nu_{N-1}, \nu$ are obtained from $\gamma_1, \gamma_2, \dots, \gamma_l$ by the modified Jacobian algorithm, the quotients being the number of times each divisor appears in the sequence $\nu_1, \nu_2, \dots, \nu_{N-1}, \nu$, then the Arf closure of g is,

$$^*g = \{0, \nu_1, \nu_1 + \nu_2, \dots, \nu_1 + \nu_2 + \dots + \nu_{N-1} + \mathbb{N}\nu\}.$$

This theorem gives us an algorithm for finding the Arf closure of a semigroup g generated by $\gamma_1, \gamma_2, \dots, \gamma_l$:

(1) Apply the modified Jacobian algorithm to $\gamma_1, \gamma_2, \dots, \gamma_l$.

(2) Obtain $\nu_1, \nu_2, \dots, \nu_{N-1}, \nu$ which are the divisors, where the quotients represent the number of times each divisor is repeated in the sequence $\nu_1, \nu_2, \dots, \nu_{N-1}, \nu$.

(3) $^*g = \{0, \nu_1, \nu_1 + \nu_2, \dots, \nu_1 + \nu_2 + \dots + \nu_{N-1} + \mathbb{N}\nu\}$.

In chapter 5, given $\gamma_1 < \gamma_2 < \dots < \gamma_l$, the generators of a semigroup g, we will present an algorithm for finding the characters of g.

Let us consider now a subring H of $k[[t]]$, and its Arf closure $^*H$. If the ring is of the form,

$$H = k + kS_{i_1} + kS_{i_2} + \dots + k[[T]]S_{i_h}, \qquad (T = \tau^\nu \text{ as explained above.})$$

then its Arf closure can be constructed in the following way [2, p. 267].

Let H_1 denote the ring

$$[I_{i_1}] = \sum k(\tfrac{S_{i_2}}{S_{i_1}})^{\alpha_2}(\tfrac{S_{i_3}}{S_{i_1}})^{\alpha_3}\dots(\tfrac{S_{i_{h-1}}}{S_{i_1}})^{\alpha_{h-1}} + k[[T]](\tfrac{S_{i_h}}{S_{i_1}}),$$

where the summation is taken over exponent systems of nonnegative integers $\alpha_2, \alpha_3, \dots, \alpha_{h-1}$ such that $\alpha_2(i_2 - i_1) + \alpha_3(i_3 - i_1) + \dots + \alpha_{h-1}(i_{h-1} - i_1)$ is less than $i_h - i_1$. This is not a necessary condition, but only the power series, whose orders are less than $i_h - i_1$ can be generated, when this restriction is obeyed. Hence, it prevents overlapping.

The Arf closure $^*H$ of H clearly contains $k + H_1 S_{i_1}$, which in turn contains H. Hence, $^*H = k +^* H_1 S_{i_1}$ from the definition of Arf closure. The ring H_{i+1} is derived from H_i in the same manner H_1 is derived from H. Then for sufficiently large N, H_N is $k[[T]]$. Let T_{i+1} be an element of least positive order in H_i, and T_1 be an element of least positive order in H. Then, we have,

$$\begin{aligned}
{}^*H &= k + kT_1 + {}^*H_2T_1T_2 \\
&= k + kT_1 + kT_1T_2 + {}^*H_3T_1T_2T_3 \\
\ldots & \ \ldots\ldots\ldots\ldots \\
&= k + kT_1 + kT_1T_2 + \ldots + kT_1T_2\ldots T_{N-1} + k[[T]]T_1T_2\ldots T_{N-1}T_N
\end{aligned}$$

Recalling that for a curve branch having the parameterization $x_1 = \varphi_1(t), \ldots, x_n = \varphi_n(t)$, the completion of the local ring is $H = k[[\varphi_1(t), \ldots, \varphi_n(t)]]$. Namely, H is a subring of the formal power series $k[[t]]$, which consists of series of the form

$$\sum_{a_1,\ldots,a_n \geq 0} \alpha_{a_1,\ldots,a_n}\varphi_1^{a_1}\cdots\varphi_n^{a_n}$$

where $\alpha_{a_1,\ldots,a_n} \in k$. Let order of $\varphi_1 = i_1$, order of $\varphi_2 = i_2$, ..., order of $\varphi_n = i_n$, and let i_1 be the smallest of the integers $i_1, i_2, \ldots, i_n$. After the first blow up, as we have explained in section 2.4, we have $X_1 = \varphi_1(t), X_2 = \frac{\varphi_2(t)}{\varphi_1(t)}, \ldots, X_n = \frac{\varphi_n(t)}{\varphi_1(t)}$ in local coordinates. By translation, the singular point of the blown up branch is brought to the origin such that we now have parameterization polynomials, $X_1' = \varphi_1(t) - c_1, X_2' = \frac{\varphi_2(t)}{\varphi_1(t)} - c_2, \ldots, X_n' = \frac{\varphi_n(t)}{\varphi_1(t)} - c_n$ where $c_1, c_2, \ldots, c_n$ are the constant terms of $X_1, X_2, \ldots, X_n$. Here, we make an important observation,

$$H_1 = [I_{i_1}] = k[[X_1', X_2', \ldots, X_n']],$$

where $H_1 = [I_{i_1}]$ is the ring defined above and i_1 is the order of φ_1. H_2 can be defined in the same way from H_1. By this observation, it is easier to construct ${}^*H$.

In the following examples, we will construct Arf closures of two curve branches in both ways.

Example 3.2.14. Let H be a subring of the formal power series $k[[t]]$, which consists of the series of the form $\sum \alpha X^iY^jZ^k$, such that $X = t^4, Y = t^9, Z = t^{15}$ and α is an element of the field k and i, j, k are natural numbers. Then, $W(H) = \{0, 4, 8, 9, 12, 13, 15 + \mathbb{N}\}$. Hence, H can be given as,

$$H = k + kt^4 + kt^8 + kt^9 + kt^{12} + kt^{13} + k[[t]]t^{15},$$

and

$$H_1 = [I_4] = \sum k(\tfrac{t^8}{t^4})^{\alpha_2}(\tfrac{t^9}{t^4})^{\alpha_3}(\tfrac{t^{12}}{t^4})^{\alpha_4}(\tfrac{t^{13}}{t^4})^{\alpha_5} + k[[t]](\tfrac{t^{15}}{t^4}).$$

Hence, H_1 can be given as

$$H_1 = k + kt^4 + kt^5 + k[[t]]t^8.$$

In fact by using the observation we mentioned above, we immediately have

$$H_1 = [[t^4, \tfrac{t^9}{t^4}, \tfrac{t^{15}}{t^4}]] = [[t^4, t^5, t^{11}]].$$

Now, we apply the same procedure to H_1, and obtain H_2.

$$H_2 = \sum k(\tfrac{t^5}{t^4})^{\alpha_2} + k[[t]](\tfrac{t^8}{t^4}) = k[[t]] = {}^*H_2$$

Also, we can have it from $H_1 = [[t^4, t^5, t^{11}]]$, as

$$H_2 = k[[t^4, \tfrac{t^5}{t^4}, \tfrac{t^{11}}{t^4}]] = k[[t]].$$

Hence, Arf closure of H can be written as,

$${}^*H = k + kt^4 + k[[t]]t^8$$

Example 3.2.15. Now, we will calculate the Arf closure of the subring of k[[t]], which consists of the series of the form $\sum \alpha X^i Y^j Z^k$, such that $X = t^6, Y = t^8 + t^{11}, Z = t^{14}$ and α is an element of the field k and i, j, k are natural numbers. By direct calculation, we can see that $W(H)$ will contain 17 and 27, because $XY - Z = t^{17}$ and $X^4 - Y^3 = 3t^{27} + 3t^{30} + t^{33}$. Hence, it will be seen that $W(H)$={0,6,8,10,12,14,16,17,18,20,22+ℕ}, and H and H_1 can be given as

$$\begin{aligned} H &= k + kt^6 + k(t^8 + t^{11}) + kt^{12} + kt^{14} + k(t^{16} + 2t^{19} + t^{22}) + \\ &\quad kt^{17} + kt^{18} + kt^{20} + k[[t]](t^{22} + t^{25}) \\ H_1 &= [I_6] = \sum k(\tfrac{t^8+t^{11}}{t^6})^{\alpha_2}(\tfrac{t^{12}}{t^6})^{\alpha_3}(\tfrac{t^{14}}{t^6})^{\alpha_4}(\tfrac{t^{16}+2t^{19}+t^{22}}{t^6})^{\alpha_5} \\ &\quad (\tfrac{t^{17}}{t^6})^{\alpha_6}(\tfrac{t^{18}}{t^6})^{\alpha_7}(\tfrac{t^{20}}{t^6})^{\alpha_8} + k[[t]](\tfrac{t^{22}+t^{25}}{t^6}) \end{aligned}$$

This ring will have the element $(t^2 + t^5)^3 - t^6 = 3t^9 + 3t^{12} + t^{15}$ so by direct calculation, we see that it will have all the power series of order greater than and equal to 8. Hence, it can be given as

$$H_1 = k + k(t^2 + t^5) + k(t^4 + 2t^7 + t^{10}) + kt^6 + k[[t]]t^8$$

Now, we apply the same procedure to H_1.

