Real Root Counting and Derivational Complexity of Term Rewrite Systems

master thesis in computer science

by

Philipp Wirtenberger

submitted to the Faculty of Mathematics, Computer Science and Physics of the University of Innsbruck

in partial fulfillment of the requirements for the degree of Master of Science

supervisor: Assoz. Prof. Dr. Georg Moser,
Institute of Computer Science

Innsbruck, 8 January 2020

universität
innsbruck

Master Thesis

Real Root Counting and Derivational Complexity of Term Rewrite Systems

Philipp Wirtenberger (01315233)
Philipp.Wirtenberger@student.uibk.ac.at

8 January 2020

Supervisor: Assoz. Prof. Dr. Georg Moser

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht.
Die vorliegende Arbeit wurde bisher in gleicher oder ähnlicher Form noch nicht als Magister-/Master-/Diplomarbeit/Dissertation eingereicht.

Datum	Unterschrift

Abstract

For the process of automatically bounding the derivational complexity of term rewrite systems by means of matrix interpretations it is helpful to prove that the absolute values of the eigenvalues of the maximum matrix are smaller equal one. Constricting the spectral radius of a non-negative real matrix to values smaller equal one can be expressed by the constraint that its characteristic polynomial has no real roots greater than one. In this thesis we discuss the applicability of various theorems related to real root counting to the synthesis of such a constraint for symbolic polynomials.

Acknowledgments

A heartfelt thank you to my supervisor for his incredible time-investment and close mentorship while the thesis was still in its research phase, and to all other colleagues in the lab whose feedback has sparked thought-provoking discussions.

I am forever grateful for my family's steadfast support.

Could not have done it without you.

Contents

1 Introduction

Automated and semi-automated complexity analysis for the runtime behaviour of computer programs has applications in a variety of fields where the predictability of the execution time of a routine is of high significance. And as long as the analysis process is fully automated it is predestined to also become a valuable tool in the everyday programmer's tool belt. Automated construction of sound mathematical *proofs* of runtime complexity is especially valuable in application domains where guaranteed runtime behaviour is mission critical. And while there is a wide gamut of practical applications promoting active research, proving such bounds in a fully automatic manner is a hard task to achieve. There are several automated tools that are geared towards proving termination rather than runtime behaviour, and some of these are accompanied by a certifier. And although established techniques to prove termination properties lend themselves well to the extension to a method proving bounds on runtime complexity, in practice such complexity-bounding methods tend to be much less strong than their termination-proving counterparts they are based on.

The derivational complexity of term rewrite systems will serve us as an abstraction for the concept of runtime complexity of computer programs. Term rewrite systems are quite akin to functional programs and there have been attempts to adapt existing proving techniques to functional programs. As an alternative to derivational complexity, there is also the notion of *runtime* complexity of a term rewrite system, closer modelling the runtime complexity of the functional program the rewrite system imitates. In this thesis, however, we are only concerned with derivational complexity of term rewrite systems. A transformation is outlined in a recent paper by Fuhs [37]. There is also a research project to adapt the methods to the intricacies of imperative JVM bytecode. In order to put the efficacy of termination provers to the test, term rewrite systems from across the literature have been collected and consolidated into what is known as the *Termination Problems Database*[1]. There are annual contests that pitch the latest proving techniques against this set of problems, with competitors from several universities. This is precisely the set of term rewrite systems we will use for benchmarking purposes in this thesis.

We want to remark that this work is concerned with the proof of upper bounds on the complexity of term rewrite systems. While the additional knowledge of lower bounds greatly helps giving precise predictions (or rather, guarantees!) on runtime behaviour, upper bounds are what is paramount to the discovery and subsequent elimination of bad runtime behaviour that could prove disastrous in certain application domains. Having reliable, proven upper bounds on runtime complexity can not only save time and complications but is sometimes considered a necessity. The proof methods we explore in

[1] http://termination-portal.org/wiki/TPDB

this thesis are based on theorems that are helpful first and foremost for the derivation of upper bounds. When starting to research, our main goal was to find bounds for as many term rewrite systems as possible and not to prove the tightest bounds. If tightening the bounds or proving lower instead of upper bounds is desired one can achieve this for example by combining the methods laid out in this document with other techniques.

The goal of this thesis is to find new methods which infer upper limits on the derivational complexity of term rewrite systems. The theorems of Chapter 2 establish that by building on existing matrix-interpretation based techniques we can achieve this goal by finding ways to limit the value of the real roots of characteristic polynomials to at most 1. Throughout this thesis we will thus explore different ways to constrain the real roots of univariate polynomials over the reals to a desired interval. For each freshly introduced method to limit the value of the roots we will discuss the construction of a logical constraint that can be integrated into existing complexity analysers. The first part of the thesis will be about the theoretical foundations that establish why we try to limit the non-negative roots of characteristic polynomials to the interval $[0; 1]$, and how this allows us to infer bounds on the derivational complexity of term rewrite systems. A survey on bounds for the value of the real roots of univariate polynomials, accompanied by an analysis of how those bounds can be transformed into logical formulae and how suited they are for the concrete bound of 1 we are interested in, will then constitute the second part of the thesis. In the third part we dissect root isolation algorithms and discuss whether we can repurpose them for our own use case. We also contrast our new constraints to related results from the literature. Finally, some of the new logical constraints are put to use and their performance is evaluated.

In the end we will find that the constraints we have built are interesting from a theoretical standpoint but not very practical. We attribute this to the fact that the theorems we surveyed are not specifically tailored to symbolic polynomials and the synthesis of logical constraints.

1.1 Related Work

This thesis was inspired by previous work on the topic of automated proofs of derivational complexity bounds for term rewrite systems. We especially want to bring the reader's attention to Neurauter et al. [63], Middeldorp et al. [60], Thiemann & Yamada [70] and Divasón et al. [30, 31].

1.2 Preliminaries

We assume familiarity with the concepts of sets, real and complex fields, sequences, functions, function application, composition of functions. $\mathbb{N}$ denotes the natural numbers *including zero*. We write $\mathbb{N} \setminus \{0\}$ to denote the natural numbers *excluding zero*. Everything but matrices is zero-indexed.

Definition 1.1. Let f be a function of a real variable. The sequence of *derivative functions*, short *derivatives*, $f', f'', f''', \ldots, f^{(n)}$ of f is obtained by repeated application

of the function

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Definition 1.2. Let $z = a + b\mathrm{i}$ be a complex number. The *absolute value* of z is

$$|z| = \sqrt{a^2 + b^2}$$

Definition 1.3. Let s be a finite real numbered sequence and $(a_i)_{i<n}$ be the sequence consisting of the non-zero elements of s. The *number of sign variations* in s is defined as

$$\max(0, \frac{a_0 a_1}{-|a_0 a_1|}) + \max(0, \frac{a_1 a_2}{-|a_1 a_2|}) + \ldots + \max(0, \frac{a_{n-2} a_{n-1}}{-|a_{n-2} a_{n-1}|})$$

1.2.1 Linear Algebra

Polynomials

Definition 1.4. A *univariate monomial* over $\mathbb{R}$ is the product of a real number and a single variable with a non-negative integer exponent.

Unless stated otherwise we refer to univariate monomials over the reals simply as monomials.

Definition 1.5. The exponent of a monomial is its *degree*.

Definition 1.6. A *univariate polynomial* over $\mathbb{R}$ is the sum of an arbitrary but strictly positive number of univariate monomials.

Unless stated otherwise we refer to univariate polynomials over the reals simply as polynomials.

Definition 1.7. The *degree of a polynomial* is the highest degree of its monomials.

Definition 1.8. Let $n \in \mathbb{N}$ and f be a univariate polynomial of degree $\geq n$. The sum of monomials of degree n is itself a monomial and called the *term of degree* n or *monomial of degree* n of f.

Remark. When referring to the monomials of a polynomial we always refer to the monomials obtained when summing all monomials of the same degree (i.e. simplifying as much as possible). This is also reflected in the functional notation we use to declare a polynomial:

Notational Convention. A polynomial f in the indeterminate x of degree n can be written as

$$f(x) = \sum_{0}^{n} a_i x^i$$

with $a_0, \ldots, a_n \in \mathbb{R}$. The term of f of degree j is thus $a_j x^j$.

Definition 1.9. Let f be a polynomial of degree n. The term of degree n is called the *leading term* of the polynomial.

Definition 1.10. A *normalized* univariate polynomial is a univariate polynomial whose leading term has no factor other than the indeterminate.

Notational Convention.

$$\mathbb{R}[x] = \{ \sum_{0}^{n} a_i x^i \mid n \in \mathbb{N} \text{ and } a_0, \ldots, a_n \in \mathbb{R} \}$$
$$\mathbb{Z}[x] = \{ \sum_{0}^{n} a_i x^i \mid n \in \mathbb{N} \text{ and } a_0, \ldots, a_n \in \mathbb{Z} \}$$

Definition 1.11. Let f be a univariate polynomial over the reals. Let $(a_i)_{i \leq n}$ be the sequence consisting of the non-zero coefficients of f. Then

$$\max(0, \frac{a_0 a_1}{-|a_0 a_1|}) + \max(0, \frac{a_1 a_2}{-|a_1 a_2|}) + \ldots + \max(0, \frac{a_{n-1} a_n}{-|a_{n-1} a_n|})$$

is the *number of sign variations* in f.

Definition 1.12. Let f be a univariate polynomial over the reals. The elements of the set $\{ x \mid f(x) = 0 \}$ are called the *roots* of the polynomial f.

Definition 1.13. A univariate polynomial over the reals which is not divisible by the square of any non-constant univariate polynomial over the reals is called *squarefree*.

Definition 1.14. Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial over the reals and let $r_0, \ldots, r_{n-1}$ be its roots. The *discriminant* Δ_n of f is

$$a_n^{2(n-1)} \prod_{i=0}^{n-2} \prod_{j=i+1}^{n-1} (r_i - r_j)^2$$

Definition 1.15. Let f be a polynomial over the integers. The greatest common divisor of the coefficients of f is called the *content* of f. The *primitive part* of f is f divided by its content. We call f *primitive* when the content of f is 1.

Definition 1.16. Let f be a polynomial over the rationals and let n be the lowest common denominator of the coefficients of f. The content of f is the content of the integer polynomial nf divided by n. The *primitive part* of f is f divided by its content. We call f *primitive* when the content of f is 1.

Matrices

Definition 1.17. Let $m, n \in \mathbb{N} \backslash \{0\}$. A $m \times n$ *matrix* over $\mathbb{R}$ is a function $\mathbf{A} : \{1, \ldots, m\} \times \{1, \ldots, n\} \to \mathbb{R}$.

Remark. Let $\mathbf{A}$ be a $m \times n$ matrix over the reals. We write $\mathbf{A}_{i,j}$ or $\mathbf{A}_{ij}$ if non-ambiguous to denote the real number $\mathbf{A}(i, j)$ for $1 \leq i \leq m$ and $1 \leq j \leq n$. We call $\mathbf{A}_{ij}$ the *matrix entry* or *coefficient* at row i and column j.

Notational Convention. We write $\mathbb{R}^{m\times n}$ to denote the set of all real matrices with m rows and n columns.

Definition 1.18. Let $n \in \mathbb{N} \setminus \{0\}$. A $n\times n$ matrix is called *square*.

Notational Convention. We write $\mathbb{R}^{n\times n}$ to denote the set of all square matrices over the reals with n rows and n columns.

The number of rows and columns is called the *order* or *dimension* of the matrix. For square matrices the order is sometimes for sake of brevity represented by a single number.

Matrix addition and multiplication with a scalar are to be calculated in a componentwise fashion.

Definition 1.19.

$$+ : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$$
$$(\mathbf{A}+\mathbf{B})_{ij} \mapsto \mathbf{A}_{ij} + \mathbf{B}_{ij} \quad \text{for } 1 \le i \le m, 1 \le j \le n.$$

Definition 1.20.

$$\cdot : \mathbb{R} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$$
$$(c \cdot \mathbf{A})_{ij} = c \cdot \mathbf{A}_{ij} \quad \text{for } 1 \le i \le m, 1 \le j \le n.$$

Example 1.21. Let $\mathbf{A} = \begin{pmatrix} 1 & 6 \\ 4 & 3 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 4 & 1 \\ 7 & 6 \end{pmatrix}$.
Then $1/2 \cdot \mathbf{A} = \begin{pmatrix} 0.5 & 3 \\ 2 & 1.5 \end{pmatrix}$ and $\mathbf{A} + \mathbf{B} = \begin{pmatrix} 5 & 7 \\ 11 & 9 \end{pmatrix}$.

Multiplication of two matrices is a bit more complicated than just multiplying the matrix elements in a componentwise manner. The exact process is outlined in Definition 1.22.

Definition 1.22.

$$\cdot : \mathbb{R}^{n_0\times n_1} \times \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^{n_0\times n_2}$$
$$(\mathbf{A} \cdot \mathbf{B})_{ij} = \sum_{k=1}^{n_1} \mathbf{A}_{ik} \cdot \mathbf{B}_{kj} \quad \text{for } 1 \le i \le n_0, 1 \le j \le n_2.$$

We give an example to make the process clear.

Example 1.23. We continue Example 1.21.
When multiplying the matrices, we get $\mathbf{A} \cdot \mathbf{B} = \begin{pmatrix} 46 & 37 \\ 37 & 22 \end{pmatrix}$.

Definition 1.24. A matrix is *non-negative* if all its entries are non-negative.

The component-wise maximum matrix of a set of $m\times n$-ordered matrices is the matrix that is composed of the component-wise maximum of the entries of the matrices. In other words, for each matrix entry, we select the highest value present in the entries across the matrices in the set.

Definition 1.25. Let $\mathbf{A} \subseteq \mathbb{R}^{m\times n}$. The *(component-wise) maximum matrix* $M_{\mathbf{A}}$ of $\mathbf{A}$ is

$$(M_{\mathbf{A}})_{ij} = \max\{\, (\mathbf{A}_k)_{ij} \mid \mathbf{A}_k \in \mathbf{A} \,\}$$

Example 1.26. The maximum matrix of $\{ \begin{pmatrix} 1 & 3 \\ 4 & 7 \end{pmatrix}, \begin{pmatrix} 5 & 0 \\ 2 & 3 \end{pmatrix}, \begin{pmatrix} 4 & 3 \\ 0 & 2 \end{pmatrix} \}$ is $\begin{pmatrix} 5 & 3 \\ 4 & 7 \end{pmatrix}$.