$$H_2 = \sum k(\tfrac{t^4+2t^7+t^{10}}{t^2+t^5})^{\alpha_2}(\tfrac{t^6}{t^2+t^5})^{\alpha_3} + k[T](\tfrac{t^8}{t^2+t^5})$$

$$H_2 = \sum k(t^2 + t^5)^{\alpha_2}(t^4 - t^7 + t^{10} + ...)^{\alpha_3} + k[[t]](t^6 - t^9 + t^{12} + ...)$$

Hence, it is possible to write H_2 as

$$H_2 = k + k(t^2 + t^5) + k(t^4 - t^7 + t^{10} + ...) + k[[t]](t^6 - t^9 + t^{12} + ...)$$

From this we obtain

$$\begin{aligned} H_3 &= \sum k(\tfrac{t^4-t^7+t^{10}+...}{t^2+t^5})^{\alpha_2} + k[[t]](\tfrac{t^6-t^9+t^{12}+...}{t^2+t^5}) \\ &= k + k(t^2 + t^5) + (t^4 + ...) + k[[t]](t^5 + ...) \end{aligned}$$

Continuing the procedure in the same manner:

$$\begin{aligned} H_4 &= k + k(t^2 + t^5) + k(t^3 + ...) + k[[t]](t^4 + ...) \\ H_5 &= k[[t]]. \end{aligned}$$

This shows that ${}^*H_5 = k[[t]]$. Hence,

$$\begin{aligned} {}^*H &= k + kt^6 + kt^6(t^2 + t^5) + kt^6(t^2 + t^5)^2 kt^6(t^2 + t^5)^3 + \\ & k[[t]](t^6(t^2 + t^5)^4)) \end{aligned}$$

Now, by using the observation mentioned above, let us construct the Arf closure of the same ring, $H = k[[t^6, t^8 + t^{11}, t^{14}]]$.

$$\begin{aligned} H &= k + kt^6 + k(t^8 + t^{11}) + kt^{12} + kt^{14} + k(t^{16} + 2t^{19} + t^{22}) + \\ & kt^{17} + kt^{18} + kt^{20} + k[[t]](t^{22} + t^{25}). \end{aligned}$$

We start by finding $[I_6] = H_1$.

$$H_1 = k[[t^6, \tfrac{t^8+t^{11}}{t^6}, \tfrac{t^{14}}{t^6}]] = k[[t^6, t^2 + t^5, t^8]].$$

Among the elements of H_1, $t^2 + t^5$ has the smallest order. We divide t^6 and t^8 by $t^2 + t^5$. Now, we have $t^2 + t^5, (t^4 - t^7 + t^{10} - t^{13} + t^{16} - t^{19} + ...), (t^6 - t^9 + t^{12} - t^{15} + t^{18} - t^{21} + ...)$ which yield

$$H_2 = k[[t^2 + t^5, t^4 - t^7 + t^{10} - t^{13} + ..., t^6 - t^9 + t^{12} - t^{15} + ...]].$$

We took the terms having degree less than $22 - 6 = 16$, because we know that H contains power series of all orders greater than or equal to 22, and $H_1 = [I_6]$. Hence, H_1 contains power series of all orders after 16. In H_1, $t^2 + t^5$ has the smallest order. Thus,

$$\begin{aligned} H_3 &= k[[t^2 + t^5, \tfrac{t^4 - t^7 + t^{10} - t^{13} + ...}{t^2 + t^5}, \tfrac{t^6 - t^9 + t^{12} - t^{15} + ...}{t^2 + t^5}]] \\ &= k[[t^2 + t^5, t^2 - 2t^5 + 3t^8 - 4t^{11} + ..., t^4 - 2t^7 + 3t^{10} - 4t^{13} + ...]]. \end{aligned}$$

Again, $t^2 + t^5$ has the least order, and we divide the other two generators by $t^2 + t^5$. We have then $(1 - 3t^3 + 6t^6 - 10t^9 + ...)$ and $(t^2 - 3t^5 + 6t^8 - 10t^{11} + ...)$. We subtract the constant term from the first polynomial, and construct H_4 as

$$H_4 = k[[t^2 + t^5, t^2 - 3t^5 + ..., t^3 - 2t^6 + ...]].$$

It is clear that $H_5 = k[[t]]$. Hence, we have

$$\begin{aligned} {}^*H &= k + kt^6 + kt^6(t^2 + t^5) + kt^6(t^2 + t^5)^2 kt^6(t^2 + t^5)^3 + \\ & \quad k[[t]](t^6(t^2 + t^5)^4)). \end{aligned}$$

Theorem 3.2.16. [2, Thm. 7, p. 285] Consider the curve branch having the parameterization, $x_1 = \varphi_1(t), x_2 = \varphi_2(t), ..., x_n = \varphi_n(t)$. If $H = k[[\varphi_1(t), \varphi_2(t), ..., \varphi_n(t)]]$ and $W({}^*H) = \{\nu_1, \nu_1 + \nu_2, ..., \nu_1 + \nu_2 + ... + \nu_{N-1} + \mathbb{N}\}$, then the multiplicity sequence of the curve branch is

$$\nu_1, \nu_2, ..., \nu_{N-1}, 1, 1, ...$$

We now summarize how we can obtain the multiplicity sequence of a curve branch, having the parameterization $x_1 = \varphi_1(t), ..., x_n = \varphi_n(t)$:

(1) We start with $H = k[[\varphi_1(t), ..., \varphi_n(t)]]$ which is the completion of the local ring at the singularity.

(2) We construct the Arf closure ${}^*H$ in $k[[t]]$.

(3) We obtain $W({}^*H)$, which has the orders of the elements of ${}^*H$.

(4) We construct the characteristic sub-semigroup of $g_\chi(W({}^*H))$ of $W({}^*H)$.

(5) We find the minimal set of generators $\chi_1, ... \chi_h$ of this characteristic semigroup

Definition 3.2.17. $\chi_1, ..., \chi_h$ are the *Arf characters* of H.

(6) We apply the modified Jacobian algorithm to the Arf characters to obtain the multiplicity sequence of the curve branch.

If a curve branch has a parameterization in the form as in Example 3.2.14:

$$x_1 = t^{a_1}, \quad x_2 = t^{a_2}, \quad \dots \quad , x_n = t^{a_n},$$

then $W(^*H)$ of the ring $H = \alpha_{i_1 \dots i_n} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}$ is the Arf closure of the semigroup generated by the integers $a_1, a_2, ..., a_n$. In this case step (1) is easy. But if only one of the parameterization polynomials has more than one term, that is if it has higher degree terms, then step (1) involves many calculations, as can be seen from the second example above. Even constructing the ring H is difficult in this case, because the effects of higher degree terms must be considered carefully. In the second example above, $X = t^6, Y = t^8 + t^{11}, Z = t^{14}$. Thus, the elements $XY - Z = t^{17}$, and $X^4 - Y^3 = 3t^{27} + 3t^{30} + t^{33}$ are in H, but their orders 17 and 27 can not be generated by the orders of X, Y, and Z. In chapter 5, we propose an algorithm for constructing the ring H. We will work with rings which contain power series of all orders after some r. Determining r is also very important in order to write H as:

$$H = k + kS_{i_1} + kS_{i_2} + \dots + k[[t]]S_{i_h},$$

where $i_h = r$. While passing from H to H_1, S_{i_k} 's ($k > 1$) are divided by S_{i_1}, and power series are formed. The terms of these power series, which are of degree higher than $i_h - i_1$ are unnecessary, because H_1 contains power series of all orders after $i_h - i_1$.

We have seen from the Examples 3.2.14 and 3.2.15 that constructing H_i's by using the observation mentioned is easier, because instead of constructing H_i's, we determine their generators.

In chapter 5, we will also propose algorithms for steps (4) and (5).

For a ring H, we can now define a *base* of $^*H$. We choose $X_1 \in^* H$, such that order of X_1 is minimal. Then, we choose $X_2 \in^* H$ such that, order of X_2 is not in the set $W(\overline{k[[X_1, X_2]]})$, and is minimal. ($\overline{k[[X_1, X_2]]}$ denotes the Arf closure of $k[[X_1, X_2]]$.) If $X_1, ..., X_{l-1}$ are chosen in this way, then we choose $X_l \in^* H$ such that, order of X_l is not in the set $W(\overline{k[[X_1, X_2, ..., X_{l-1}]]})$

and it is minimal. This process stops after finitely many steps. Such a set $\{X_1, X_2, ..., X_m\}$ is called a *base* of $^*H$.

Theorem 3.2.18. [2, Thm. 4, p. 272] If $\{X_1, X_2, ..., X_m\}$ is a base of $^*H$, then the integers, order of X_1, order of X_2, ..., order of X_m, depend only on $^*H$, and form a subset of the Arf characters of $^*H$.

We call order of X_1, order of X_2, ..., order of X_m the *base characters of* $^*H$. This number m is called the *projection dimension of* $^*H$. If H is the completion of the local ring of a curve branch, then the number m of the base characters of $^*H$ shows the minimum dimension in which the branch is capable of existing.

Chapter 4

THE PROBLEM OF FROBENIUS AND AN ALGORITHM FOR ITS SOLUTION

4.1 The Problem of Frobenius

We have seen that, the semigroup $W(^*H)$ of the completion of local ring at the singularity is used in order to produce the Arf characters of the curve. If the generators of the semigroup are relatively prime, then the semigroup contains all the integers after some n. Looking for the largest integer which is not included in a semigroup is a famous problem known as the *problem of Frobenius*. Formally, given relatively prime positive integers $a_1, ..., a_k$ all greater than 1, a number a is said to be generated by $a_1, ..., a_k$ over $\mathbb{N}$, if there exist nonnegative integers $x_1, ..., x_k$ such that,

$$a = a_1x_1 + a_2x_2 + ... + a_kx_k.$$

The *problem of Frobenius* is to determine the largest integer

$$g(a_1, ..., a_k)$$

that can not be generated by $a_1, ..., a_k$.[1]

The problem of Frobenius has a solution in closed form for $k = 2$,

$$g(a_1, a_2) = a_1 a_2 - a_1 - a_2.$$

For $n > 2$, there are no known solutions in closed form.

This problem has a vast literature but there are only a few algorithms for the most general case. Brauer and Shockley gave an algorithm for the solution of problem of Frobenius in three variables [5]. Heap and Lynn, who gave a graph theoretic algorithm were the first to give a general algorithm [11]. Selmer gave a bound for $g(a_1, ..., a_k)$ and illustrated his general results in some cases where explicit relations for $g(a_1, ..., a_k)$ are easily obtained [16]. Rödseth considered the problem of Frobenius in three variables and found a formulae for finding $g(a_1, a_2, a_3)$. He also gave a note on the general case [15]. Sertöz and Özlük considered the problem as the investigation of the lattice points of the region $\{(x_1, ..., x_n) \in \mathbb{R}^n \mid x_1, ..., x_n > 0\}$ [18]. After showing that $g(a_1, ..., a_n)$ is the largest element of a finite set S constructed using an infinite set, they proposed an algorithm.

Lewin proposed an algorithm for the most general case [14]. He uses a representation of positive integers by binomial coefficients. First, he shows that for every positive integer n and b, there are nonnegative integers $a_1, a_2, ..., a_b$ such that $a_i > 0$ implies $a_i > a_{i+1}$, $a_i = 0$ implies $a_{i+1} = 0$, and they satisfy

$$n = \sum_{i=1}^{b} \binom{a_i}{b - i + 1}.$$

$a_1, ..., a_b$ is called an a-sequence of length b corresponding to the integer n, and is referred to as the $a_{(b)}$-sequence of n. An algorithm for constructing a-sequences is outlined in Lewin [14, p. 69]. According to this, the $a_{(4)}$-sequence of $n = 100$ is $a_{(4)}(100) = 8, 6, 5, 0$. In fact $100 = \binom{8}{4} + \binom{6}{3} + \binom{5}{2}$.

Two distinct positive integers have distinct representations [14, p. 69]. Hence, there is a one to one correspondence between the set of positive integers

[1] In fact, this problem has an interesting interpretation as follows. Given sufficient supply of coins of denominations $a_1, ..., a_k$, determine the largest amount which can not be formed by means of these coins.

and the set of a-sequences of length b. x^+ is defined to be $\max(x,0)$ and $a^{(i)}_{(b)}(n)$ denotes the i-th element of the $a_{(b)}$ sequence of n. Now another sequence called the $c_{(b)}$-sequence of n is defined as follows. $c^{(i)}_{(b)}(n) = (a^{(i)}_{(b)}(n) + i - b)^+, i = 1, ..., b$. This is a monotonically decreasing sequence. Since there is a one to one correspondence between the a-sequences and the c-sequences of the same length, there is also a correspondence between the set of positive integers and the set of c-sequences of length b. For example, the c-sequence of 100 of length 4 is $c_{(4)}(100) = 5,\ 4,\ 4,\ 0$.

Let $a_1, a_2, ..., a_k$ be relatively prime positive integers all greater than 1 and $a_1 < a_2 < ... < a_k$. Define $d_l = a_{l+2} - a_{l+1}$ for $1 \le l \le k-2$. Lewin constructs the table $T = \{a_{ij}\}$, where a_{ij}'s are defined in the following way by considering the c-sequences of length $k-2$:

$$a_{ij} = ia_2 + \sum_{l=1}^{k-2} c^{(l)}_{(k-2)}(j) d_l, \qquad 0 \le j < \begin{pmatrix} k-2+i \\ k-2 \end{pmatrix}$$

where $c^{(l)}_{(k-2)}(j)$ is the l-th element of $c_{(k-2)}$ sequence of j. From the above equation, a_{ij}'s are in fact the elements generated by $a_2, ..., a_k$. This can be seen if the equation is written term by term. That is,

$$\begin{aligned} a_{ij} &= ia_2 + c^{(1)}_{(k-2)}(j)(a_3 - a_2) + c^{(2)}_{(k-2)}(j)(a_4 - a_3) + ... + c^{(k-2)}_{(k-2)}(j)(a_k - a_{k-1}) \\ &= (i - c^{(1)}_{(k-2)}(j))a_2 + (c^{(1)}_{(k-2)}(j) - c^{(2)}_{(k-2)}(j))a_3 + ... + \\ &\quad (c^{(k-3)}_{(k-2)}(j) - (c^{(k-2)}_{(k-2)}(j))a_{k-1} + c^{(k-2)}_{(k-2)}(j)a_k \end{aligned}$$

Since c-sequences are monotonically decreasing, the coefficients of $a_3, ..., a_k$ are nonnegative integers. The coefficient of a_2, $(i - c^{(1)}_{(k-2)}(j))$ is nonnegative because $j < \begin{pmatrix} k-2+i \\ k-2 \end{pmatrix}$.

The i_m-th row consists of $\begin{pmatrix} k-2+i_m \\ k-2 \end{pmatrix}$ elements satisfying $ia_2 \le a_{i_m j} \le ia_k$.

Then Lewin uses the following lemma of Brauer and Shockley [5].

Lemma 4.1.1. Let $a_1, a_2, ..., a_k$ be relatively prime positive integers all greater than 1. Let L be a complete system of residues $l \not\equiv 0 \pmod{a_1}$. For each $l \in L$, there is a smallest positive integer $t_l \equiv l \pmod{a_1}$ which is generated by $a_2, ..., a_k$, that is $t_l = \sum_{i=2}^{k} a_i x_i$ where x_i's are nonnegative integers. Then,

$$g(a_1, ..., a_k) = \max\{t_l : l = 1, ..., a_1\} - a_1.$$

Proof: Let a be a positive integer. If $a \equiv 0 \pmod{a_1}$, then a is a multiple of a_1, so it is generated by a_1. If $a \equiv l \not\equiv 0 \pmod{a_1}$, by the definition of t_l, it follows that $t_l - a_1$ is the largest integer $\equiv l \pmod{a_1}$ which can not be generated by $a_2, ..., a_k$, and also the largest integer $\equiv l \pmod{a_1}$ which can not be generated by $a_1, a_2, ..., a_k$. Hence, max $\{t_l : l = 1, ..., a_1\} - a_1$ is the largest integer that can not be generated by $a_1, ..., a_k$. □

Since the table constructed by Lewin consists of the elements generated by $a_2, ..., a_k$, the t_l's defined above correspond to certain a_{ij}'s in the table. Let a_{i_0, j_0} be the maximum of these a_{ij}'s. Then

$$g(a_1, a_2, ..., a_k) = a_{i_0 j_0} - a_1.$$

The table being infinite, finding t_l's is a problem. Recall that t_l is the smallest positive integer generated by $a_2, ..., a_k$, which is equivalent to l (mod a_1). The table is not constructed in such a way that the integers are generated in ascending order, so when you generate an a_{ij} that is equivalent to l (mod a_1), there is no guarantee that, a smaller integer equivalent to $l \pmod{a_1}$ will not be generated later in the table. Hence, Lewin finds an upper bound for i_0 of $a_{i_0 j_0}$. All the elements in the table up to this row are calculated by the formula and since t_l's are among these, they can be found. This makes Lewin's algorithm slow, because it has to do unnecessary calculations. On the other hand, the algorithm works fast for certain sequences called *almost arithmetic sequences* [14].

4.2 An Algorithm for the Solution of the Problem of Frobenius

We will give an algorithm for the solution of the problem of Frobenius which uses Lemma 4.1.1. The main problem we have, while using this lemma to find the solution of the problem of Frobenius, is how to obtain the t_l's. Recall that t_l is the smallest positive integer generated by $a_2, ..., a_k$ and $t_l \equiv l \pmod{a_1}$. If we have the elements generated by $a_2, ..., a_k$ in ascending order, the first representative of a residue class l will be t_l. In this way, we will stop when we have representatives of all residue classes and the last of them will be the maximum one, that is the maximum of t_l's. What is more, there is no need to deal with t_l's. It is sufficient to have representatives of all residue classes and the last representative of a residue class we find is the maximum of t_l's, just because the list is in ascending order. Our algorithm depends on these ideas.