Definition 1.27. The *Kronecker delta* function is defined as:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 1.28. The $n\times n$ matrix whose entries are corresponding to the Kronecker delta function

$$(\mathbf{I}_n)_{ij} = \delta_{ij}$$

is called the identity matrix of order n.

Definition 1.29. Let the function $\mathbf{A}$ be a $n\times n$ matrix over $\mathbb{R}$, and let $n > 1, 1 \leq k \leq n, 1 \leq l \leq n$. The *minor* of $\mathbf{A}$ is the determinant (cf. Definition 1.30) of the function $\mathbf{A}' : (\{1,\ldots,n\}^2 \to \mathbb{R}) \times \{1,\ldots,n\}^2 \to (\{1,\ldots,n-1\}^2 \to \mathbb{R})$ evaluated for $\mathbf{A}$, k and l s.t.

$$\operatorname{minor}(\mathbf{A}, k, l) = |\mathbf{A}'(\mathbf{A}, k, l)|,$$

$$\mathbf{A}'(\mathbf{A}, k, l, i, j) = \begin{cases} \mathbf{A}_{i+1,j+1} & \text{if } i \geq k \text{ and } j \geq l, \\ \mathbf{A}_{i+1,j} & \text{if } i \geq k \text{ and } j < l, \\ \mathbf{A}_{i,j+1} & \text{if } i < k \text{ and } j \geq l, \\ \mathbf{A}_{ij} & \text{otherwise.} \end{cases}$$

Definition 1.30. The *determinant* of a square matrix $\mathbf{A}$ of order n with coefficients in $\mathbb{R}$ is given by the recursive formula

$$\det(\mathbf{A}) = |\mathbf{A}| = \begin{cases} \mathbf{A}_{1,1} & \text{if } n = 1, \\ \sum_{j=1}^{n} (-1)^{1+j} \mathbf{A}_{1j} \cdot \operatorname{minor}(\mathbf{A}, 1, j) & \text{otherwise.} \end{cases}$$

Example 1.31. Continuing from Example 1.21. $\mathbf{A}$ is a square matrix of order 2 so there will be one recursive step in the calculation of the determinant. So we first calculate $\operatorname{minor}(\mathbf{A}, 1, 1) = |(3)| = 3$ and $\operatorname{minor}(\mathbf{A}, 1, 2) = |(4)| = 4$. Now with the recursive step taken care of, the remainder of the calculation looks as follows: $|\mathbf{A}| = \operatorname{minor}(\mathbf{A}, 1, 1) - 6 \cdot \operatorname{minor}(\mathbf{A}, 1, 2) = -21$. Same procedure with matrix $\mathbf{B}$: $\operatorname{minor}(\mathbf{B}, 1, 1) = 6$, $\operatorname{minor}(\mathbf{B}, 1, 2) = 7$ and thus $|\mathbf{B}| = 4 \cdot \operatorname{minor}(\mathbf{B}, 1, 1) - \operatorname{minor}(\mathbf{B}, 1, 2) = 17$.

Definition 1.32. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$. The *characteristic polynomial* of $\mathbf{A}$ is $\chi_{\mathbf{A}}(\lambda) = |\lambda \mathbf{I}_n - \mathbf{A}|$.

Remark. The reason we prefer to define the characteristic polynomial as $|\lambda\mathbf{I}_n - \mathbf{A}|$ over $|\mathbf{A} - \lambda\mathbf{I}_n|$ is that in the former case the characteristic polynomial is *normalized* which may make subsequent calculations easier if done by hand.

Definition 1.33. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\chi_{\mathbf{A}}$ its characteristic polynomial. The *eigenvalues* of $\mathbf{A}$ are the, not necessarily real, roots of $\chi_{\mathbf{A}}$.

Definition 1.34. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$. The *spectral radius* $\rho(\mathbf{A})$ of $\mathbf{A}$ is the maximum of the absolute values of its eigenvalues.

Definition 1.35. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$. Let λ_i be an eigenvalue of $\mathbf{A}$. Then we have

$$\chi_{\mathbf{A}}(\lambda) = (\lambda - \lambda_i)^{m_i} f(\lambda)$$

for some polynomial f with $f(\lambda_i) \neq 0$. The power m_i is the *(algebraic) multiplicity* of the eigenvalue λ_i.

Remark. If the multiplicity of an eigenvalue is greater than one, we call it a *multiple root.*

Definition 1.36. The $n\times n$ matrix whose entries correspond to the function

$$(\mathbf{J}_{\lambda,n})_{ij} = \begin{cases} \lambda & \text{if } i = j, \\ 1 & \text{if } i = j - 1, \\ 0 & \text{otherwise.} \end{cases}$$

for some constant λ is called a *Jordan block.*

Definition 1.37. A $n\times n$ matrix whose entries correspond to the function

$$(\mathbf{A})_{ij} = \begin{cases} (\mathbf{J}_{\lambda_k,n_k})_{i-l,j-l} & \text{if } l < i,j \leq l + n_k \text{ for some } k \in \mathbb{N} \text{ where } l = \sum_{v=0}^{k-1} n_v, \\ 0 & \text{otherwise.} \end{cases}$$

for some sequence of Jordan blocks $\mathbf{J}_{\lambda_0,n_0}, \ldots, \mathbf{J}_{\lambda_m,n_m}$ where $\sum_{i=0}^{m} n_i = n$ is said to be in *Jordan Normal Form.*

Remark. The Jordan Normal Form of a square matrix is unique up to the order of Jordan blocks.

1.2.2 Term Rewriting

A comprehensive list of common notation used in term rewriting is presented in [24].

We will introduce all the concepts of term rewriting that we are going to use throughout the thesis. For a proper introduction to the topic of term rewriting we point to [10].

Definition 1.38. We declare a countably infinite set of symbols as *variables.*

Definition 1.39. A *function symbol* is a symbol different from a variable, with a corresponding arity. Nullary function symbols are called *constants.* A set of function symbols is called *signature.*

From a given set of variables and a signature we can construct more complicated expressions: terms, the basic building blocks of term rewriting.

Definition 1.40. Exhaustive composition of variables and function symbols results in the set of *terms* over the signature and a countably infinite set of variables.

Later on we will need a conception that tells us how complex a certain term is compared to other terms. To this end, we simply count the number of of elements the term is composed of, i.e. the number of function symbols and variables. We call this the size of the term, symbolized by $|\cdot|$.

Definition 1.41. Let t be a term. Counting multiples, the *total number* of function symbols and variables is the *size* $|t|$ of the term.

Example 1.42. Let $t_0 = g(c), t_1 = x, t_2 = f(g(x), f(c, x))$ be terms over an arbitrary fitting signature, with x a variable. Then the size of the terms is

- $|t_0| = 2$
- $|t_1| = 1$
- $|t_2| = 6$

A term rewrite system, abbreviated as *TRS*, consists of rewrite rules that operate on the terms of the TRS. Simply put, a rewrite rule transforms one term into another. Rewrite rules are defined as pairs of terms. Some restrictions apply to the `gestalt` of these pairs in order to capture the concept of rewriting. We do not want to transform a variable into some more restricted term, and likewise it makes no sense if after rewriting a term new variables appear in the reduct (i.e. the reduced term).

Definition 1.43. A *rewrite rule* is an ordered pair of terms (t_1, t_2) with the following properties:

- t_1 is not a variable
- All variables that are part of t_2 are also contained in t_1

To ease readability, we write $t_1 \to t_2$ for the rewrite rule (t_1, t_2).

Definition 1.44. A *term rewrite system* is a signature paired with a set of rewrite rules over the terms induced by the signature.

Definition 1.45. Let S_1 and S_2 be arbitrary sets. A *dyadic relation* on S_1 and S_2 is any subset of the set of all ordered pairs (s_1, s_2) where $s_1 \in S_1$ and $s_2 \in S_2$.

Definition 1.46. A dyadic relation on a set S is called *well-founded* if there exists no infinite sequence of elements $s_1, s_2, s_3, \ldots$ of S for which all pairs (s_i, s_{i+1}) are contained in the relation.

Definition 1.47. We choose an unused constant whose only purpose is be to be replaced by a term at a later point; and we call it a *hole*. A term that has a single "hole" is called a *one-hole context*.[2]

We write $s[t]$ to denote the term that emerges when the hole of a context s is replaced by a term t. Similar to *closure under substitution* we will define the notion of *closure under context*. Both together will comprise the necessary attributes for a dyadic relation on terms to be considered a *rewrite relation*. If our rewrite rules are closed under context this grants us the ability to apply them not only to the full term we are working with as a whole but also to the various subterms it is composed of. See Example 1.57 for a demonstration of this.

Definition 1.48. Let $\circ$ be a dyadic relation on terms. We say that $\circ$ is *closed under context* if $t_1 \circ t_2$ implies $s[t_1] \circ s[t_2]$ for any one-hole context s.

Let us see an example:

Example 1.49. Let $\circ = \{(f(f(c)), g(c)), (c, g(c))\}$ be a dyadic relation on terms over an arbitrary fitting signature. The relation is *not* closed under context because e.g. $c \circ g(c) \not\Rightarrow f(f(c)) \circ f(f(g(c)))$. Let $\bullet = \{(f(f(c)), g(c)), (c, g(c))\} \cup \{(t_1[t_2], t_1[g(c)]) \mid t_1$ is one-hole context and $t_2 = f(f(c))$ or $t_2 = c\}$ be the smallest extension of $\circ$ that is closed under context. And indeed, we see that $c \bullet g(c) \Rightarrow f(f(c)) \bullet f(f(g(c)))$.

Definition 1.50. Let the *substitution* σ be a mapping from variables to terms. We can apply the substitution to terms, denoted by $t\sigma$ for a term t, by recursively applying it to the arguments of function symbols, substituting variables where possible.[3]

Definition 1.51. Let $\circ$ be a dyadic relation on terms. We say that $\circ$ is *closed under substitution* if $t_1 \circ t_2$ implies $t_1\sigma \circ t_2\sigma$ for any substitution σ.

The closure properties we have just introduced will now allow us to capture the concept of rewriting in a sensible way: thanks to closure under context we can operate on subterms. And due to closure under substitution we can use variables as placeholders for arbitrary terms. Combined, we can use the rewrite rules of a term rewrite system as one would intuitively do. We will see a bit later how to chain steps applying rewrite rules to form *derivations* that transform a starting term to a possibly different term.

Definition 1.52. We call a dyadic relation on terms a *rewrite relation* if it is both closed under substitution and closed under context.

Example 1.53. Let x be a variable and let $S = \{(g(x), x), (g(c), f(c))\}$ be a set of rewrite rules over an arbitrary fitting signature. Then the dyadic relation $\circ_S = \{(s[g(x)]\sigma, s[x]\sigma) \mid \sigma : x \mapsto t$ for some term t and s is a one-hole context$\} \cup \{(s[g(c)], s[f(c)]) \mid s$ is a one-hole context$\}$ is the smallest extension of S that is closed under context and closed under substitution and thus a *rewrite relation*.

[2] We want to stress that our very crude definition of context as a "term with a hole" is, albeit common in literature, perhaps not the ideal formal definition. For a more modern definition of contexts in both term rewriting and lambda calculus we suggest [51].

[3] In the literature, sometimes a prefix notation is used to describe the application of substitutions to terms.

Definition 1.54. Let $\circ$ be a rewrite relation. A term t_0 is said to be *terminating w.r.t.* $\circ$ if there is no infinite sequence $t_0, t_1, t_2, \ldots$ of terms for which we have $t_n \circ t_{n+1}$ for every natural number n.

When musing about rewrite rules that can be applied to subterms, we are inevitably going to encounter a situation in which we have to make the choice on which applicable subterm to use the rule. Until now, there was no need to specify which subterm of a term we are concerned with. We will use rewrite relations, which are inherently closed under context, in practice shortly and to fully capture the intuitive notion of rewriting terms we have yet to model this mechanism of choice. We will first find a way to enumerate the options such a decision point offers. Thus, we will now define a simple numbering system that allows us to point to a specific subterm of a term. The number we assign a subterm we call its *position* in the enclosing term. The sequence of numbers we associate a subterm with is built as follows. Picture the term in question as a tree: the outermost function symbol of the term becomes the root, the arguments of a function symbol are its children in the tree, and variables and constants have no child nodes. Then we traverse this tree from top to bottom until arriving at the node of the desired subterm, adding to the sequence of numbers for every branching decision (unary and n-ary) on the way. Defining this formally, we arrive at Definition 1.55.

Definition 1.55. Let p be a finite sequence of natural numbers. The subterm of a term t at *position* p is denoted $t|_p$ and defined by induction on p:

- $t|_\varepsilon = t$
- $t|_{ip} = t_i|_p$ if $t = f(t_1, \ldots, t_n)$ *with* f *some* n*-ary function symbol and* $1 \leq i \leq n$
- *otherwise*: p not a valid position

Now that we are fully prepared to select which subterm we want to reduce with a given rewrite rule we formally define the process of repeated application of rewrite rules as a sequence of rewrite steps we call a *derivation*. The derivation will describe the rule and the subterm involved in each rewrite step. It thus makes sense to model a rewrite step as a tuple consisting of an ordered pair of starting and resulting term taken from the rewrite relation, and of a position, together indicating the rewrite rule and the subterm we chose to use. This has the advantage that we immediately see the effect of the rewrite operation. This holds especially true for a common notational convention we will introduce: Following the idea set forth in Definition 1.43 we use an infix arrow symbol to represent the relation, distinguishing individual rewrite relations by a subscript where necessary. For a single step, we superimpose the position p as a superscript. So for a rewrite relation r_1 and a single rewrite step from a term t_1 to a term t_2 at position p we write $t_1 \to_{r_1}^{p} t_2$.

Remark. As long as we introduce no ambiguity, we commonly drop the position from the rewrite step tuple. If there *is* ambiguity we assume the rewrite rule to be applied at the leftmost-outermost suitable part of the term.

Definition 1.56. Let R be a rewrite relation, $(t_0, t_1), (t_1, t_2), \ldots, (t_{n-1}, t_n) \in R^n$. We call this n-tuple a *n-step derivation* over R.

Example 1.57. We continue Example 1.53. Starting with the term $g(f(f(g(g(c)))))$ we obtain from the rewrite relation the following derivations:

$$g(f(f(g(g(c))))) \circ_S g(f(f(g(c)))) \circ_S g(f(f(c))) \circ_S f(f(c))$$
$$g(f(f(g(g(c))))) \circ_S g(f(f(g(c)))) \circ_S g(f(f(f(c)))) \circ_S f(f(f(c)))$$
$$g(f(f(g(g(c))))) \circ_S g(f(f(g(c)))) \circ_S f(f(g(c))) \circ_S f(f(c))$$
$$g(f(f(g(g(c))))) \circ_S g(f(f(g(c)))) \circ_S f(f(g(c))) \circ_S f(f(f(c)))$$
$$g(f(f(g(g(c))))) \circ_S g(f(f(g(f(c)))))) \circ_S g(f(f(f(c)))) \circ_S f(f(f(c)))$$
$$g(f(f(g(g(c))))) \circ_S g(f(f(g(f(c)))))) \circ_S f(f(g(f(c)))) \circ_S f(f(f(c)))$$
$$g(f(f(g(g(c))))) \circ_S f(f(g(g(c)))) \circ_S f(f(g(c))) \circ_S f(f(c))$$
$$g(f(f(g(g(c))))) \circ_S f(f(g(g(c)))) \circ_S f(f(g(c))) \circ_S f(f(f(c)))$$
$$g(f(f(g(g(c))))) \circ_S f(f(g(g(c)))) \circ_S f(f(g(f(c)))) \circ_S f(f(f(c)))$$

As a measure of complexity we will study the relation of the maximum number of possible rewrite steps to the size of the starting term. In accordance to this goal, we define the derivation length of a term to be the maximum number of steps a derivation can have when starting from said term. The measure of derivational complexity will then be the maximum of the derivation lengths of all terms of the signature up to a certain size.

Definition 1.58. Let t_0 be a terminating term and R a rewrite relation. The *derivation length of* t_0 is $\mathbf{max}\{n \mid (t_0, t_1), (t_1, t_2), \ldots, (t_{n-1}, t_n) \in R^n$ for some terms $t_1, \ldots, t_n\}$. If there is no term t_1 such that $(t_0, t_1) \in R$ then we set the derivation length of t_0 to zero.

Example 1.59. Let us consider the derivations from Example 1.57, starting at the term $g(f(f(g(g(c)))))$. The derivations are all equally long. The derivations have three steps and thus the derivation length of $g(f(f(g(g(c)))))$ is considered to be 3.

Definition 1.60. Let R be rewrite relation over an arbitrary fitting signature. The derivational complexity for terms up to size n of the term rewrite system induced by the signature and R is defined as $\mathbf{max}\{$derivation length of $t \mid |t| \leq n$ for some term $t\}$.

Example 1.61. Consider the rewrite rules from Example 1.53 but this time over the concrete signature $\{ (f, 1), (g, 1), (c, 0) \}$. Let n be the upper limit on the size of the terms we take into consideration. If $n = 1$, our only term is c, with a derivation length of 0. For $n = 2$, we have the set of terms $\{ f(c), g(c), c \}$ with a maximum derivation length 1. In fact, the derivational complexity for this term rewrite system can simply be modelled by the predecessor function, for terms up to a size n the maximal derivation length is $n - 1$.

2 On the Relation of the Spectral Radius and the Derivational Complexity of Term Rewrite Systems

Matrix interpretations as a mathematical construct were initially designed to prove a term rewrite system to be terminating (cf. [46, 33]). But just as other tools for proving termination of term rewrite systems, matrix interpretations too can be used to conduct automated complexity analysis ([62]). If we manage to bound the growth rate of the entries of the component-wise maximum matrix of all matrices of a matrix interpretation when it is multiplied with itself, then we can infer bounds for the derivational complexity of the corresponding term rewrite system.

Especially helpful in this context is the Perron-Frobenius Theorem.

Theorem 2.1 (Perron-Frobenius (weak form), [36])**.** *Let* $\mathbf{A} \in \mathbb{R}_0^{n\times n}$*. Then* $\rho(\mathbf{A})$ *is an eigenvalue of* $\mathbf{A}$*.*

Theorem 2.2 ([31])**.** *Let* k *be a natural number and* $\mathbf{A} \in \mathbb{C}^{n\times n}$*. The entries of* $\mathbf{A}^k$ *are polynomially bounded in* k *if and only if the spectral radius of* $\mathbf{A}$ *is smaller equal one.*

Theorem 2.3 ([31])**.** *Let* k *and* d *be natural numbers and* $\mathbf{A} \in \mathbb{C}^{n\times n}$*. The entries of* $\mathbf{A}^k$ *are bounded by* $\mathcal{O}(k^d)$ *if and only if the spectral radius of* $\mathbf{A}$ *is smaller equal one and for all eigenvalues of* $\mathbf{A}$ *of absolute value equal one the dimension of their corresponding Jordan blocks of* $\mathbf{A}$ *is smaller equal* $d + 1$*.*

Theorem 2.4 ([63])**.** *Let* $\mathbf{A} \in \mathbb{R}_0^{n\times n}$ *be the component-wise maximum matrix of all matrices of a matrix interpretation compatible to a term rewrite system* $\mathcal{R}$*. If* $\rho(\mathbf{A}) \leq 1$*, then* $dc_{\mathcal{R}}(k) \in \mathcal{O}(k^{d+1})$ *where* $d := max_\lambda(0, m_\lambda) - 1$ *and* λ *are the eigenvalues with absolute value exactly one.*

This is the theoretical basis that motivates us to search for criteria which assert $\rho(\mathbf{A}) \leq 1$ for a strictly positive real matrix $\mathbf{A}$.

3 Upper Bounds on the Eigenvalues of Square Matrices

A publication of Bendixson [13] which spawned a response letter from Hirsch [45] detailing some generalizations is of interest to us. In his letter, Hirsch proposed a generalization of Bendixson's *Théorème I* that includes the following result:

Theorem 3.1. *Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a matrix with exclusively real eigenvalues. Further, let i be a non-zero natural number smaller equal n and let m be the maximum of the absolute values of the row entries $\{ \mathbf{A}_{ij} \mid 1 \leq j \leq n \}$. Then the absolute value of the i-th eigenvalue of $\mathbf{A}$ is smaller equal $m \cdot n$.*

Hence if the absolute values of all matrix entries are smaller equal $\frac{1}{n}$ then there is no eigenvalue of absolute value greater one.

Geršgorin [41] adds the following criterion to this.

Theorem 3.2. *Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a matrix with exclusively real eigenvalues. Further, let i be a non-zero natural number smaller equal n and let $r_i = (\sum_{j=1}^{n} |\mathbf{A}_{ij}|) - |\mathbf{A}_{ii}|$. Let λ_i denote the i-th eigenvalue of $\mathbf{A}$. Then $\mathbf{A}_{ii} - r_i \leq \lambda_i \leq \mathbf{A}_{ii} + r_i$.*

To determine whether all eigenvalues of the matrix are of absolute value smaller equal one, we have to analyse each row separately. According to Theorem 2.1 it suffices to constrain the non-negative eigenvalues. Say the diagonal element of the row we are studying is non-negative. Then it is enough to show that the sum of the absolute values of the row elements is smaller or equal one. If on the other hand the diagonal element is smaller zero then we have to determine the sum of the absolute values of all non-diagonal elements of the row and add to this the (strictly negative) value of the diagonal element. If this then does not exceed the value one, we can say the same of the absolute values of the eigenvalues of the matrix.

We want to remark that the theorem works analogously when observing columns instead of rows. In fact, all matrix transformations that do not affect the spectral radius can be done before applying the criterion, which may lead to differing intervals constraining the eigenvalues. Trying several different transformations can lead to more precise results, since the eigenvalues will lie in the intersection of the obtained intervals. In the spirit of these thoughts, Geršgorin proposes to sharpen intervals that do not intersect with intervals for the other eigenvalues by replacing

$$r_i = (\sum_{j=1}^{n} |\mathbf{A}_{ij}|) - |\mathbf{A}_{ii}|$$

with

$$r_i = \mathbf{max}\{ \max_{j=1}^{i-1}\{g(j)\}, \max_{j=i+1}^{n}\{g(j)\} \}$$

where

$$g(x) = \frac{u}{2} - \frac{1}{2}[u^2 - 4\mathbf{A}_{xi}((\sum_{k=1}^{n}|\mathbf{A}_{xk}|) - |\mathbf{A}_{ii}|)]^{1/2}$$

and

$$u = |\mathbf{A}_{ii} - \mathbf{A}_{xx}| - ((\sum_{k=1}^{n}|\mathbf{A}_{xk}|) - |\mathbf{A}_{xi}| - |\mathbf{A}_{xx}|).$$

4 Upper Bounds on the Zeroes of Polynomials

Tackling the problem head-on, we will review bounds for the value of non-negative roots found in the literature. This falls into "à priori bounds", one of the 29 categories for root finding identified by McNamee in his bibliography on the topic [58]. Most of the bounds are rather crude, and if they happen to be very precise then more often than not they are rather complex to calculate. Since we need to keep the size of our constraints within reason if hoping for a chance of successfully bounding the complexity of term rewrite systems (TRSs for short), we will only list bounds which we deem promising. Bounds that are overly complex to calculate are not likely to yield positive results.

These constraints will model the condition $\rho(\chi_{\mathbf{A}}) \leq 1$ in a sufficient but not necessarily complete manner meaning the constraints always imply the desired condition but the reverse may not hold true. In practical terms it remains to be seen if we can cover enough of the set of polynomials with spectral radius less or equal to 1 to find derivational complexity bounds for a lot of the TRSs of the Termination Problems Database. Keep in mind that the condition of Theorem 2.4 is itself not complete, only the condition stated in Theorem 2.3 is.

4.1 A Word on Lower Bounds and Negative Roots

Due to the nature of the ultimate goal of this thesis, which is the synthesis of constraints that keep the spectral radius of a matrix smaller equal one, we will purely elaborate on finding upper bounds for non-negative roots in the course of the chapter. Bear in mind that this presents no restriction in and of itself, as adaptation of bespoke bounds to tasks that require one to identify lower bounds on non-negative roots or bounds on non-positive roots is fairly straightforward. The means to do so are covered in the next two lemmata.

Substituting the indeterminate for its negated counterpart will change both whether we are looking at non-negative or non-positive roots and an upper or lower bound *at the same time*. In contrary, substituting the indeterminate by its inverse will change an upper to a lower bound and vice versa, but have no effect on the sign of the bound. Combining these two substitutions then finally allows to apply the bound on the opposite interval without changing the nature (i.e. whether it is considered upper or lower) of the bound. We formulate these ideas in the lemmas and Corollary 4.3 below.

Lemma 4.1 ([64])**.** *Let f be a polynomial in x and g the same polynomial as f but with the indeterminate substituted by $-x$ and let n be a real number. The polynomial f has*

no real roots greater than n if and only if g has no real roots less than $-n$.

Lemma 4.2 ([64])**.** *Let f be a polynomial in x and let g be the same polynomial as f but with the indeterminate substituted by $\frac{1}{x}$ and let n be a real number. The polynomial f has no real roots greater than n if and only if g has no real roots less than $\frac{1}{n}$.*

Corollary 4.3 ([64])**.** *Let f be a polynomial in x and let g be the same polynomial as f but with the indeterminate substituted by $-\frac{1}{x}$ and let n be a real number. The polynomial f has no real roots greater than n if and only if g has no real roots greater than $-\frac{1}{n}$. Conversely, the polynomial f has no real roots smaller than n if and only if g has no real roots smaller than $-\frac{1}{n}$.*

With the knowledge that we can investigate the left-hand part of the x-axis as easily as the right-hand part and that we can readily transform an upper to a lower bound, we are now well-equipped to study the various methods providing us with upper bounds on the non-negative roots of a polynomial.

4.2 Descartes' Rule of Signs

A precursor of the Budan-Fourier-Theorem (cf. Theorem 5.5 and Theorem 6.1), *Descartes' Rule of Signs* is a criterion that allows to find an upper bound for the number of roots in certain intervals of a polynomial. In its initial form it was formulated to facilitate bounding the number of strictly *positive* roots of a polynomial. Despite being a fairly simple rule, historically speaking it is of quite some relevance since it inspired the theorems of Budan and Fourier and in turn the theorems of Sturm and Vincent, which form the basis for modern root finding algorithms. All that Descartes' Rule entails is looking at the sign differences of the coefficients of the polynomial. For a procedure that is based on such a simple principle it is surprisingly versatile. Not only do we get an upper bound on the number of roots but also the information whether the number of roots will be even or odd. And although it has originally been stated as a method to determine a bound on the number of strictly positive real roots the same procedure can be used to count the number of strictly negative roots. By shifting the polynomial before counting the sign changes, we can determine the number of roots in a strictly positive interval starting at a real number different to zero. Combined with the possibility to also evaluate the maximum number of roots in strictly negative intervals, we can determine the bound on the number of roots in any strictly positive or strictly negative real interval with endpoints ∞ or $-\infty$ respectively. For our purpose, the ability to bound the number of real roots in the interval $(1, \infty)$ is of main interest. If we know the number of roots at position zero (e.g. the polynomial has no multiple roots) then Descartes' rule can also be used to bound the total number of real roots of a polynomial and at the same time gauge the parity of the number of roots. This astounding theorem was published in 1637 by the French mathematician René Descartes.

We already defined the procedure for counting the sign changes of a polynomial in the preliminaries. Remember that any monomials that equal zero are simply dropped. Constant polynomials have no sign change.