Given relatively prime integers $a_1 < a_2 < ... < a_k$ all greater than 1, we aim to find $g(a_1, ..., a_k)$. We begin by constructing a set L consisting of all residues $l \not\equiv 0 \pmod{a_1}$. We will refer to a set that is sorted in ascending order as a *list*. G will be a list, which contains elements that are generated by $a_2, ..., a_k$. The initial value of G will be $G = [0, a_2]$. We begin by finding the elements between a_2 and $2a_2$, generated by $a_2, ..., a_k$. Hence, we want the elements generated to be less than $C = 2a_2$. We add $C = 2a_2$ at the end of the list. Now, $G = [0, a_2, 2a_2]$. We sum up the generator a_3 with the first element in the list G. During the algorithm it will be determined by a counter c_3, with which element in the list will a_3 be summed up. At the beginning $c_3 = 1$, so a_3 is summed up with the first element in the list, which is 0. (The other generators $a_4, ..., a_k$ have counters $c_4, ..., c_k$ for the same purpose.) If the element generated by summing up a_3 with the first element in the list is less than $C = 2a_2$, then we add it in the list G and sort G in ascending order, and increment the counter c_3. Then we sum up a_3 with the second element in the list, since $c_3 = 2$. If the element generated is less than $C = 2a_2$, we add it to the list and sort the list in ascending order, and increment c_3 again. We continue summing up a_3 with the elements in the list. and applying the same procedure until we generate an element greater than or equal to $C = 2a_2$.

Then we take the generators $a_4, ..., a_k$ one at a time and apply the same procedure. At the end of this, we have all the elements between a_2 and $2a_2$, that belong to the semigroup generated by $a_2, ..., a_k$. Then we find which residue classes have representatives among these elements, and remove them from the set L. If L is not empty after this procedure (i.e., all of the residue classes do not have representatives among these elements of G), then we look for the elements between $2a_2$ and $3a_2$, generated by $a_2, ..., a_k$. by following the above procedure. In this case, we want the elements generated by $a_2, ..., a_k$ to be less than or equal to $C = 3a_2$. We begin by summing up a_3 with the c_3-th element in the list. (We know that the elements generated by summing up a_3 with the previous elements have already been added to the list. Hence, these counters prevent generating the same elements several times.) If the element thus generated is less than or equal to $C = 3a_2$, then we add it to the list and order the list, and we increment c_3. We continue in the same manner, until we generate an element that is greater than $C = 3a_2$. Then we take the generator a_4 and begin by summing up a_4 with the c_4-th element in the list. We apply the same procedure defined above. When we finish doing this for the remaining generators, we find the elements between $2a_2$ and $3a_2$, generated by $a_2, ..., a_k$. Again we find which other residue classes have representatives in this ordered interval, and subtract these residue classes from the set L. After finitely many steps, L becomes empty, signaling the end of the algorithm. The integer, which is the representative of the last residue class is the maximum of t_l's defined in the Lemma 4.1.1. Hence, we subtract a_1 from this integer to obtain $g(a_1, a_2, ..., a_k)$.

See figure 4.1 for the flow chart of this algorithm.

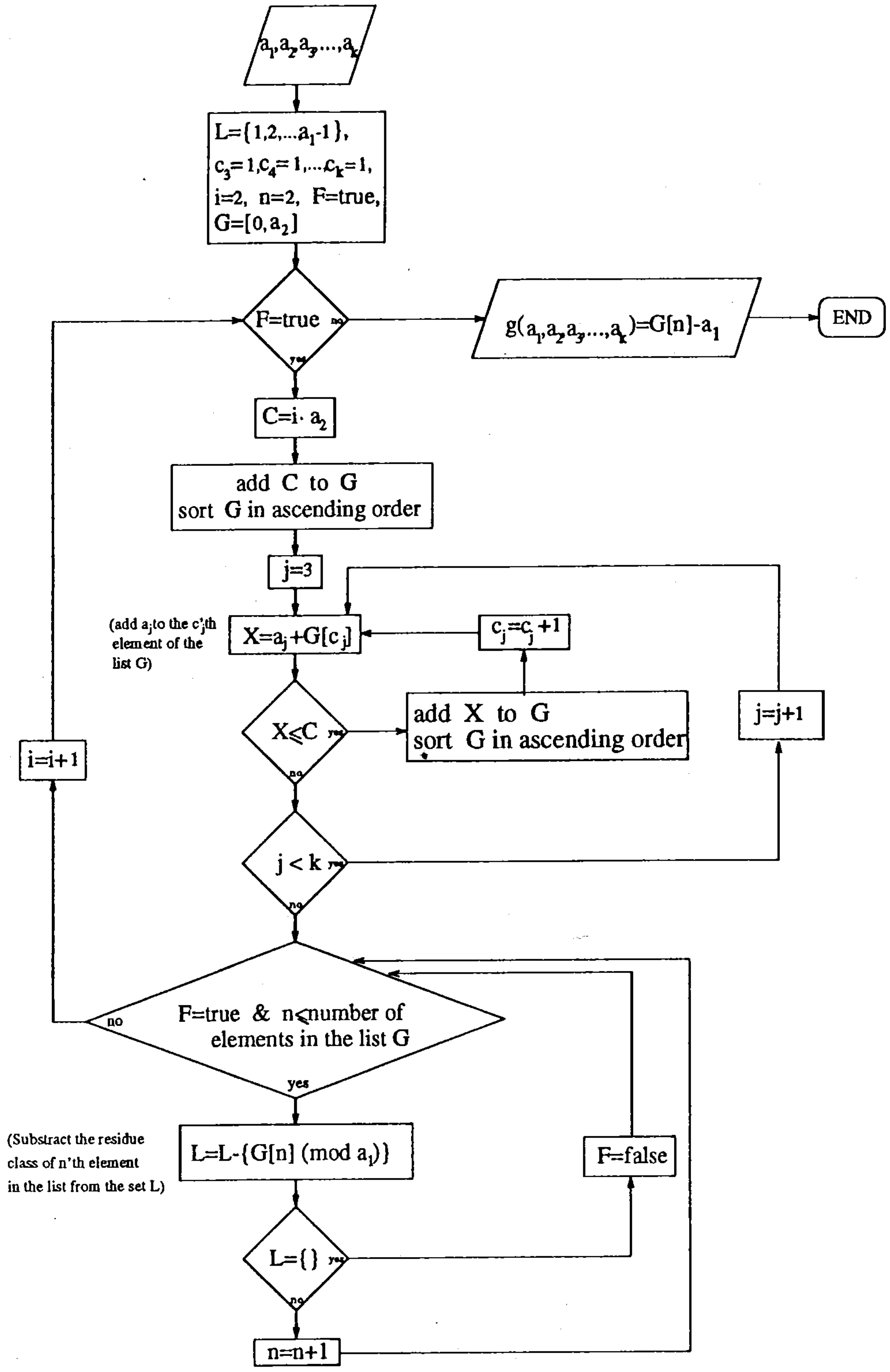
a1,a2,a3,...,ak
L={1,2,...a1-1}, c3=1,c4=1,...ck=1, i=2, n=2, F=true, G=[0,a2]
F=true
no
yes
g(a1,a2,a3,...,ak)=G[n]-a1
END
C=i· a2
add C to G
sort G in ascending order
j=3
(add aj to the cj'th element of the list G)
X=aj+G[cj]
cj=cj+1
X≤C
yes
no
add X to G
sort G in ascending order
j=j+1
i=i+1
j < k
yes
no
F=true & n≤number of elements in the list G
no
yes
(Substract the residue class of n'th element in the list from the set L)
L=L-{G[n] (mod a1)}
F=false
L={}
yes
no
n=n+1

Figure 4.1: Flow chart of the Algorithm for the problem of Frobenius

Chapter 5

AN ALGORITHM FOR FINDING THE ARF CHARACTERS OF A BRANCH

In this chapter, we propose an algorithm for finding the Arf characters of a given curve branch. Our algorithm follows the program given at the end of chapter 3, where we have also explained which stages of the program are difficult.

In order to be able to apply the algorithm, we work with polynomials. As an input, we have a parameterization of a branch curve,

$$\begin{aligned} x_1 &= \varphi_1(t) \\ &\vdots \\ &\vdots \\ x_n &= \varphi_n(t) \end{aligned}$$

where $\varphi_1(t), ..., \varphi_n(t)$ are polynomials in t.

The first step of the algorithm is to generate $H = k[[\varphi_1(t), ..., \varphi_n(t)]]$, the completion of the local ring. We want H to contain power series of all orders after some r. (Recall that order of φ is the degree of the smallest degree term present in an element φ of $k[[t]]$. We will denote order of φ by $ord(\varphi)$.) We restrict the input of our algorithm to branch curves, which have polynomial parameterizations, and the completion of the local ring of which contain power series of all orders after some r.