There are exactly two cases in which the upper bound gives the exact number of roots: for 0 and 1 sign variations. Cardano has found out about these two cases before the existence of Descartes' rule, so in a sense a rule specialized to these two special cases could be considered a precursor of Descartes' Law of Signs.

Theorem 4.4 (Descartes' Law of Signs, [25])**.** *Let f be a non-constant polynomial in x and let n be a real number not equal zero.*
If $n < 0$:

> *The number of roots in the interval $(-\infty, n)$ is smaller or equal to the number of sign changes in the polynomial $f(-x+n)$. If there is a discrepancy between this bound and the actual number of roots, then the difference has to be even.*

If $n > 0$:

> *The number of roots in the interval (n, ∞) is smaller or equal to the number of sign changes in the polynomial $f(x+n)$. If there is a discrepancy between this bound and the actual number of roots, then the difference has to be even.*

Example 4.5. Consider the polynomial $f(x) = x^2 + 3$. Since there are no sign changes, we know that no strictly positive real root exists. To determine the number of strictly negative real roots we count the sign variations in $(-x)^2 + 3 = x^2 + 3$. No sign variations either. Additionally we have $f(0) > 0$. Hence the polynomial has no real roots, both its roots are complex.

From Descartes' Law of Signs we can derive a set of constraints on the characteristic polynomial of a matrix that fulfil the condition that the spectral radius of this matrix is smaller equal one. In the next sections, we will add more constraints fitting this condition. To ease quick reference, the constraints for each section will be listed at the very end of the sections. So in this case, Constraint 4.7 is listed at the end of this section.

If we combine Constraint 4.7 with constraints describing the shape of a characteristic polynomial, we have obtained the first sufficient formula, built from inequalities on the entries of a matrix, which models that the spectral radius of said matrix is no greater than one. If this matrix then is compatible to a TRS we have just established an upper bound on the derivational complexity of the term rewrite system. In Example 4.6 we look at how such a set of inequalities looks like for a symbolic 3×3 matrix. While the disjuncts in Constraint 4.7 are not overly complicated, the characteristic polynomial itself gets increasingly complex for symbolic matrices of higher dimensions (i.e. 5×5 or more). This is a problem inherent to all methods presented in this thesis that try to limit the absolute value of the eigenvalues of a matrix. It is quite a severe issue nonetheless: The colossal growth of the characteristic polynomial restricts the applicability of these methods to smaller matrices. Other methods of deriving bounds for the derivational complexity, such as EDA [60], are not affected by this and scale quite nicely to matrices of increasing size. As earlier results show[1], some examples could be solved due to the

[1] Cf. http://colo6-c703.uibk.ac.at/ttt2/hz/polymatrix/nontriangular/index.php, Lemma 12

option of using matrices of greater size. Still, for characteristic polynomials of small size, Descartes' Law of Signs provides a set of constraints of moderately low complexity. Let us see how innocuous the set of inequalities actually looks for a matrix of comparatively small size such as 3×3.

Example 4.6. Let $\chi_3(x) = px^2 + qx + r$ be the characteristic polynomial of some 3×3 matrix over the reals with no root at position 0. According to Constraints 4.7, if

$$\begin{aligned} &(p \neq 0 \vee q \neq 0 \vee r \neq 0) && \wedge \\ &[(p \geq 0 \wedge 2p + q \geq 0 \wedge p + q + r \geq 0) && \vee \\ &(p \leq 0 \wedge 2p + q \leq 0 \wedge p + q + r \leq 0)] \end{aligned}$$

Then the biggest strictly positive real root of χ_3 is smaller equal one and, due to Theorem 2.1, so is the largest absolute value of the eigenvalues of the matrix.

Ever since its inception, this theorem by Descartes has inspired numerous similar root bounding criteria. Whether it is root bounding, or even root counting or isolation, there are procedures aplenty that can be, in one form or another, traced back to Descartes' famous Law of Signs. Descartes' Law has not become completely obsolete either. Where Descartes' Law of Signs shines even today is in its intuitiveness and simplicity. But just as Descartes' rule can be seen as a generalization of Cardano's rule, Descartes' rule itself was generalized and has evolved over time. In the next section we will elaborate on some of the work the French mathematician Laguerre has done based on Descartes' theorem.

4.2.1 Constraints for Descartes' Rule of Signs

Constraint 4.7 (Definition 4.4)**.** *Let* $\sum_{0}^{n} a_n x^n = f \in \mathbb{R}[x] \setminus \{0\}$. *Further, let*

$$\begin{aligned} &[a_n \geq 0 \wedge n \cdot a_n + a_{n-1} \geq 0 \wedge \binom{n}{2} \cdot a_n + (n-1) \cdot a_{n-1} + a_{n-2} \geq 0 \wedge \ldots \wedge n \cdot a_n + \\ &(n-1) \cdot a_{n-1} + \ldots + a_1 \geq 0 \wedge a_n + a_{n-1} + \ldots + a_0 \geq 0] && \vee \\ &[a_n \leq 0 \wedge n \cdot a_n + a_{n-1} \leq 0 \wedge \binom{n}{2} \cdot a_n + (n-1) \cdot a_{n-1} + a_{n-2} \leq 0 \wedge \ldots \wedge n \cdot a_n + \\ &(n-1) \cdot a_{n-1} + \ldots + a_1 \leq 0 \wedge a_n + a_{n-1} + \ldots + a_0 \leq 0] \end{aligned}$$

Then f has no real root greater one.

4.3 Laguerre's Bound

A bound probably made popular by Uspensky's Theory of Equations [73], the mathematician Laguerre proposed both a theorem detailing conditions that guarantee a number to be a upper bound on the value of the strictly positive roots of a polynomial as well as

a simple method to find a number for any polynomial which, while not being all that close to an ideal bound, satisfies the conditions of the theorem. Speaking more strictly, Laguerre provided a proof of Descartes' Theorem and subsequently elaborated on some possible generalisations, which we will discuss in this chapter.

For the subsequent theorems, we need to introduce a recursive function.

Lemma 4.8. *Let $f(x) = \sum_{0}^{n} a_n x^n$ for some real numbers $n, a_0, \ldots, a_n$ and let c be a real number with $c > 0$. Then we have*

$$f_{\mathbf{m}}(x) = (x-c)(f_{\mathbf{0}}(c)x^{m-1} + f_{\mathbf{1}}(c)x^{m-2} + \ldots + f_{\mathbf{m-1}}(c)) + f_{\mathbf{m}}(c) \quad (m \leq n)$$

where the coefficient polynomials $f_{\mathbf{0}}, \ldots, f_{\mathbf{m-1}}$ and the remainder polynomial $f_{\mathbf{m}}$ are recursively defined as

$$f_{\mathbf{0}}(x) = a_n$$
$$f_{\mathbf{m}}(x) = f_{\mathbf{m-1}}(x) \cdot x + a_{n-m} \quad (m \leq n)$$

Remark. Observe that $f_{\mathbf{n}} = f$.

Obreshkov [64] discusses two alternative variants of Descartes' Law of Signs originally given by Laguerre [56]:

Theorem 4.9. *Let $n, a_0, \ldots, a_n, c, f, f_{\mathbf{0}}, \ldots, f_{\mathbf{n}}$ be defined as in Lemma 4.8. The number of roots $> c$ is smaller or equal to the number of sign changes in the sequence $f_{\mathbf{0}}(c), f_{\mathbf{1}}(c), \ldots, f_{\mathbf{n}}(c)$. If there is a discrepancy between this bound and the actual number of roots, then the difference has to be even.*

As described in [64, p. 68] using the theorem on the formula $x^n f(\frac{1}{x}) = 0$ leads to the following corollary:

Corollary 4.10. *Let $n, a_0, \ldots, a_n, c, f$ be defined as in Lemma 4.8. The number of roots $< c$ is smaller or equal to the number of sign changes in the sequence $a_0, a_1 \cdot c + a_0, \ldots, a_n \cdot c^n + a_{n-1} \cdot c^{n-1} + \cdots + a_0$. If there is a discrepancy between this bound and the actual number of roots, then the difference has to be even.*

Theorem 4.11. *Let $n, a_0, \ldots, a_n, c, f, f_{\mathbf{0}}, \ldots, f_{\mathbf{n}}$ be defined as in Lemma 4.8. If, for some $m \leq n$, $f_{\mathbf{k}}(c) \geq 0$ for all $k \leq m$ then, for all $c' > c$, $f_{\mathbf{k}}(c') \geq 0$ for all $k \leq m$.*

Based on this theorem, Laguerre stated another corollary: If for some position c every polynomial of the sequence of Lemma 4.8 before the last is non-negative we know that the non-constant parts of the last polynomial of the sequence are non-negative. Hence, if the last polynomial of the sequence is strictly positive at position c, the constant part of the polynomial is strictly positive too. Due to Theorem 4.11, we thus can conclude that the last polynomial of the sequence is greater zero if it is evaluated for any number greater equal c. This concept is captured in the next corollary.

Corollary 4.12. *Let $n, a_0, \ldots, a_n, c, f, f_{\mathbf{0}}, \ldots, f_{\mathbf{n}}$ be defined as in Lemma 4.8. If, for all $m < n$, $f_{\mathbf{m}}(c) \geq 0$ and if $f_{\mathbf{n}}(c) > 0$ then all strictly positive roots of f are strictly smaller than c.*

Example 4.13. Consider the polynomial $f(x) = x^4 - x + 2$. Using the recursive definitions from Lemma 4.8 we construct $f_{\mathbf{0}}(x) = 1, f_{\mathbf{1}}(x) = x, f_{\mathbf{2}}(x) = x^2, f_{\mathbf{3}}(x) = x^3 - 1$ and finally $f_{\mathbf{4}}(x) = x^4 - x + 2 = f(x)$. $f_0 \geq 0$ holds trivially, f_1 and f_2 are non-negative for any position that is non-negative, f_3 for all real numbers greater equal one, and $f = f_4$ is always strictly positive. Theorem 4.9 thus tells us that 1 is a strict upper bound for the number of strictly positive real roots of f. Since we chose f such that it has no real roots, we can observe here that the bound on strictly positive real roots is not as tight as an upper bound in this case could be.

Alas for our practical purposes, Laguerre presented conditions that lead to a strict upper bound (cf. Constraint 4.14). We desire to find constraints that provide us with an upper bound that is not strict, since we very much want to allow roots at position one. Manual inspection of samples of successful complexity proofs achieved by matrix-interpretation based methods mentioned in this thesis showed that a substantial amount of them had (multiple) roots at position one. Excluding these characteristic polynomials because the constraints are too strict would substantially weaken the approach (cf. Theorem 6.7).

4.3.1 Constraints for Laguerre's Bound

Constraint 4.14 (Corollary 4.10)**.** *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial built from real numbers satisfying the formula*

$$\begin{aligned}&(a_n \geq 0 \wedge a_n + a_{n-1} \geq 0 \wedge \ldots \wedge a_n + \ldots + a_0 \geq 0) \quad \vee \\ &(a_n \leq 0 \wedge a_n + a_{n-1} \leq 0 \wedge \ldots \wedge a_n + \ldots + a_0 \leq 0)\end{aligned}$$

Then f has no real root greater or equal one.

4.4 Kioustelidis' Bound

This bound was first presented by Kioustelidis in 1986 ([53]) and then later, according to his adviser (cf. [22]), independently discovered by Johnson in the process of assembling his PhD thesis ([48]).

Theorem 4.15. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with at least one strictly negative coefficient and a strictly positive leading coefficient. Then f has no non-negative real root greater or equal to $2 \cdot \max(\{ (\frac{|a_i|}{a_n})^{\frac{1}{n-i}} \mid i \in \mathbb{N}, i < n, a_i < 0 \})$.*

We explicitly want to point to Johnson's work [48] as a recommendation for further reading on the efficiency of real root counting algorithms.

4.5 Lagrange's Bound

Apart from his original work, Lagrange [54, 55] published two more theorems that are of relevance to us. One he attributes to MacLaurin and Newton, the other to MacLaurin

only. The theorems are:

Theorem 4.16 (MacLaurin)**.** *Let f be a polynomial of degree $n > 0$ with at least one strictly negative coefficient. Let m be the degree of the highest-degreed negatively-signed monomial of all the monomials f consists of. Let a_n be the leading coefficient and a_i be the strictly negative coefficient of the highest absolute value. Then f has no non-negative real root greater or equal to $1 + \sqrt[n-m]{\frac{|a_i|}{a_n}}$. Moreover, $f(x) > 0$ for all $x > 1 + \sqrt[n-m]{\frac{|a_i|}{a_n}}$.*

Theorem 4.17 (MacLaurin, Newton)**.** *Let f be a polynomial of degree $n > 0$ and let m be a real number. Further, let $f^{(i)}(m) > 0$ for all natural numbers $0 \leq i < n$. Then f has no non-negative real root greater than m.*

Lagrange also suggested a bound himself (cf. [54, Partie 12 – Scolie I]):

Theorem 4.18. *Let f be a normalized polynomial of degree $n > 1$ with at least two strictly negative coefficients. Let s be the sequence consisting of the $(n-k)$th root of the absolute value of the k-th coefficient for all strictly negative coefficients of f (excluding a_n). Adding the highest and second highest number of this sequence yields an upper bound on the non-negative real roots of f.*

Lagrange, however, did not include a proof. His proposition was verified by different authors, e.g. [22, 12].

Obreshkov ascribes a slight variation of the theorem to Cauchy (cf. [64, p. 50]):

Theorem 4.19. *Let f be a normalized polynomial of degree $n > 1$ with at least one non-leading strictly negative coefficient. Let m be the number of strictly negative coefficients and let s be the sequence consisting of the $(n-k)$th root of the product of the absolute value of the k-th coefficient and m for all strictly negative coefficients of f. The highest number of this sequence is an upper bound on the non-negative real roots of f.*

Since we want the roots to be no larger than 1, we have to equate $\sqrt[n-k]{m \cdot |a_k|}$ with one. We exponentiate by $n-k$ and get $m \cdot |a_k| = 1^{n-k} = 1$. We know the coefficient a_k is strictly negative, so we can equate $m \cdot |a_k|$ with $-m \cdot a_k$. Expressed in terms of a logical constraint this becomes probably the most elegant criterion we have encountered so far (cf. Constraint 4.22). For illustrative purposes we provide an example.

Example 4.20. Consider the polynomial $f(x) = x^2 - 3x$. The number of strictly negative coefficients is one, as is the index of this coefficient. Thus the absolute value of the only strictly negative coefficient, 3, is an upper bound for the non-negative roots of f.

4.5.1 Constraints for Lagrange's Bound

Constraint 4.21 (Theorem 4.18)**.** *Let* $f(x) = \sum_{i=0}^{n} a_i x^i$ *be a polynomial built from real numbers satisfying the formula*

$$
\begin{aligned}
&\exists m_0\ m_1.\, n > 1 \wedge m_0 + m_1 \leq 1 \wedge \\
&[\bigvee_{i=0}^{n-2} \bigvee_{j=i+1}^{n-1} (a_i < 0 \wedge a_j < 0 \\
&\quad \wedge ((m_0 = \sqrt[n-i]{|a_i|} \wedge m_1 = \sqrt[n-j]{|a_j|}) \vee (m_1 = \sqrt[n-i]{|a_i|} \wedge m_0 = \sqrt[n-j]{|a_j|})))] \wedge \\
&[\sqrt[n]{|a_0|} \neq m_0 \vee (\sqrt[n]{|a_0|} \geq \sqrt[n-1]{|a_1|} \wedge \ldots \wedge \sqrt[n]{|a_0|} \geq \sqrt[1]{|a_{n-1}|})] \\
&\quad \wedge \ldots \wedge \\
&[\sqrt[1]{|a_{n-1}|} \neq m_0 \vee (\sqrt[1]{|a_{n-1}|} \geq \sqrt[n]{|a_0|} \wedge \ldots \wedge \sqrt[1]{|a_{n-1}|} \geq \sqrt[2]{|a_{n-2}|})] \wedge \\
&[\sqrt[n]{|a_0|} \neq m_1 \vee ((\sqrt[n-1]{|a_1|} = m_0 \vee \sqrt[n]{|a_0|} \geq \sqrt[n-1]{|a_1|}) \wedge \ldots \wedge \\
&\qquad (\sqrt[1]{|a_{n-1}|} = m_0 \vee \sqrt[n]{|a_0|} \geq \sqrt[1]{|a_{n-1}|}))] \\
&\quad \wedge \ldots \wedge \\
&[\sqrt[1]{|a_{n-1}|} \neq m_1 \vee ((\sqrt[n]{|a_0|} = m_0 \vee \sqrt[1]{|a_{n-1}|} \geq \sqrt[n]{|a_0|}) \wedge \ldots \wedge \\
&\qquad (\sqrt[2]{|a_{n-2}|} = m_0 \vee \sqrt[1]{|a_{n-1}|} \geq \sqrt[2]{|a_{n-2}|}))]
\end{aligned}
$$

Then the non-negative real roots of f *have an upper bound of* 1*.*

Remark. The constraint is not suited to integer polynomials (roots of strictly positive integers cannot be smaller one, which leads to the contradiction $m_0 + m_1 > 1$). We want to note, however, that Lagrange's Bound might be a fruitful method in a context where the coefficients can be real.

Constraint 4.22 (Theorem 4.19)**.** *Let* $f(x) = \sum_{i=0}^{n} a_i x^i$ *be a polynomial built from real numbers satisfying the formula*

$$
[\bigvee_{i=0}^{n-1} (a_i < 0 \wedge \bigwedge_{j=0}^{n-1} (a_j \geq 0 \vee a_i \leq a_j) \wedge m = \sum_{j=0}^{n-1} \max(-\frac{a_j}{|a_j|}, 0) \wedge -m \cdot a_i \leq 1)] \wedge a_n = 1
$$

Then f *has no real root greater one.*

4.6 Laguerre (cont'd)

With the help of Laguerre's and Lagrange's works, Villarino [74] proves a theorem that states a sufficient and necessary criterion on a (univariate) polynomial and a divisor to

ensure non-negativity of the coefficients of both the quotient and the remainder of the corresponding polynomial division.

Theorem 4.23 ([74])**.** *Let f be a polynomial in x which has a strictly positive leading coefficient and at least one strictly positive root n. Then there is some $m \geq n$, a quotient polynomial g and a remainder polynomial h such that*

$$f(x) = (x - m) \cdot g + h$$

and all coefficients of g and h are non-negative.

Villarino gives the following practical use case for the above theorem: The leading coefficient of the quotient coincides with the leading coefficient of the original polynomial. This implies the quotient polynomial is not zero. Since the divisor in Theorem 4.23 is of the form $x - m$, we can infer that for $l > m$, $f(l)$ is > 0. We thus have proven, using the above theorem, that $f(x) > 0$ for all numbers greater than m. We could optionally test for $f(m) > 0$ *($h = 0$)* if we want to know whether $f(x) > 0$ for all numbers greater *or equal* to m.

He then also gives an example demonstrating that the non-negative real number fulfilling Theorem 4.23 is not necessarily the smallest number for which f becomes and remains strictly positive. If we look at $f(x) = (x-1)(x-2)(x-3)(x-4)$, the largest root would be 4 while the first whole number to elicit purely non-negative coefficients in quotient and remainder is 10. Expanding f we have $f(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$; and thus $f(x) = (x^3 + 35x + 300) \cdot (x - 10) + 3024$, where we observe that quotient and remainder are non-negative. This example clearly illustrates that there can be a significant gap between obtainable and ideal value. In fact, Villarino comments that to fit Theorem 4.23, the number must not be smaller than the roots of polynomials $f_{\mathbf{1}}, \ldots, f_{\mathbf{n}}$ we introduced in Section 4.3 for a polynomial f. We specifically exclude $f_{\mathbf{0}}$ since the leading coefficient is required by Theorem 4.23 to be strictly positive and $f_{\mathbf{0}}$ therefore has no root. In the next example we show the smallest fitting number for $f(x) = (x-1)(x-2)(x-3)(x-4)$ is thus 10.

Example 4.24. Let $f(x) = (x-1)(x-2)(x-3)(x-4) = x^4 - 10x^3 + 35x^2 - 50x + 24$. We determine the highest strictly positive roots of the polynomials $f_{\mathbf{1}}, f_{\mathbf{2}}, f_{\mathbf{3}}, f_{\mathbf{4}}$ as defined in Lemma 4.8.

$f_{\mathbf{1}} = x - 10$	*largest root:* 10
$f_{\mathbf{2}} = x^2 - 10x + 35$	-
$f_{\mathbf{3}} = x^3 - 10x^2 + 35x - 50$	*largest root:* 5
$f_{\mathbf{4}} = x^4 - 10x^3 + 35x^2 - 50x + 24$	*largest root:* 4

From this we can infer that Theorem 4.23 will hold for all $m \geq 10$.

Remark. In respect to Theorem 4.23, note that $f_{\mathbf{0}}(m), \ldots, f_{\deg(\mathbf{f}-\mathbf{1})}(m)$ are precisely the coefficients of g and $f_{\deg(\mathbf{f})}(m)$ is the single coefficient of h.

Theorem 4.25 ([74])**.** *Let f be a polynomial in x of degree n which has a strictly positive leading coefficient and at least one strictly positive root. Let $f_{\mathbf{0}}, \ldots, f_{\mathbf{n}}$ be as defined in Lemma 4.8. Let m be a strictly positive real number such that the entries in the sequence $f_{\mathbf{1}}(m), \ldots, f_{\mathbf{n}}(m)$ are all non-negative and at least one of them strictly positive. Then m is greater or equal to the largest strictly positive root of the (non-zero) polynomials $f_{\mathbf{1}}, \ldots, f_{\mathbf{n}}$ and thus implicitly greater or equal to the largest strictly positive root of $f_{\mathbf{n}} = f$. The same is true for the inverse direction.*

Proof (adapted from [74]).

$\Rightarrow$: Suppose $f_{\mathbf{k}}(m') = 0$ for some $m' > m$ and some $k \in 1, \ldots, n$. As per the assumption, $f_{\mathbf{k}}(m) \geq 0$. Suppose $f_{\mathbf{k}}(m) > 0$. Since the hypothesis assures non-negativity of $f_{\mathbf{1}}(m), \ldots, f_{\mathbf{k-1}}(m)$, by Lemma 4.12 m is strictly greater than the largest root of $f_{\mathbf{k}}$. This contradicts the assumption that there is a root of $f_{\mathbf{k}}$ that is larger than m. Now suppose that $f_{\mathbf{k}}(m) = 0$. Referring to Lemma 4.8 we obtain the identity $f_{\mathbf{k}}(m) = (m - m')(f_{\mathbf{0}}(m')m^{n-1} + \ldots + f_{\mathbf{k-1}}(m')) + f_{\mathbf{k}}(m') = 0$. Since $f_{\mathbf{0}}(m') = a_n = f_{\mathbf{0}}$ and m are both strictly positive, as well as $m - m' < 0$, we see that $f_{\mathbf{k}}(m) < 0$ which contradicts the assumptions.

$\Leftarrow$: Suppose all entries of the sequence $f_{\mathbf{1}}(m), \ldots, f_{\mathbf{n}}(m)$ are zero. Then, according to the identities stated in Lemma 4.8, the coefficients $a_{n-1}, \ldots, a_0$ have to be zero and thus $f(x) = a_n x^n$. This contradicts the hypothesis that f has at least one strictly positive root. W.l.o.g. we assume m to be equal to the largest strictly positive root of the polynomials in the sequence $f_{\mathbf{1}}, \ldots, f_{\mathbf{n}}$. If m were to be greater than the largest strictly positive root, due to Lemma 4.11, the non-negativity result would still apply. Suppose $f_{\mathbf{k}}(m) < 0$ for some $k \in 1, \ldots, n$. Since $f_{\mathbf{k}}(m)$ is strictly negative, at least one coefficient has to be strictly negative. Based on the strictly negative coefficient(s), Theorem 4.16 gives us a bound $u > m$ such that for any $u' > u$, $f_{\mathbf{k}}(u') > 0$. The Intermediate Value Theorem then suggests the existence of a root $m' > m$ for $f_{\mathbf{k}}$ which contradicts the maximality of m.

□

4.7 Regrouping

Not tailored to be applicable to symbolic polynomials, where we have no information on concrete shape or form, we still want to introduce another method to find a root bound as exhibited in Obreshkov's book *Verteilung und Berechnung der Nullstellen reeller Polynome* [64]. In the context of our research question, the greatest bane of this method is that it is very much an impromptu method, which works exceedingly well when given a concrete polynomial to work with. At first glance, reverse engineering the method to use it for the synthesis of suiting polynomials seems far from optimal. The method's flexibility leaves room for lots of disjuncts in the final constraints. Heuristics reducing the amount of disjuncts may work reasonably well, though, since there is a common pattern that can be focused on.

We first introduce a lemma that captures the intuitive notion that, if a polynomial whose degree is smaller or equal to the lowest exponent of the indeterminate in a polynomial with strictly positive coefficients only is subtracted from the latter polynomial, once the polynomial resulting from this subtraction becomes strictly positive, say at position n, it will remain strictly positive from this point onwards. This immediately follows from the fact that if we divide both the minuend and subtrahend by the indeterminate, let it be denoted by x, to the power of the degree of the subtrahend (an operation that does not affect the sign), then for increasing x, the minuend will grow bigger whilst the subtrahend gradually becomes smaller, rendering a change to a negative sign due to an increase in x impossible.

Lemma 4.26. *Let f and g be two normalized polynomials with non-negative coefficients only and all monomials of f, with degree smaller to the degree of g, being zero. Let h be the result of subtracting g from f. If there is a real number $n > 0$ such that $h(n) > 0$, then h is strictly positive for every number greater equal n.*

We reformulate Lemma 4.26 to account for our intention of having a non-strict upper bound (we want to allow roots at position exactly one).

Lemma 4.27. *Let f and g be two normalized polynomials with non-negative coefficients only and all monomials of f, with degree smaller to the degree of g, being zero. Let h be the result of subtracting g from f. If there is a real number $n > 0$ such that $h(n) \geq 0$, and if $h > 0$ in $(n, n + \varepsilon]$ for some $\varepsilon > 0$, then h is strictly positive for every number greater than n.*

To this end we introduce the notion of local minima and local maxima, as well as the locally order preserving property and locally order reversing property of a function. We follow the standard textbook definitions.

Definition 4.28. Let f be a univariate polynomial over the reals. A *local minimum* is a real number n that satisfies $f(n - m) > f(n)$ and $f(n + m) > f(n)$ for all $m > 0$ smaller equal some real number $\varepsilon > 0$.

Definition 4.29. Let f be a univariate polynomial over the reals. A *local maximum* is a real number n that satisfies $f(n - m) < f(n)$ and $f(n + m) < f(n)$ for all $m > 0$ smaller equal some real number $\varepsilon > 0$.

Definition 4.30. Let f be a univariate polynomial over the reals and let n be a real number. Let $f(n - m) < f(n) < f(n + m)$ for all $m > 0$ smaller equal some real number $\varepsilon > 0$. Then f is *locally order preserving* in the neighbourhood of n.

Definition 4.31. Let f be a univariate polynomial over the reals and let n be a real number. Let $f(n - m) > f(n) > f(n + m)$ for all $m > 0$ smaller equal some real number $\varepsilon > 0$. Then f is *locally order reversing* in the neighbourhood of n.

First, we check if h is locally order preserving in the neighbourhood of n to identify whether h is strictly increasing in $(n, n + \varepsilon]$. This is a sufficient criterion but covers only

one case. In case the derivative is exactly zero at position n, we can use the derivative test as a second sufficient criterion. The function h fits the criterion if it has a local minimum at n. These two criteria are presented in Lemma 4.33.