Suppose, on the other hand, that a curve branch is parameterized as above where $\varphi_1(t), ..., \varphi_n(t)$ are formal power series such that the ring $H = k[[\varphi_1(t), ..., \varphi_n(t)]]$ contains power series of order r or higher. Suppose that we can guess an upper bound for r, i.e., we guess that H will contain all power series of order R or higher, where $R \geq r$. Let $\varphi_{i,R}(t)$ be the part of $\varphi_i(t)$ consisting of elements of order less than or equal to R. For example, if

$$\varphi_1(t) = a_1 t^m + a_2 t^{m+1} + ...,$$

where $a_1 \neq 0$, then

$$\varphi_{1,R}(t) = a_1 t^m + a_2 t^{m+1} + ... + a_{R+m-1} t^R.$$

It then follows that

$$k[[\varphi_1(t), ..., \varphi_n(t)]] = k[[\varphi_{1,R}(t), ..., \varphi_{n,R}(t)]].$$

For this reason restricting our attention to only polynomial parameterization is not a significant restriction.

As input we have a set of polynomials $\varphi_1(t), ..., \varphi_n(t)$, which must be the parameterization of a curve branch passing through $(0, ..., 0)$. From the definition of branch given in section 2.2, the completion of the local ring of a branch of a curve at $(0, ..., 0)$ contains power series of all orders after some r. For a given parameterization $x_1 = \varphi_1(t), ..., x_n = \varphi_n(t)$, if $ord(\varphi_1(t)), ..., ord(\varphi_n(t))$ are relatively prime, this is a sufficient but not necessary condition for that parameterization to correspond to a branch. Hence, it is also sufficient for the termination of the algorithm in finite steps.

Step (1) The first step of the algorithm is to construct the ring H in the form

$$H = k + kS_{i_1} + kS_{i_2} + ... + k[[t]]S_{i_h},$$

where $S_{i_1}, S_{i_2}, ..., S_{i_h}$ are arbitrary elements of order $i_1, i_2, ..., i_h$, respectively, chosen from H, and H contains power series of all orders after i_h.

To do this, we adapt the algorithm used to solve the problem of Frobenius. In section 4.2, we have proposed a method to find the integers between ia_2

and $(i+1)a_2$, generated by $a_2, ..., a_k$. The procedure here will be similar. We will generate polynomials instead of integers, and we will compare the orders of polynomials, instead of comparing the integers. In fact, if each parameterization polynomial has only one term, that is

$$x_1 = t^{a_1}; \quad x_2 = t^{a_2}, \quad ..., \quad x_n = t^{a_n},$$

then constructing H is equivalent to constructing the semigroup generated by $a_1, ..., a_n$. In that case, the method given in the algorithm to solve the problem of Frobenius for generating the elements of the semigroup generated by $a_2, ..., a_k$, is sufficient to generate H. When the parameterization polynomials have higher order terms, then constructing H involves more work, because the difference of two polynomials having the same order is a nonzero element of H, and all these differences must be considered.

We begin by dividing each input polynomial by an appropriate coefficient so that the coefficients of the smallest degree terms of all input polynomials become 1. Then, we construct a generator set G consisting of these polynomials. We sort G according to the orders of the polynomials such that, order of the i-th element in G is less than or equal to the order of the $i+1$-th element in G. Note that we are working with the orders and not the degrees of the polynomials.

If the orders are equal, then the degree of the second term of the i-th element is less than or equal to the degree of the second term of the $i+1$-th element. If the first j-terms of the i-th and the $i+1$-th elements in G have the same degrees, then degree of the $j+1$-th term of the i-th element in G is less than the degree of the $j+1$-th term of the $i+1$-th element in G. We denote this sorted set as $G = [G[1], G[2], ..., G[n]]$, and its i-th element as $G[i]$.

We have another set S sorted in ascending order which contains the orders of the polynomials generated by the polynomials $G[1]. ..., G[n]$, as the algorithm continues. The initial value of S is $[0, i_1]$, where $i_1 = ord(G[1])$. We denote the i-th element of this list as $S[i]$.

We refer to the ordered sets S and G as lists.

We have polynomial sets corresponding to the elements of the list S. If α is an element of S, the polynomial set corresponding to α will be the set of polynomials in H of order α. We denote this set by $poly(\alpha)$. As the algorithm continues, when a polynomial of order α is generated. it will be added to the set $poly(\alpha)$. Since $S = [0, i_1]$ initially, we have $poly(0) = \{1\}$, and $poly(i_1) = \{G[1]\}$ as the initial polynomial sets. We need such a construction in order to be able to consider differences of all the different polynomials having the same order.

(1.1) We begin with finding the polynomials in H that have orders between $ord(G[1]) = i_1$ and $ord(G[1]^2) = 2i_1$. Hence, we want the polynomials generated to be of order less than or equal to $C = ord(G[1]^2) = 2i_1$. We add C at the end of the list S. There must now be a polynomial set corresponding to C. Hence, we will have $poly(C) = \{G[1]^2\}$.

Now, at the beginning of this part, we have $S = [0, i_1, 2i_1]$, and the corresponding polynomial sets $poly(0)$, $poly(i_1)$, and $poly(2i_1)$.

(1.1.1) In this step of the algorithm, we will find the polynomials in H, generated by the first and the second elements $G[1]$ and $G[2]$ in the list G, and having orders less than or equal to $C = 2i_1$.

We take $G[2]$, the second element in generator list G. Since zero is the first element of the list S, it is the possible least order. Hence, we begin by multiplying $G[2]$ with the elements of the set $poly(0)$. When the polynomial $G[2]$ is multiplied with a polynomial of order 0, its order does not change. Hence, we first control if $ord(G[2])$ is less than or equal to $C = 2i_1$, because at this step we want to generate polynomials of order less than or equal to $C = 2i_1$. If $ord(G[2])$ is less than or equal to C, then we will multiply $G[2]$ with polynomials of 0 order, which are the elements of the set $poly(0)$. The polynomial set $poly(0)$ has one element, which is 1. We multiply $G[2]$ with 1, and obtain $f = G[2]$. Now, there are two possibilities: $ord(f)$ is in the list S, and $ord(f)$ is not in the list S.

If $ord(f)$ is in the list S, then there is already a polynomial set consisting of a polynomial or polynomials of the same order. That means $poly(ord(f))$ is nonempty. We subtract f from the elements of this set $poly(ord(f))$. In this way, the differences of f and the polynomials having the same order with f are considered. The nonzero difference polynomials obtained are divided by appropriate coefficients so that the coefficients of their smallest degree terms become 1. Then the obtained polynomials are added to the list G as new generators, and G is sorted according to orders of its elements as defined above. For every new generator added to the list G, n is incremented. After these, we add f to the polynomial set $poly(ord(f))$. By this construction, all the effects of the higher degree terms of the polynomials can be considered.

If $ord(f)$ is not in the list, then we add $ord(f)$ to the list S, and sort S in ascending order. There must be a polynomial set corresponding to every element of the list S, so we now have $poly(ord(f)) = \{f\}$.

There is only one polynomial of order 0, because $poly(0) = \{1\}$. Thus, we have now finished multiplying $G[2]$ with polynomials of order 0.

We now turn back to the beginning of (1.1.1). We have multiplied $G[2]$ with polynomials of the least order 0, which was the first element of the list S. The second element of the list is i_1. We will multiply $G[2]$ with polynomials of order i_1, that is, with the elements of the polynomial set $poly(i_1)$. (As the algorithm continues, a counter c_2 is used to determine the set of polynomials, whose elements will be multiplied with $G[2]$. In the beginning of (1.1.1), c_2 was 1. $S[c_2] = S[1] = 0$ so we used the set $poly(0)$. After the procedure was completed, before turning back to the beginning of (1.1.1), c_2 was incremented. Thus, $c_2 = 2$ now. Since $S[c_2] = S[2] = i_1$, we will multiply the elements of $poly(i_1)$ with $G[2]$. Hence, at any step we will multiply the elements of $poly(S[c_2])$ with $G[2]$. All the other generators $G[3], ..., G[n]$, have counters $c_3, ..., c_n$ for the same purpose. Initially, all the counters $c_2, ..., c_n$ are 1. When a new generator is added to the list G, a counter which is equal to 1 is assigned to this new generator.)

When the polynomial $G[2]$ is multiplied with a polynomial of order i_1, a polynomial of order $ord(G[2]) + i_1$ is obtained.

If $ord(G[2]) + i_1$ is less than or equal to $C = 2i_1$, we multiply the first polynomial of the polynomial set $poly(i_1)$ with $G[2]$, and obtain f. We apply to f what we have done above. Then we multiply the second polynomial of the polynomial set $poly(i_1)$ with $G[2]$ and obtain f. We again apply to f the same procedure. We repeat this for all the polynomials in the set $poly(i_1)$. We increment c_2, and turn back the beginning of (1.1.1).

If $ord(G[2]) + i_1$ is greater than $C = 2i_1$, then this signals the end of the step (1.1.1). The polynomials generated by $G[1], G[2]$, which are of orders less than $C = 2i_1$, are found.

Then the stopping condition of this step is $ord(G[2]) + S[c_2] > C = 2i_1$.

(1.1.2) In this step of the algorithm, we will find the polynomials generated by the first, second, and third elements $G[1]$, $G[2]$, and $G[3]$ in the list G, and having orders less than or equal to $C = 2i_1$. We take the third generator $G[3]$ in G, which has counter $c_3 = 1$ at the beginning. We compare the addition of

$ord(G[3])$ and c_3-th element in the list S, with $C = 2i_1$. If $ord(G[3]) + S[c_3]$ is greater than $C = 2i_1$, this is the end of step (1.1.2). If $ord(G[3]) + S[c_3]$ is less than or equal to C, then we take the polynomial set corresponding to the c_3-th element in the list S, which is $poly(S[c_3])$. We take the first element of the set $poly(S[c_3])$, and multiply it with $G[3]$ to obtain f. We apply the same procedure explained above. For all the elements in the polynomial set $poly(S[c_3])$, we repeat this procedure. We increment c_3, and turn back to the beginning of (1.1.2). We continue in the same manner, until $ord(G[3]) + S[c_3] > C = 2i_1$, signaling the end of step (1.1.2). At this point, we have the polynomials generated by the first, second, and third elements $G[1]$, $G[2]$, and $G[3]$ in the list G, and having orders less than or equal to $C = 2i_1$.

We repeat this process for the remaining generators of G. At the end of this, we have polynomials of orders less than or equal to $C = 2i_1$, generated by $G[1], G[2], ..., G[n]$. Also, new generators coming out from the differences of polynomials of the same orders were added to the generator list.

Step (1.1) is completed by counting the number of polynomials of consecutive orders, last of which is $2i_1$. We can do this by using the list S, because it contains all the orders of the generated polynomials. The list S has C as its last element. By a counter m_1, we count the number of consecutive integers in the list S, last of which is C. This is the end of step (1.1).

If $m_1 \geq ord(G[1]) = i_1$, then this signals the end of the algorithm. Because if we have i_1 polynomials of consecutive orders, we can have polynomials of all the remaining orders by multiplying these with $G[1]$ consecutively. If m_1 is less than i_1, we start step (1.2).

(1.2) In this step, we find the polynomials in H that have orders between $ord(G[1]^2) = 2i_1$ and $ord(G[1]^3) = 3i_1$. Hence, we want the polynomials generated to be of order less than $C = ord(G[1]^3) = 3i_1$. We add C at the end of the list S. There must now be a polynomial set corresponding to C. Hence, we will have $poly(C) = \{G[1]^3\}$

(1.2.1) We take $G[2]$. In the end of step (1.1.1), for c_2 we had $ord(G[2]) + S[c_2] > 2i_1$. In this step, we first control if $ord(G[2]) + S[c_2] \leq C = 3i_1$.

If $ord(G[2]) + S[c_2]$ is less than or equal to $C = 3i_1$, we multiply the first polynomial of the polynomial set $poly(S[c_2])$ with $G[2]$, and obtain f. We apply to f what we have done above. Then we multiply the second polynomial of the polynomial set $poly(S[c_2])$ with $G[2]$ and obtain f. We again apply to f the same procedure. We repeat this for all the polynomials in the set $poly(S[c_2])$.

We increment c_2, and turn back the beginning of (1.2.1).

If $ord(G[2]) + S[c_2]$ is greater than $C = 3i_1$, then this signals the end of the step (1.2.1).

Again, the stopping condition of this step is $ord(G[2]) + S[c_2] > C = 2i_1$.

The steps $(1.2.2), ..., (1.2.n-1)$ repeat the same procedure for all the elements of the list G. At the end of this, we have polynomials in H that have orders less than or equal to $C = 3i_1$.

Step (1.2) is completed by counting the number of polynomials of consecutive orders, last of which is $3i_1$.

(1.k) In this step, we find the polynomials in H that have orders between $ord(G[1]^k) = ki_1$ and $ord(G[1]^{k+1}) = (k+1)i_1$. At the end of step (1.k), we have polynomials in H of orders less than or equal to $C = (k+1)i_1$. The algorithm stops at the end of this step, where we have $m_1 \geq i_1$, showing that there are at least i_1 polynomials of consecutive orders. If we have this condition satisfied, we can find the integer i_h after which power series of all orders exist in H. Since C is the last one of the m_1 consecutive integers, $i_h = C - m_1 + 1$. Recall that our conditions on the input polynomials assure that the algorithm stops in finite steps.

Now, to construct H, we can choose one polynomial from every polynomial set corresponding to the elements of the list S, which are less than or equal to i_h. At the end, we have $S = [0, i_1, i_2, ..., i_h, ..., C]$. We can choose $1 \in poly(0)$, $S_{i_1} \in poly(i_1)$, $S_{i_2} \in poly(i_2)$, ..., $S_{i_h} \in poly(i_h)$ to construct [2]

$$H = k + kS_{i_1} + kS_{i_2} + ... + k[[t]]S_{i_h}.$$

See figure 5.1 for the flow chart of this algorithm.

Step (2) The second step is constructing the Arf closure $^*H$ in $k[[t]]$. When the above algorithm is applied to input polynomials $\varphi_1, ..., \varphi_n$, we obtain the polynomials $S_{i_1}, S_{i_2}, ..., S_{i_h}$ of the ring H, and we have [2]

$$H = k + kS_{i_1} + kS_{i_2} + ... + k[[t]]S_{i_h}.$$

We will use the method given in section 3.2 to construct the Arf closure $^*H$ of H. Recall that H_1 denotes the ring

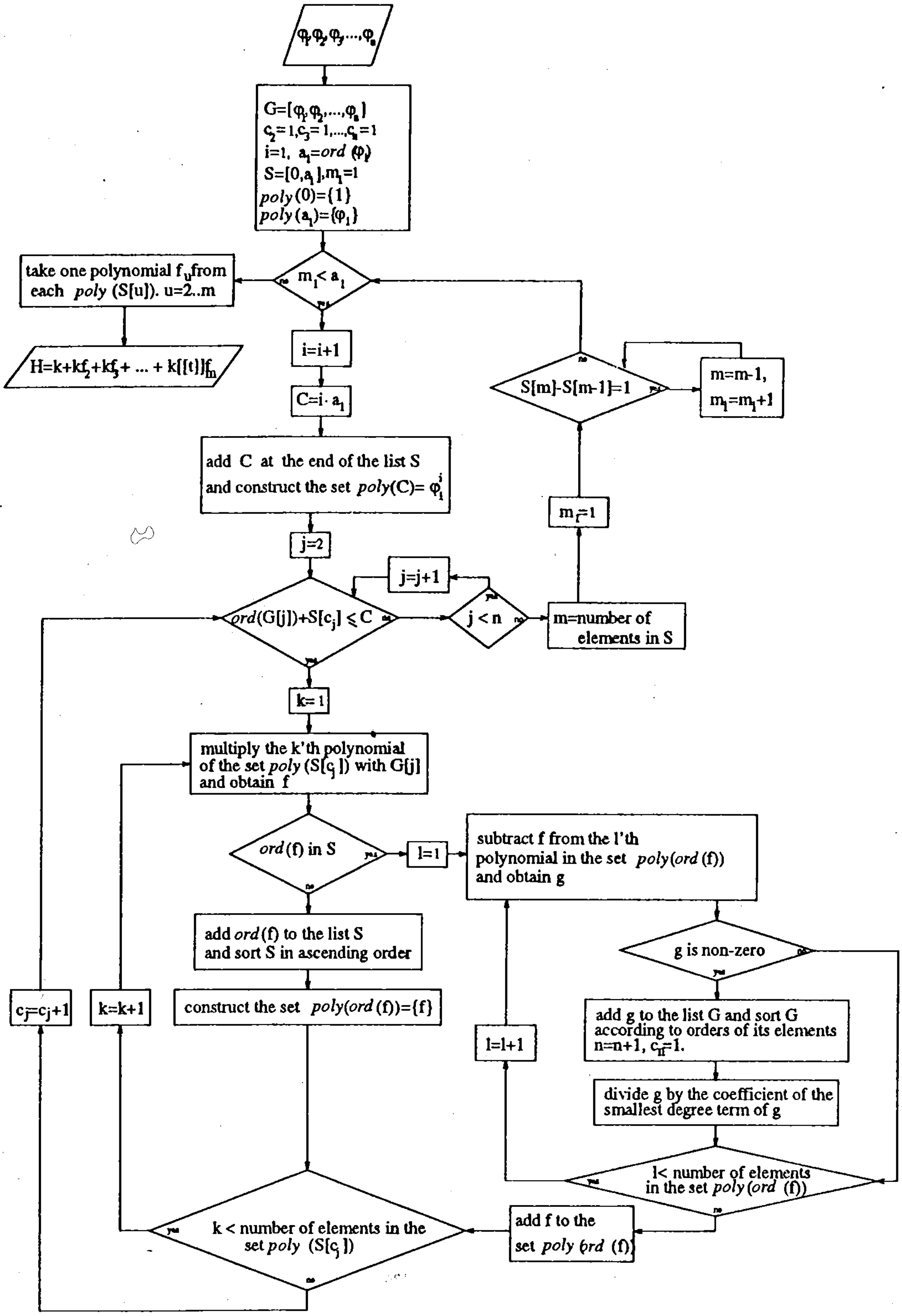
φ1,φ2,φ3,...,φn
G=[φ1,φ2,...,φn]
c2=1,c3=1,...,cn=1
i=1, a1=ord (φ1)
S=[0,a1],m1=1
poly(0)={1}
poly(a1)={φ1}
m1< a1
no
yes
take one polynomial fu from each poly (S[u]). u=2..m
H=k+kf2+kf3+ ... + k[[t]]fm
i=i+1
C=i· a1
add C at the end of the list S and construct the set poly(C)= φ1^i
S[m]-S[m-1]=1
no
yes
m=m-1, m1=m1+1
m1=1
j=2
j=j+1
ord(G[j])+S[cj] ⩽ C
no
yes
j < n
yes
no
m=number of elements in S
k= 1
multiply the k'th polynomial of the set poly (S[cj]) with G[j] and obtain f
ord(f) in S
yes
no
l=1
subtract f from the l'th polynomial in the set poly(ord (f)) and obtain g
add ord(f) to the list S and sort S in ascending order
g is non-zero
no
yes
cj=cj+1
k=k+1
construct the set poly(ord (f))={f}
add g to the list G and sort G according to orders of its elements n=n+1, cn=1.
l=l+1
divide g by the coefficient of the smallest degree term of g
l< number of elements in the set poly(ord (f))
yes
no
add f to the set poly (ord (f))
k < number of elements in the set poly (S[cj])
yes
no