Lemma 4.32. *Let f be a non-zero polynomial that has at least one real root. Then f has a derivative of an order smaller equal to its degree that has no real roots.*

Proof. Let n be the degree of f. Suppose $n = 0$. Then $f = c \neq 0$ for some real number c apparently has no real root, which contradicts the assumption. Hence n necessarily is strictly positive. Since the leading term of f is non-zero per the assumption, it suffices to look at the n-th derivative of the leading monomial (all other monomials evaluate to zero since the order of the derivative exceeds their degree) to verify the claim: $(ax^n)^{(n)} = (n!)a \neq 0$. □

Lemma 4.33. *Let f be a non-zero polynomial of degree n with a root $f(m) = 0$. Further let $k \leq n$ such that $f'(m) = \ldots = f^{(k-1)}(m) = 0$ but $f^{(k)}(m) \neq 0$. Suppose $f^{(k)}(m) > 0$. Then $f(m)$ is a local minimum if k is even, and f is locally order preserving in the neighbourhood of m if k is odd. Conversely, suppose that $f^{(k)}(m) < 0$. Then $f(m)$ is a local maximum if k is even, and f is locally order reversing in the neighbourhood of m if k is odd.*

Example 4.34. Let $f = x^3 + x$. The polynomial has a single real root at position zero. We have $f'(0) = 3 \cdot 0^2 + 1 = 1$. Since already the first derivative is not equal to zero at position 0 and, in fact, strictly positive, we can assert according to Lemma 4.33 that f is locally order preserving in the neighbourhood of 0.

We will now illustrate how Lemma 4.27 can be used to determine a bound for the positive roots.

Example 4.35. Let $f(x) = x^3 - 2x^2 + 1.25x - 0.25$. We can unpick this into $f = i + \sum_{j=0}^{n} h_j * x^{k_j}$ for some natural number n. The most straight-forward choice in our case probably is $h_0 = x - 2$, $k_0 = 2$, $h_1 = 1.25x - 0.25$, $k_1 = 0$ and $i = 0$. We now have to prove that both h_0 and h_1 comply to Lemma 4.27 for some bound m. Since $h_0(2) = 2 - 2 = 0$, $h_0'(2) > 0$ and $h_1(2) > 0$ we can infer that 2 is an upper bound for the value of strictly positive roots of f. Note that since $f = (x - 1)(x - 0.5)^2$ the highest valued real root has value 1 but the best bound the regrouping method can provide is 2. Clearly, the polynomial in this example fits the necessary criteria in our pursuit to find polynomials with spectral radius less or equal 1 but would not have been flagged as compatible when using the regrouping method.

Let us contrast this with an example where the regrouping method gives way to an optimal bound. To this end, let $f(x) = (x + 0.5)^2(x - 1) = x^3 - 0.75x - 0.25$. Once again, the optimal bound would be 1. Since f immediately fits the criteria of Lemma 4.27, we simply choose $h_0 = f$ and $k_0 = 0$. We have $h_0(1) = 1 - 0.75 - 0.25 = 0$ as well as $h_0'(1) = 3 - 0.75 > 0$. Hence we can conclude that 1 is an upper limit on the signed value of the real roots of f, and in this case it is indeed exactly equal to the maximal strictly positive root.

By limiting the degree of f and restricting the number of polynomials h_j, the template Constraint 4.36 generates a finite set of constraints. One pattern commonly found in practical use of the regrouping method on concrete, i.e. non-symbolic, polynomials gives way to the following heuristic of which shape to impose on the polynomials h_j available. Starting from the leading normalized monomial we construct the first polynomial h_0 by summing the monomials of gradually decreasing degree, until we encounter a change from a strictly negative to a strictly positive coefficient or have reached the monomial of the lowest degree. This strictly positive coefficient will start forming the next polynomial $h_k + 1$, and so forth. Remaining strictly positive monomials of the lowest degrees will form the polynomial i. Let the degree we have chosen for f be d. The maximum number of sign combinations, constant zero monomials need not be accounted for as they fit with either the positive or negative sign case, and thus the maximum number of polynomial combinations, built from a finite number of polynomials h_j and the polynomial i consisting of the monomials with non-negative coefficients farthest to right, f can consist of using the heuristic is 2^d. The number of polynomials h_j such a combination can have with our heuristic is equal to $\lceil \frac{d}{2} \rceil$. So we have 2^d disjuncts consisting of no more than $\lceil \frac{d}{2} \rceil + 2$ (we need to add one for the polynomial i and the conjunct $f = i + \sum_{j=0}^{\lceil \frac{d}{2} \rceil} h_j$ respectively) conjuncts each.

4.7.1 A Template for Constraints Based on Regrouping of the Polynomial

Constraint 4.36 (Lemma 4.26)**.** *Let f be a real polynomial for which*

$$
\begin{aligned}
& f = i + \sum_{j=0}^{n} h_j \wedge n \in \mathbb{N}^+ \cup \{0\} \wedge \\
& [\, \forall j \in \mathbb{N}^+ \cup \{0\}.\, 0 \leq j \leq n.\, \exists v\ d \in \mathbb{N}^+ \cup \{0\}.\, \exists b_0 \cdots b_v\ c_v \cdots c_d \in \mathbb{R}. \\
& \quad h_j = (\sum_{u=v}^{d} c_u \cdot x^u) - (\sum_{u=0}^{v} b_u \cdot x^u) \wedge h_j(1) > 0 \wedge v \leq d \wedge \\
& \quad b_0 \geq 0, \ldots, b_v \geq 0 \wedge c_v \geq 0, \ldots, c_d \geq 0 \wedge b_v > 0 \wedge c_d > 0\,] \wedge \\
& \exists a_0 \cdots a_n \in \mathbb{R}.\, i = \sum_{k=0}^{n} a_k \cdot x^k \wedge a_0 \geq 0 \wedge \ldots \wedge a_n \geq 0
\end{aligned}
$$

Then f has no real root greater one.

4.8 Eneström-Kakeya

The Eneström-Kakeya Theorem [34, 52] is a non-necessary criterion only but otherwise perfectly suited to our aims. It is conceivably simple and would even work if we did not know about Theorem 2.1. There are also examples where the upper bound of absolute value one is actually sharp, as witnessed by the linear polynomial $x + 1$.

Theorem 4.37. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial whose coefficients are strictly positive. Further, let $a_n \geq a_{n-1} \geq \ldots \geq a_0$. Then f has no real root with absolute value greater one.*

A translation of the original work by Eneström [34] is provided in the appendix of [39]. In this survey the authors present an extensive historical overview on the developments of the Eneström-Kakeya Theorem, which will be our rough guideline for the course of this section. While discussing the theorems we will inspect each of them from the perspective of our research question: what would the theorem look like when the real roots of a polynomial with real coefficients shall not exceed one?

We first want to lay out the formal structure of this section so as to aid the reader in navigating the rather extensive assortment of theorems. The analysis of the theorems is presented as a sequence of multiple paragraphs focussed on a single theorem. Each appraisal comes in three parts: We first formally present the theorem. Below most theorems we discuss their applicability to the specific case of an upper bound equating one and describe the process of constructing a constraint modelling our research problem. Then we segue into the next evaluation by discussing the history of the next theorem or its relation to the previous theorem. As in earlier sections, the constraints themselves are listed at the end of the section.

First to propose a modification of the original Eneström-Kakeya Theorem was probably Hurwitz [47] by sketching a variant that was both sufficient and *necessary*. Perhaps even more important to us is that Hurwitz in this paper asked the question what the sufficient and necessary criteria would be such that the Eneström-Kakeya bound is sharp, i.e. the polynomial has a root with absolute value exactly one. As we elaborate on in greater detail in another section of this thesis, we are most interested in polynomials that have some roots at position exactly 1. Including additional constraints that enforce this could potentially boost the efficiency of our method drastically, provided these additional constraints are simple enough. Indeed those by Hurwitz are.

Theorem 4.38. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial whose coefficients are strictly positive. Let $m > 1$ be a factor of $n+1$. Further let $a_n = a_{n-1} = \ldots = a_{n+1-m} \geq a_{n-m} = a_{n-m-1} = \ldots = a_{n+1-2m} \geq \ldots \geq a_{m-1} = a_{m-2} = \ldots = a_0$. Then f has no real root with absolute value greater one and at least one real root with absolute value equal one.*

Joyal, Labella and Rahman [50] proposed a theorem that abolishes the positivity requirement for the coefficients.

Theorem 4.39. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with a non-zero leading coefficient. Further, let $a_n \geq a_{n-1} \geq \ldots \geq a_0$. Then f has no real root with absolute value greater than $\frac{a_n - a_0 + |a_0|}{|a_n|}$.*

Assume the leading coefficient is strictly positive. If additionally the trailing coefficient is non-negative, the bound evaluates to $\frac{a_n}{a_n} + \frac{|a_0| - a_0}{|a_n|} = 1 + 0 = 1$. Would the trailing coefficient be strictly negative, the bound would equate $1 + \frac{2|a_0|}{a_n}$ which cannot ever

evaluate to one, which means that we can ignore this case for our purposes. In case the leading coefficient is strictly negative, the assumptions in the above theorem demand the trailing coefficient to follow suit. When both leading and trailing coefficients are strictly negative we have $\frac{2a_0}{a_n} - 1$ which evaluates to one if and only if $a_n = a_0$. This specializes the theorem to the condition $a_n = a_{n-1} = \ldots = a_0$. From these case distinctions we construct Constraint 4.66.

Remark. Note that since to achieve the goal of this thesis we need to work with *characteristic polynomials*, the absolute value of the leading coefficient equates 1.

Aziz and Zargar contributed several new generalizations [9, 8] to the theorem. The first extension of the theorem relaxes the requirements for the value of the leading coefficient. Their second generalization also allows for more flexibility in the trailing coefficient. The new theorems are as follows:

Theorem 4.40. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with a non-zero leading coefficient. Further, let $m \cdot a_n \geq a_{n-1} \geq \ldots \geq a_0$ and m be a real number greater equal one. Then if k is a real root of f we can assert that $-\frac{ma_n - a_0 + |a_0|}{|a_n|} + 1 - m \leq k \leq \frac{ma_n - a_0 + |a_0|}{|a_n|} + 1 - m$.*

In case both the leading and trailing coefficients are non-negative, the real roots of f lie between $1 - 2m$ and 1. Once again, the case where the trailing coefficient is strictly negative whereas the leading coefficient is not, is not of interest for the purpose of our work, since the upper bounds of the roots of f would become $1 + \frac{2a_0}{a_n}$ which with these presupposition can in no way equate to 1. If both a_n and a_0 are strictly negative, we have $\frac{2a_0}{a_n} + 1 - 2m$ as the upper bound. We can further simplify this expression if we look at the broader assumptions we can work with: Since we impose the constraints we are searching for on characteristic polynomials only, we know that the leading coefficient has (absolute) value one. This simplifies the expression to $1 - 2a_0 - 2m$. It is now easy to see that the bound evaluates to one if and only if $a_0 = -m$. Thus we can construct Constraint 4.67. While when imposed on characteristic polynomials the second disjunct still comes down to $a_n \geq a_{n-1} = \ldots = a_0$, we do now have the option to choose any real number smaller equal $a_n = -1$ for the non-leading coefficients. The first disjunct comes with even greater freedom, since again, the leading coefficient does no longer have to be the coefficient of biggest numeric value.

The next theorem further reduces the restrictions we need to impose on the polynomial by weakening the conditions for the trailing monomial.

Theorem 4.41. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with a strictly positive leading coefficient whose other coefficients are non-negative. Further, let $m_0 \cdot a_n \geq a_{n-1} \geq \ldots \geq a_1 \geq m_1 \cdot a_0$, let m_0 be a real number greater equal one and let m_1 be a strictly positive real number smaller equal one. Then if k is a real root of f we can assert that $-\frac{2(1-m_1)a_0}{a_n} - 2m_0 + 1 \leq k \leq \frac{2(1-m_1)a_0}{a_n} + 1$.*

We first want to remark that the choice of m_0 has no influence on the upper bound of the root, save the indirect influence on available choices for m_1. Further, note that $1 - m_1$ is non-negative. In contrast to the earlier theorems, we can omit the absolute value function since the divisor is guaranteed to be strictly positive.

If m_1 is equal one then the non-strict upper bound for the roots is one. If, on the other hand, m_1 is smaller one then the bound is one if we further demand that a_0 is zero. Put simply, this means that for a bound of value one the factor m_1 vanishes. In both cases the value of the leading term is of no concern.

We have gathered these insights in Constraint 4.68. You may notice that the resulting constraint equates the first disjunct of Constraint 4.67.

The third Eneström-Kakeya-related theorem by Aziz and Zargar provides an interesting twist as the leading coefficient no longer has to be the highest-valued one.