Figure 5.1: Flow chart of the Algorithm for the construction of H

$$H_1 = [I_{i_1}] = \sum k(\frac{S_{i_2}}{S_{i_1}})^{\alpha_2}(\frac{S_{i_3}}{S_{i_1}})^{\alpha_3}...(\frac{S_{i_{h-1}}}{S_{i_1}})^{\alpha_{h-1}} + k[[t]](\frac{S_{i_h}}{S_{i_1}}),$$

and $^*H = k+^*H_1S_{i_1}$. The ring H_{i+1} is derived from H_i in the same manner, H_1 is derived from H. Then for sufficiently large N, $H_N = k[[t]]$. Let T_{i+1} be an element of least positive order in H_i, then we have [2]

$$\begin{aligned} ^*H &= k + kT_1 + ^*H_2T_1T_2 \qquad (T_j = S_{i_j}) \\ &= k + kT_1 + kT_1T_2 + ^*H_3T_1T_2T_3 \\ ... & \;............. \\ &= k + kT_1 + kT_1T_2 + ... + kT_1T_2...T_{N-1} + ^*k[[t]]T_1T_2...T_{N-1}T_N \end{aligned}$$

Here, we give an algorithm to determine T_i's.

(2.1) We begin with $S_{i_1}, S_{i_2}, ..., S_{i_h}$ of H. At the end of this step, we will have H_1 defined above. $T_1 = S_{i_1}$ from the definition. We divide $S_{i_2}, S_{i_3}, ..., S_{i_h}$ by S_{i_1}. The results of these divisions may be power series of infinite terms, but the terms of these power series which have degrees greater than $i_h - i_1$ are unnecessary, because H_1 contains power series of all orders after $i_h - i_1$. For this reason in each power series $\frac{S_{i_r}}{S_{i_1}}$ we truncate terms of orders higher than $i_h - i_1$, and denote the resulting polynomial still by the same notation $\frac{S_{i_r}}{S_{i_1}}$. Hence, from the divisions $\frac{S_{i_2}}{S_{i_1}}, ..., \frac{S_{i_h}}{S_{i_1}}$, we have polynomials, which have no terms having degrees greater than $i_h - i_1$. These polynomials obtained from the divisions are generators for the elements of H_1 of orders less than $i_h - i_1$. The algorithm we proposed to construct H in step (1) is applied to these polynomials with slight modification. In constructing H, we do not know an r such that H contains power series of all orders after r. In this case, we know that the ring H_1 contains power series of all orders after $i_h - i_1$. This gives us stopping condition, since it is not necessary to generate polynomials of orders greater than $i_h - i_1$. This can be done by comparing $i_h - i_1$ with C at every step (1.k) of step (1). If at step (1.k), we have $C \geq i_h - i_1$, we equate $C = i_h - i_1$ to generate polynomials of orders less than or equal to $i_h - i_1$, and we stop at the end of this step (1.k). Also, recall that in the algorithm proposed in step (1), new generators come from the differences of two polynomials having the same order. If a new generator obtained in this way has order greater than $i_h - i_1$, it does not have any effect on constructing H_1. Thus, we do not need to add new generators of orders greater than $i_h - i_1$ in the generator set G. And for the same reason whenever a new generator of order $\leq i_h - i_1$ is obtained, we truncate its elements of order $> i_h - i_1$ before we add it to the generator set G. With these modifications, we apply the algorithm proposed in step (1) to construct H_1. We have

$$H_1 = k + kS_{j_1} + kS_{j_2} + ... + k[[t]]S_{j_s}.$$

(2.2) T_2 is an element of least positive order in H_1. We choose T_2 as j_1. We apply the same procedure defined in (2.1) to construct H_2.

We continue in the same manner, until at the end of the step $(2.N-1)$ we have

$$H_{N-1} = k + kS_{\alpha_1} + kS_{\alpha_2} + k[[t]]S_{\alpha_3},$$

where $\alpha_2 - \alpha_1 = 1$, signaling the end of step (2) of the algorithm. Since $\alpha_2 - \alpha_1 = 1$, we have $H_N = k[[t]]$, and so ${}^*H_N = k[[t]]$. From the definition, $T_N = S_{\alpha_1}$. We have now all T_i's necessary to construct ${}^*H$:

$${}^*H = k + kT_1 + kT_1T_2 + ... + kT_1T_2...T_{N-1} + k[[t]]T_1T_2...T_{N-1}T_N$$

Step (3) The third step of the algorithm is constructing $W({}^*H)$. We can easily write from the above equation

$$\begin{aligned} W({}^*H) &= \{0, ord(T_1), ord(T_1) + ord(T_2), ord(T_1) + ord(T_2) + ord(T_3), ..., \\ &\quad ord(T_1) + ord(T_2) + ... + ord(T_N) + \mathbb{N}\}. \end{aligned}$$

Step (4) The fourth step is the construction of the characteristic subsemigroup of $g_\chi(W({}^*H))$ of $W({}^*H)$, and finding a minimal set of generators $\chi_1, ..., \chi_l$ of this characteristic semigroup. These are Arf characters of H. Now, we propose an algorithm to find Arf characters of H from $W({}^*H)$. We have

$$W({}^*H) = \{0, \nu_1, \nu_1 + \nu_2, ..., \nu_1 + \nu_2 + ... + \nu_N + \mathbb{N}\}.$$

Here, we give a notation. Let $S(a_1, ..., a_k) = \{\alpha_1 a_1 + ... + \alpha_k a_k \mid \alpha_1, ..., \alpha_k \in \mathbb{N}\}$. Hence, $S(a_1, ..., a_k)$ denotes the semigroup generated by $a_1, ..., a_k$. We also recall here that to construct the Arf closure of $S(a_1, ..., a_k)$, we use the method given using the Theorem 3.2.13; we apply modified Jacobian algorithm to $a_1, ..., a_k$. We obtain from this $d_1, d_2, ..., d_{m-1}, d$ which are the divisors, where the quotients represent the number of times each divisor is repeated in the sequence $d_1, d_2, ..., d_{m-1}, d$, and d is the greatest common divisor of $d_1, ..., d_{m-1}$. Then,

$${}^*S(a_1, ..., a_n) = \{0, d_1, d_1 + d_2, ..., d_1 + d_2 + ... + d_{m-1} + \mathbb{N}d\}.$$

Since there can not be any element less than ν_1 in the characteristic subsemigroup, $\chi_1 = \nu_1$. We construct the set $W({}^*H) - {}^*S(\chi_1)$. Let χ_2 be the smallest element in this set. Using this, let χ_3 be the smallest integer in $W({}^*H) - {}^*S(\chi_1, \chi_2)$. We continue in the same manner. Let χ_{i+1} be the smallest integer $W({}^*H) - {}^*S(\chi_1, \chi_2, ..., \chi_i)$. This procedure stops at finitely many steps. As a result, we have the Arf characters of H, which are $\chi_1, ..., \chi_l$.

Using an observation of Arf mentioned in section 3.2, we can describe an alternate method for step (2). By using the input parameterization polynomials $\varphi_1(t), ..., \varphi_n(t)$, we constructed the ring H in step (1). We have,

$$H = k + kS_{i_1} + kS_{i_2} + ... + k[[t]]S_{i_h}.$$

(2.1) We take $T_1 = \varphi_1(t)$, which has the least order (i_1) among the input polynomials $\varphi_1(t), ..., \varphi_n(t)$. We divide $\varphi_2(t), ..., \varphi_n(t)$ by $\varphi_1(t)$. In each power series $\frac{\varphi_i(t)}{\varphi_1(t)}$ (for $i = 2, ..., n$), we truncate terms of orders higher than $i_h - i_1$, and denote the resulting polynomial still by the same notation $\frac{\varphi_i(t)}{\varphi_1(t)}$. Now, we have $X_1 = \varphi_1(t), X_2 = \frac{\varphi_2(t)}{\varphi_1(t)}, ..., X_n = \frac{\varphi_n(t)}{\varphi_1(t)}$. We remove the constant terms of these polynomials, so that we have $X_1 - c_1, X_2 - c_2, ..., X_n - c_n$, where $c_1, c_2, ..., c_n$ are the constant terms of $X_1, X_2, ..., X_n$.

(2.2) We take $T_2 = X_i - c_i$, which has the least order among $X_1 - c_1, X_2 - c_2, ...X_n - c_n$. Then we apply the same procedure in step (2.1) to the polynomials $X_1 - c_1, X_2 - c_2, ..., X_n - c_n$, and obtain another n polynomials.

(2.N) We stop, when we have a polynomial of order 1 among these n polynomials, showing that $H_N = k[[t]]$.

At the end we have

$${}^*H = k + kT_1 + kT_1T_2 + ... + kT_1T_2...T_{N-1} + k[[t]]T_1T_2...T_{N-1}T_N.$$

Chapter 6

CONCLUSION

Arf theory [2] totally solves the problem of obtaining the multiplicity sequence from the local ring of the curve at the singular point without applying successive blow ups to the curve. The completion of the local ring at the singularity of the branch carries all the information necessary to obtain the multiplicity sequence, and Arf theory describes all the processes necessary to obtain the multiplicity sequence from this ring. In this way, resolution process of cusp type singularities of curves is understood thoroughly.

By depending on Arf theory, we proposed an algorithm which finds the Arf characters of a curve branch which has a parameterization in polynomials. The algorithm does not work fast. This is generally observed in the algorithms dealing with polynomials. In our algorithm, it is not possible to use effective short-cuts, which makes algorithm faster. Even only the first part of the algorithm, constructing the completion of the local ring at the singularity of the branch takes a long time, because all differences of the polynomials of the same order must be considered. The new polynomials coming from the differences must be interpreted as generators. This process is necessary to consider all the effects of the higher order terms of the parameterization polynomials. Doing so many calculations makes the algorithm slow. This kind of work generally does not have short-cuts.

We also proposed an algorithm for the solution of the problem of Frobenius in the most general case. In literature, there are only a few algorithms in the most general case. Some of these work fast in special cases, but they work very slowly for other cases. Compared with the ones applied, our algorithm works faster in most cases, since by counters we refrain from doing the same calculations several times.

Since resolution process of cusp type singularities of algebraic curves is totally solved, there is now the problem of whether the Arf theory can be adapted to surface singularities. For a curve branch having the parameterization, $x_1 = \varphi_1(t)$, ..., $\varphi_n(t)$, its blow up on U_i has parameterization in local coordinates, $X_1 = \frac{\varphi_1(t)}{\varphi_i(t)}$, ..., $X_i = \varphi_i(t)$, ..., $X_n = \frac{\varphi_n(t)}{\varphi_i(t)}$. These divisions lead to power series which are the elements of $k[[t]]$. Surface singularities have parameterization in two variables. Hence, the divisions are not as in the one variable case.

The existence of a smooth variety birationally equivalent to a given singular variety is shown by Hironaka for characteristic zero [12]. He uses successive blow ups and measures the change in the singularity by certain invariants coming from the local ring. Later Bennett has shown that Hilbert-Samuel polynomials can be used to measure the change in the singularity [4]. However, Arf theory seems to apply to any characteristic and there is hope that the idea of keeping track of the "gaps" in the local ring can lead to a further understanding of the resolution of singularities in higher dimensions.

Computers are generally ignored in the mathematical community. The main reason for this is a major misunderstanding of computers, as well as mathematics! Computers do not only crunch numbers, which practically no mathematician is interested in, but faithfully execute a given set of rules, and do this at a remarkable speed. The trick is to find those "set of rules" which will produce output to enhance our understanding of mathematics. In our case, for example, it is not important what the Arf characters of a certain branch is, but it is of paramount importance how and why those invariants come out. After all what is mathematics? Let us quote W. Thurston from a recent article [19, p. 11]:

> "*There is a real joy in doing mathematics, in learning ways of thinking that explain and organize and simplify. One can feel this joy discovering new mathematics, rediscovering old mathematics, learning a way of thinking from a person or text, or finding a new way to explain or to view an old mathematical structure.*"

We experienced this joy while analyzing Arf theory for programming, and we sincerely hope that we succeeded in conveying some of it to the reader.

Chapter 7

APPENDIX

In this appendix, we give the outputs of executions of the programs of the proposed algorithms in chapters 4 and 5.

In Example A.1 and Example A.2, the execution of the program "Frobenius" written for Maple V can be seen. In Example A.1, the largest integer that does not exist in the semigroup generated by 137, 251, 256 is found. In Example A.2, the largest integer that does not exist in the semigroup generated by 137, 251, 256, 385 is found.

Example A.1.

```
> Frobenius();

                Please, enter relatively prime integers as input

                (separated by commas, ending with a semicolon):

137,251,256;
         The largest integer that does not exist in the semigroup generated

                                 by these integers is:

                                        4948
```

Example A.2.

```
> Frobenius();
                    Please, enter relatively prime integers as input

                    (separated by commas, ending with a semicolon):


137,251,256,385;
bytes used=1000336, alloc=786288, time=3.02
bytes used=2000664, alloc=982860, time=7.22
bytes used=3001808, alloc=1179432, time=11.50
      The largest integer that does not exist in the semigroup generated

                              by these integers is:

                                      3282
```

In Example A.3 and Example A.4, the execution of the program "Arf" written for Maple V can be seen. In Example A.3, Arf characters of the curve branch given in Example 3.2.14 is found. $(X = t^4, Y = t^9, Z = t^{15}.)$ In Example A.4, Arf characters of the curve branch given in Example 3.2.15 is found. $(X = t^6, Y = t^8 + t^{11}, Z = t^{14}.)$

Example A.3.

```
> Arf();

                    Please, input parameterization polynomials in t

                    (separated by commas, ending with a semicolon):


t^4,t^9,t^15;
                              The Arf characters are:

                                      [4, 9]
```

Example A.4.

```
> Arf();
                              Please, input parameterization polynomials in t

                              (separated by commas, ending with a semicolon):

t^6,t^8+t^11,t^14;
                                        The Arf characters are:

                                              [6, 8, 15]
```

REFERENCES

[1] Abhyankar, S., *Algebraic Geometry For Scientists and Engineers*, American Mathematical Society, 1990.

[2] Arf, C., "Une interprétation algébrique de la suite des ordres de multiplicité d'une branche algébrique", *Proc. London Math. Soc.* (2) **50** (1949), 256-287.

[3] Atiyah, M. F., Macdonald, I. G., *Introduction to Commutative Algebra*, Addison-Wesley, 1969.

[4] Bennett, B. M., "On the characteristic functions of a local ring", *Ann. of Math.* **91** (1970), 25-87.

[5] Brauer, A. T., Shockley, J. E., "On a problem of Frobenius", *J. reine angew. Math.* **211** (1962), 215-220.

[6] Brieskorn, E., Horst, K., *Plane Algebraic Curves*, Birkhäuser, 1986.

[7] Cox, D., Little, J., O'Shea, D., *Ideals, Varieties, and Algorithms*, Springer-Verlag, 1992.

[8] Du Val, P., "The Jacobian algorithm and the multiplicity sequence of an algebraic branch", *Rev. Faculté Sci. Univ. Istanbul* (Série A) **7** (1942), 107-112.

[9] Harris, J., *Algebraic Geometry: A First Course*, Springer-Verlag, 1992.

[10] Hartshorne, R., *Algebraic Geometry*, Springer-Verlag, 1977

[11] Heap, B. R., Lynn, M. S., "A graph theoretic algorithm for the solution of a linear diophantine equation", *Numer. Math.* **6** (1964), 110-141.

[12] Hironaka, H., "Resolution of singularities of an algebraic variety over a field of characteristic zero", *Ann. of Math..* (1) **79** (1964), 109-326.

[13] Lang, S., *Algebra*, Addison-Wesley, 1984.

[14] Lewin, M., "An algorithm for a solution of a problem of Frobenius", *J. reine angew. Math* **276** (1975), 68-82.

[15] Rödseth, Ö. J., "On a linear diophantine problem of Frobenius", *J. reine angew. Math* **301** (1978), 171-178.

[16] Selmer, E. S., "On the linear diophantine problem of Frobenius", *J. reine angew. Math* **293/294** (1977), 1-17.

[17] Semple, J. G., "Singularities of space algebraic curves", *Proc.London Math.Soc.* (2) **44** (1938), 149-174.

[18] Sertöz, S., Özlük, A., "On a diophantine problem of Frobenius", *Bull. Tech. Univ. Istanbul* **39** (1986), 41-51.

[19] Thurston, W., "On proof and progress in mathematics", *Bull. Amer. Math. Soc.* (30) **2** (1994), 161-177.