Theorem 4.42. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with a non-zero leading coefficient. Further, let m_0 be a natural number smaller than n, let m_1 be a strictly positive real number smaller equal one and let $a_n \leq a_{n-1} \leq \ldots \leq a_{m_0+1} \leq a_{m_0} \geq a_{m_0-1} \geq \ldots \geq a_1 \geq m_1 \cdot a_0$. Then if k is a real root of f we can assert that $\frac{a_{n-1}-2a_{m_0}+m_1a_0+(m_1-2)|a_0|}{|a_n|} - \frac{a_{n-1}}{a_n} + 1 \leq k \leq \frac{-a_{n-1}+2a_{m_0}-m_1a_0+(2-m_1)|a_0|}{|a_n|} - \frac{a_{n-1}}{a_n} + 1$.*

We can ignore the scaling by the factor ${a_n}^{-1}$ as all that matters is that the dividend equates to zero. To account for the summands where the term to some part consists of the absolute value function, we need to make a case distinction on the sign of the trailing coefficient. If a_0 has a negative sign then the value of m_1 is of no significance since the terms containing it as factor cancel out. In stark contrast to Theorem 4.41, m_1 does play a role when the trailing coefficient is non-negative. On the whole, the resulting constraint is bit more complex than what we are used to from the earlier extensions of the Eneström-Kakeya Theorem. Now on to the actual construction of the constraint. Assume $m_0 = n - 1$. Then the terms $2a_{n-1}$ and $2a_{m_0}$ cancel out. If a_0 is strictly negative we are left with $2a_0$ only, which may not become zero under this assumption. Thus the only plausible case is a non-negative trailing coefficient. All remaining summands have a_0 as factor, whence equating a_0 with zero leads to a zero-sum in total. For a strictly positive a_0 the sum is $2a_0 - 2m_1a_0$ which equates to zero only if we choose m_1 to be zero. This covers the case where $m_0 = n - 1$. The cases where $m_0 < n - 1$ allow for less simplification since $2a_{n-1}$ and $2a_{m_0}$ do not cancel out. The conditions for $0 < m_0 < n - 1$ are all of the same shape. If a_0 is strictly negative we have to demand that $2a_{n-1} - 2a_{m_0} + 2a_0 = 0$, else we demand $2a_{n-1} - 2a_{m_0} + 2(1 - m_1)a_0 = 0$. Respective to the former, we take note that $a_{m_0} \geq a_{n-1}$ and thus $2a_{n-1} - 2a_{m_0}$ cannot be strictly positive which contradicts the assumption that a_0 is strictly negative and our claim that the sum is zero. We are thus left with the latter equation, which can be further simplified by dropping the common factor 2. On a side note, setting m_1 to 1 gives us the following specialization of Theorem 4.42, which closely resembles the original Eneström-Kakeya Theorem (Theorem 4.37): *[...] and let $a_n \leq a_{n-1} \geq \ldots \geq a_0 \geq 0$. Then f has no real root with absolute value greater than one.* There is one case remaining: m_0 equating zero leads to a constraint of yet another shape. Since a lot of the summands cancel out when a_0 is not strictly positive, we have $a_{n-1} = 0$ in this case. Lastly, with $a_0 > 0$ we have to demand $(2 - m_1)a_0 - a_{n-1} = 0$. Since $0 < m_1 \leq 1$ and $a_{n-1} \leq m_1 a_0$ imply $a_{n-1} \leq a_0$ and we can bound $0 = -a_{n-1} - m_1a_0 + 2a_0$ by $2a_0 - 2a_{n-1}$ which in turn implies $a_0 \leq a_{n-1}$, we can infer that $a_{n-1} = a_0$. This simplifies the constraint to $(1 - m_1)a_0 = 0$ which with

respect to our assumptions is equivalent to $m_1 = 0$. The resulting logical formula is listed at the end of the section as Constraint 4.69.

The next theorem (due to [61]) is an abstraction of the four previous theorems. As Mogbademu et al. remark, we can obtain any of the previous theorems when we parametrize their theorem accordingly. Although we do not concern ourselves with polynomials with complex coefficients in this thesis, we still want to note that this new theorem explicitly adds support for complex coefficients. Please note that we adjusted the theorem to account for two small typos we found in the original proof and theorem.

Theorem 4.43. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with complex coefficients and a non-zero leading coefficient. Further, let m_0 be a natural number smaller than n, let m_1 be a strictly positive real number smaller equal one, m_2 a strictly positive real number, m_3 a real number and let $(m_2)^n \cdot a_n \leq (m_2)^{n-1} \cdot a_{n-1} \leq \ldots \leq (m_2)^{m_0+1} \cdot a_{m_0+1} \leq (m_2)^{m_0} \cdot a_{m_0} + (m_2)^{m_0-1} \cdot m_3 \geq (m_2)^{m_0-1} \cdot a_{m_0-1} \geq \ldots \geq (m_2)^2 \cdot a_2 \geq m_2 \cdot a_1 \geq m_1 \cdot a_0$. Then if k is a real root of f we can assert that $\frac{1}{|a_n|}[m_2 a_n - m_3 + (m_2)^{1-n}(2m_1 - 1)a_0 - (m_2)^{1-n}|a_0| - (m_2)^{m_0-n+1} \cdot 2a_{m_0}] - \frac{m_3}{a_n} \leq k \leq \frac{1}{|a_n|}[-m_2 a_n + m_3 + (m_2)^{1-n}(1 - 2m_1)a_0 + (m_2)^{1-n}|a_0| + (m_2)^{m_0-n+1} \cdot 2a_{m_0}] - \frac{m_3}{a_n}$.*

As Mogbademu et al. note, this theorem is a generalization to each of the five preceding theorems. We obtain Theorem 4.37 if we set m_2 to 1, m_0 to n, m_3 to 0, m_1 to 1 and demand positivity of a_0. If we no longer impose constraints on the sign of a_0 it simplifies to Theorem 4.39. When we then also set m_3 to $(m-1) \cdot a_n$ for some $m \geq 1$ we are presented with the bounds of Theorem 4.40.

Among the first researchers to expand the scope and applicability of the Eneström-Kakeya Theorem were Govil and Jain. While Govil's and Jain's earlier work [44, 42] on generalizations of the Eneström-Kakeya Theorem almost exclusively pertained to the applicability of the Eneström-Kakeya Theorem to polynomials with complex coefficients, they did give one theorem that improves the original theorem by Eneström and Kakeya by adding a lower bound.

Theorem 4.44. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a non-zero polynomial whose coefficients are non-negative. Further, let $a_n \geq a_{n-1} \geq \ldots \geq a_0$. Then f has no real root with absolute value greater one or smaller $\frac{a_0}{2a_n - a_0}$.*

Govil and Jain sharpened their results in [43] to obtain a theorem that unlike their earlier work would also provide an upper bound for polynomials with complex coefficients. This eventually lead to a paper authored by Dewan and Govil [27] focused on a theorem specifically tailored to polynomials with real coefficients. By observing that the upper bound is smaller or equal to $\frac{a_n - a_0 + |a_0|}{|a_n|}$ and showing that the difference between upper and lower bound is less than one, Dewan and Govil could demonstrate that this theorem was an improvement over Theorem 4.39. They also provided select examples for which the bounds can be seen to be sharper.

Theorem 4.45. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with a non-zero leading coefficient, $u = \frac{a_n - a_{n-1}}{2}(|a_n|^{-1} - (a_n - a_0 + |a_0|)^{-1}) + [(\frac{a_n - a_{n-1}}{2})^2 \cdot (|a_n|^{-1} - (a_n - a_0 + |a_0|)^{-1})^2 +$*

$\frac{a_n - a_0 + |a_0|}{|a_n|}]^{1/2}$, $m = u^n \cdot (u|a_n| + a_n - a_0)$ and $l = \frac{1}{2m^2} \cdot \{-u^2(a_1 - a_0)(m - |a_0|) + [u^4(a_1 - a_0)^2(m - |a_0|)^2 + 4|a_0|u^2 m^3]^{1/2}\}$. Further, let $a_n \geq a_{n-1} \geq \ldots \geq a_0$. Then

$$\frac{a_n - a_0 + |a_0|}{|a_n|} \geq u \geq 1 \geq l \geq 0$$

and if k is a real root of f we can assert that $l \leq |k| \leq u$.

As Dewan and Govil remark, if a_0 is strictly positive this theorem simplifies to Theorem 4.37. To demonstrate its edge over Theorem 4.39 the aforementioned authors provided some concrete examples:

Example 4.46 (cf. [27])**.** Let $f(x) = 6x^4 + 4x^3 + 3x^2 + 2x - 100$. Theorem 4.39 asserts that $|k| \leq \frac{103}{3}$ while Theorem 4.45 asserts that $|k| \leq \frac{1}{6} - \frac{1}{206} + [(\frac{1}{6} - \frac{1}{206})^2 + \frac{103}{3}]^{1/2} \approx 6.02351$ for all real roots k of f.

Example 4.47 (cf [27])**.** Let $f(x) = \frac{1}{2}x^5 + \frac{1}{2}x^4 + \frac{2}{5}x^3 + \frac{3}{10}x^2 + \frac{1}{5}x - 1000$. Theorem 4.39 asserts that $|k| \leq 4001$ while Theorem 4.45 asserts that $0.93 \leq |k| \leq \sqrt{4001}$ for all real roots k of f because $m = \sqrt{100025}(\sqrt{1000.25} + 1000.25)$ and $l = \frac{1}{2m^2}(-4003000.5(m - 1000) + \sqrt{16008001 \cdot 1001000.25(m - 1000)^2 + 16004000m^3}\,) \approx 0.93160$.

Of more interest to us is the question whether the theorem also performs better when aiming for our use case of an upper bound of exactly one. The SMT solver of our choice claims that it actually does not.

Example 4.48. The bound u equals 1 only if $\frac{a_n - a_0 + |a_0|}{|a_n|} = 1$. Giving the following formula as input to a SMT-solver

```
1  (declare-fun an () Real) ; a_n
2  (declare-fun am () Real) ; a_{n-1}
3  (declare-fun ao () Real) ; a_0
4  (define-fun absolute ((x Real)) Real
5  (ite (>= x 0) x (- x)))
6  (assert (>= an am))
7  (assert (>= am ao))
8  (assert (not (= an 0)))
9  (assert (not (= (/ (+ (- an ao) (absolute ao)) (absolute an)) 1)))
10 (assert (= 1 (+ (* (/ (- an am) 2) (- (^ (absolute an) (- 1)) (^ (+ (- an ao) (absolute
   ao)) (- 1)))) (^ (+ (* (^ (/ (- an am) 2) 2) (^ (- (^ (absolute an) (- 1)) (^ (+ (- an ao
   ) (absolute ao)) (- 1))) 2)) (/ (+ (- an ao) (absolute ao)) (absolute an))) (/ 1 2)))))
11 (check-sat)
```

returns *unsat*, while omitting line 9 makes the formula satisfiable. This suggests that for an upper bound of 1, Theorem 4.39 and Theorem 4.45 coincide.

Before doing joint work with Zargar, Aziz released together with Mohammad two early papers based on the Eneström-Kakeya Theorem [6, 7]. They immediately introduce an improvement for polynomials with strictly decreasing coefficients.

Theorem 4.49. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial whose coefficients are strictly positive. Then f has no real root with absolute value greater* $\max \frac{a_0}{a_1}, \frac{a_1}{a_2} \ldots, \frac{a_{n-1}}{a_n}$.

If we set $\max \frac{a_0}{a_1}, \frac{a_1}{a_2} \ldots, \frac{a_{n-1}}{a_n}$ to 1, we get Constraint 4.70.

A generalization of both Theorem 4.49 and Theorem 4.39 (with a_0 set to 0) is the following:

Theorem 4.50. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial whose coefficients are strictly positive and let a_{-1} and a_{n+1} be defined as zero. Further let m_0, m_1 be non-negative real numbers. W.l.o.g. let $m_0 > m_1$. Let $\forall_{i=0}^{n} m_0 m_1 a_{i+1} + (m_0 - m_1)a_i - a_{i-1} \geq 0$. Then f has no real root with absolute value greater m_0.*

When stating the theorem, Aziz and Mohammad note that the specializations are achieved by setting $m_1 = 0$ or $m_0 = 1$, $m_1 = 0$, respectively.

Next they prove a theorem that shows that the interval the bounded roots of Theorem 4.37 must lie in so they may have multiplicity > 1 is constrained. This could be interpreted as a hint to limited performance of the (original) Eneström-Kakeya Theorem in the context of our experiments, since TRS of non-linear derivational complexity can only be bounded by matrix interpretations whose characteristic polynomials have multiple roots.

Theorem 4.51. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial of degree n whose coefficients are strictly positive. Further, let $a_n \geq a_{n-1} \geq \ldots \geq a_0$. Then all real roots of f with absolute value greater $\frac{n-1}{n}$ are simple.*

In the rest of the paper Aziz and Mohammad further restrict the interval $[0; 1]$ the roots can lie in when the hypothesis of the Eneström-Kakeya Theorem is fulfilled; by defining additional subintervals that are free of roots. This is not significant for our work, since we are perfectly complacent with the interval $[0; 1]$, further limiting the interval would provide no additional benefit to our ability to find compatible matrix interpretations suited for the complexity analysis.

In the second paper we first see a consequence of the Eneström-Kakeya Theorem that is obtained by substituting the indeterminate x by the fraction $\frac{x}{m}$ for some m.

Theorem 4.52. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial whose coefficients are strictly positive. Further, let m be a real number and $a_n \geq m a_{n-1} \geq \ldots \geq m^{n-1} a_1 \geq m^n a_0$. Then f has no real root with absolute value greater $\frac{1}{m}$.*

Upon closer inspection we can see that this corollary is of limited interest to us, since for an upper bound of one it coincides with the original theorem it is devised from.

Probably of more interest is a development of Theorem 4.51. Note that for the goal of this thesis we want to avoid having simple roots only (cf. Theorem 6.7).

Theorem 4.53. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial of degree $n \geq 2$ whose coefficients are strictly positive and let m be a strictly positive real number. Further, let $\forall_{i=1}^{n-1} m a_{i+1} \geq a_i$. Then all real roots of f with absolute value greater $\frac{m(n-1)}{n}$ are simple.*

They also introduce a theorem that could serve as an alternative to Theorem 4.52. If we have a polynomial with non-positive coefficients and we still desire a bound that is smaller than one, this condition may be of use.

Theorem 4.54. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial of degree n and let m be a strictly positive real number. Further, let $\forall_{i=0}^{n-1} |a_n| \geq m^{n-i} |a_i|$ and let k be the highest-valued strictly positive root of the polynomial $g(x) = x^{n+1} - 2x^n + 1$. Then f has no real root with absolute value greater $\frac{k}{m}$.*

The theorem can be stated with a more precise bound:

Theorem 4.55. *Let* $f(x) = (\sum_{i=0}^{m_0} a_i x^i) + a_n x^n$ *be a polynomial of degree* n *where* $0 \leq m_0 \leq n-1$, $a_{m_0} \neq 0$, $a_n \neq 0$, *and let* m_1 *be a strictly positive real number. Further, let* $\forall_{i=0}^{m_0} |a_n| \geq m_1^{n-i}|a_i|$ *and let* k *be the highest-valued strictly positive root of the polynomial* $g(x) = x^{n+1} - x^n - x^{m_0+1} + 1$. *Then* f *has no real root with absolute value greater* $\frac{k}{m}$.

This is then later followed by a variant of Theorem 4.52 that weakens the hypothesis to include polynomials with coefficients of arbitrary sign.

Theorem 4.56. *Let* $f(x) = \sum_{i=0}^{n} a_i x^i$ *be a polynomial and let* m_0 *be a natural number smaller or equal to* n. *Further, let* m_1 *be a strictly positive real number and let* $m_1^n|a_n| \leq m_1^{n-1}|a_{n-1}| \leq \ldots \leq m_1^{m_0+1}|a_{m_0+1}| \leq m_1^{m_0}|a_{m_0}| \geq m_1^{m_0-1}|a_{m_0-1}| \geq \ldots \geq m_1|a_1| \geq |a_0|$. *Then* f *has no real root with absolute value greater* $\frac{2m_1^{m_0+1}|a_{m_0}|}{m_1^n|a_n|} - m_1 + 2\sum_{i=0}^{n} \frac{|a_i - |a_i||}{m_1^{n-i-1}|a_n|}$.

The simpler Theorem 4.44 which we already mentioned earlier as part of Govil and Jain's work is a direct corollary. We set $m_0 = 0$ and $m_1 = 1$ and choose $x^n f(\frac{1}{x})$ as the polynomial, which maps a coefficient a_i of f to a_{n-i}.

Aziz and Mohammad reference Theorem 3.2 and with the help of it develop the following result.

Theorem 4.57. *Let* $f(x) = (\sum_{i=0}^{m} a_i x^i) + a_n x^n$ *be a polynomial of degree* n *where* $0 \leq m \leq n-1$. *Further, let* c *be a strictly positive real number. Then* f *has no real root of absolute value greater than* $\max(c, \sum_{i=0}^{m}(c^{i-n+1} \cdot |\frac{a_i}{a_n}|))$.

Aziz and Mohammad base their proof of Theorem 4.54 on this theorem. The theorem is quite adequate when trying to bound the roots to absolute values smaller equal one. Particularly interesting for us is the case where we set the real number c to 1 (since we very much wish for roots of value exactly 1), albeit having more leeway when demanding nothing more than $c \leq 1$ would give us greater flexibility when choosing coefficients that satisfy the constraint. The resulting constraint is not overly big or complex, regardless of whether we choose to restrict c to $c = 1$ or $0 < c \leq 1$. Confer Constraint 4.71.

The next theorem is at its most useful when used on polynomials with complex coefficients but it still holds some merit in its simplified form adapted to polynomials with real coefficients.

Theorem 4.58. *Let* $f(x) = \sum_{i=0}^{n} a_i x^i$ *be a non-constant polynomial with real coefficients and let* m *be a strictly positive real number. Then if* k *is a real root of* f *we can assert that* $x \leq me^{\imath\alpha} - \frac{m}{2n}$ *or* $x \geq me^{\imath\alpha} + \frac{m}{2n}$, *where* $\max(|f(-m)|, |f(m)|) = |f(me^{\imath\alpha})|$ *and thus* $\alpha \in \{\, y\pi \mid y \in \mathbb{Z} \,\}$.

By checking for compliance of the polynomial $x^n f(\frac{1}{x})$ to Theorem 4.58 we can in case of compatibility then use the reciprocal of the lower bound of the strictly positive, root-free interval for $x^n f(\frac{1}{x})$ as upper bound of the strictly positive, root-free interval for f, while the reciprocal of the upper bound of the interval for $x^n f(\frac{1}{x})$ becomes

the lower bound of the interval for f. This is due to the fact that when multiplying a function with the factor x^n we are at most adding new but always retaining the existing roots, and with $x > 0$ we have if $\frac{1}{x}$ is smaller equal a real number then x is greater equal to this number. Since $me^{\imath\alpha} - \frac{m}{2n}$ and $me^{\imath\alpha} + \frac{m}{2n}$ are strictly positive if $\max(|f(-m)|, |f(m)|) = |f(m)|$ and strictly negative if $\max(|f(-m)|, |f(m)|) = |f(-m)|$ for non-constant polynomials, we need to analyse $x^n f(-\frac{1}{x})$ instead and then negate the obtained bounds if we have $\max(|f(-m)|, |f(m)|) = |f(-m)|$ for $x^n f(\frac{1}{x})$. By negating the bounds the upper bound of the interval again becomes the lower bound and vice versa. Incidentally, the resulting interval coincides with the interval we obtained where $\max(|f(-m)|, |f(m)|) = |f(m)|$, making the distinction between $\max(|f(-m)|, |f(m)|) = |f(m)|$ and $\max(|f(-m)|, |f(m)|) = |f(-m)|$ unnecessary.

Corollary 4.59. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a non-constant polynomial with real coefficients and let m be a strictly positive real number. Let $\max(|x^n f(-\frac{1}{m})|, |x^n f(\frac{1}{m})|) = |x^n f(\frac{1}{m})|$ and let $\max(|f(-m)|, |f(m)|) = |f(m')|$ where $m' \in \{-m, m\}$. Then if k is a strictly positive real root of f we can assert that k is neither in $(\frac{1}{m+\frac{m}{2n}}, \frac{1}{m-\frac{m}{2n}})$ nor in $(m' - \frac{m}{2n}, m' + \frac{m}{2n})$.*

Alas, no matter the choice in m the theorem will not give us guarantees about the interval $(1, \infty)$.

For polynomials with non-negative coefficients Theorem 4.56 becomes:

Theorem 4.60. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with non-negative coefficients and a non-zero leading coefficient, and let m_0 be a natural number smaller or equal to n. Further, let m_1 be a strictly positive real number and let $m_1^n a_n \leq m_1^{n-1} a_{n-1} \leq \ldots \leq m_1^{m_0+1} a_{m_0+1} \leq m_1^{m_0} a_{m_0} \geq m_1^{m_0-1} a_{m_0-1} \geq \ldots \geq m_1 a_1 \geq a_0$. Then f has no real root with absolute value greater $\frac{2m_1^{m_0-n+1} a_{m_0}}{a_n} - m_1$.*

This prompted Gardner and Govil [38] to propose the following lower and upper bounds:

Theorem 4.61. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with a non-zero leading coefficient and let m_0 be a natural number smaller or equal to n. Further, let m_1 be a non-negative real number and let $m_1^n a_n \leq m_1^{n-1} a_{n-1} \leq \ldots \leq m_1^{m_0+1} a_{m_0+1} \leq m_1^{m_0} a_{m_0} \geq m_1^{m_0-1} a_{m_0-1} \geq \ldots \geq m_1 a_1 \geq a_0$. Then f has no real root with absolute value smaller than $\min(\frac{m_1 |a_0|}{2m_1^{m_0} a_{m_0} - a_0 - m_1^n (a_n - |a_n|)}, m_1)$ or greater than*

$$\max((m_1^{n+1}|a_0| - m_1^{n-1} a_0 - m_1 a_n + m_1^{n-m_0+1} a_{m_0} + m_1^{n-m_0-1} a_{m_0} + \sum_{i=1}^{m_0-1} (m_1^{n-i+1} a_i - m_1^{n-i-1} a_i) + \sum_{i=m_0+1}^{n-1} (m_1^{n-i-1} a_i - m_1^{n-i+1} a_i))/|a_n|, \frac{1}{m_1}).$$

Setting the variables m_0 and m_1 to specific values gives us several useful simpler bounds, some of which we have already seen before.

Corollary 4.62. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with a non-zero leading coefficient. Further, let $a_n \geq a_{n-1} \geq \ldots \geq a_0$. Then f has no real root with absolute value smaller than $\frac{|a_0|}{a_n - a_0 + |a_n|}$ or greater than $\frac{|a_0| - a_0 + a_n}{|a_n|}$.*

Corollary 4.63. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with a non-zero leading coefficient. Further, let $a_n \leq a_{n-1} \leq \ldots \leq a_0$. Then f has no real root with absolute value smaller than $\frac{|a_0|}{-a_n + a_0 + |a_n|}$ or greater than $\frac{|a_0| + a_0 - a_n}{|a_n|}$.*

Similar in form but based on a different hypothesis than Theorem 4.60 is a bound by Dewan and Bidkham [26].

Theorem 4.64. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with a non-zero leading coefficient, and let m_0 be a natural number smaller or equal to n. Further, let m_1 be a strictly positive real number and let $m_1^n a_n \leq m_1^{n-1} a_{n-1} \leq \ldots \leq m_1^{m_0+1} a_{m_0+1} \leq m_1^{m_0} a_{m_0} \geq m_1^{m_0-1} a_{m_0-1} \geq \ldots \geq m_1 a_1 \geq a_0$. Then f has no real root with absolute value greater $\frac{2 m_1^{m_0-n+1} a_{m_0} - m_1 a_n}{|a_n|} + \frac{|a_0| - a_0}{m_1^n}$.*

For the case where m_1 can be set to one Dewan and Bidkham then also propose a tighter bound that is an extension of Theorem 4.45 and a refinement of the bound presented by Joyal et. al (cf. Thm 4.39).

Theorem 4.65. *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial with a non-zero leading coefficient, $u = \frac{|a_n - a_{n-1}|}{2}(|a_n|^{-1} - (-a_n + 2a_{m_0} - a_0 + |a_0|)^{-1}) + [(\frac{|a_n - a_{n-1}|}{2})^2 \cdot (|a_n|^{-1} - (-a_n + 2a_{m_0} - a_0 + |a_0|)^{-1})^2 + \frac{-a_n + 2a_{m_0} - a_0 + |a_0|}{|a_n|}]^{1/2}$, $m_1 = u^n \cdot (u|a_n| + 2a_{m_0} - a_n - a_0)$ and $l = \frac{1}{2m_1^2} \cdot \{-u^2(a_1 - a_0)(m_1 - |a_0|) + [u^4(a_1 - a_0)^2(m_1 - |a_0|)^2 + 4|a_0|u^2 m_1^3]^{1/2}\}$. Further, let m_0 be a natural number smaller or equal n and let $a_n \leq a_{n-1} \leq \ldots \leq a_{m_0+1} \leq a_{m_0} \geq a_{m_0-1} \geq \ldots \geq a_0$. Then if k is a real root of f we can assert that $l \leq |k| \leq u$.*

4.8.1 Constraints Based on the Eneström-Kakeya Theorem

Constraint 4.66 (Theorem 4.39)**.** *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial built from real numbers satisfying the formula*

$$\exists a_0 \exists a_1 \ldots \exists a_n.$$
$$(a_n \geq a_{n-1} \wedge a_{n-1} \geq a_{n-2} \wedge \ldots \wedge a_1 \geq a_0 \wedge a_n > 0 \wedge a_0 \geq 0) \vee$$
$$(a_n < 0 \wedge a_n = a_{n-1} \wedge a_n = a_{n-2} \wedge \ldots \wedge a_n = a_0)$$

Then f has no real root with absolute value greater one.

Constraint 4.67 (Theorem 4.40)**.** *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial built from*

real numbers satisfying the formula

$$\begin{aligned}&\exists a_0 \exists a_1 \ldots \exists a_n \exists m.\\ &(m \cdot a_n \geq a_{n-1} \wedge a_{n-1} \geq a_{n-2} \wedge \ldots \wedge a_1 \geq a_0 \wedge m \geq 1 \wedge a_n > 0 \wedge a_0 \geq 0) \vee\\ &(a_n < 0 \wedge m \cdot a_n \geq a_{n-1} \wedge a_{n-1} \geq a_{n-2} \wedge \ldots \wedge a_1 \geq a_0 \wedge m \geq 1 \wedge a_0 = -m)\end{aligned}$$

Then f has no real root greater one.

Constraint 4.68 (Theorem 4.41)**.** *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial built from real numbers satisfying the formula*

$$\begin{aligned}&\exists a_0 \exists a_1 \ldots \exists a_n \exists m_0.\\ &m_0 \cdot a_n \geq a_{n-1} \wedge a_{n-1} \geq a_{n-2} \wedge \ldots \wedge a_1 \geq a_0 \wedge m_0 \geq 1 \wedge a_n > 0 \wedge a_0 \geq 0\end{aligned}$$

Then f has no real root greater one.

Constraint 4.69 (Theorem 4.42)**.** *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial built from real numbers satisfying the formula*

$$\begin{aligned}&\exists a_0 \exists a_1 \ldots \exists a_n \exists m_1.\\ &m_1 > 0 \wedge m_1 \leq 1 \wedge a_n \neq 0 \wedge\\ &[\ (a_n \leq a_{n-1} \wedge a_{n-1} \geq a_{n-2} \wedge \ldots \wedge a_1 \geq m_1 \cdot a_0 \wedge\\ &\quad (a_0 = 0 \vee (m_1 = 1 \wedge a_0 > 0))) \vee\\ &\quad (a_n \leq a_{n-1} \wedge a_{n-1} \leq a_{n-2} \wedge a_{n-2} \geq a_{n-3} \wedge \ldots \wedge a_1 \geq m_1 \cdot a_0 \wedge\\ &\quad a_{n-1} - a_{n-2} + a_0 - m_1 a_0 = 0) \vee\\ &\qquad\qquad\qquad\qquad \vdots\\ &\quad (a_n \leq a_{n-1} \wedge a_{n-1} \leq a_{n-2} \wedge \ldots \wedge a_2 \leq a_1 \wedge a_1 \geq m_1 \cdot a_0 \wedge\\ &\quad a_{n-1} - a_1 + a_0 - m_1 a_0 = 0) \vee\\ &\quad (a_n \leq a_{n-1} \wedge a_{n-1} \leq a_{n-2} \wedge \ldots \wedge a_2 \leq a_1 \wedge a_1 \leq m_1 \cdot a_0 \wedge\\ &\quad (a_{n-1} = 0 \vee (m_1 = 1 \wedge a_0 > 0)))\ ]\end{aligned}$$

Then f has no real root greater one.

Constraint 4.70 (Theorem 4.49)**.** *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial built from*

real numbers satisfying the formula

$$\begin{aligned}&\exists a_0 \exists a_1 \ldots \exists a_n.\\ &a_n > 0 \wedge a_{n-1} > 0 \wedge \ldots \wedge a_0 > 0 \wedge\\ &\frac{a_{n-1}}{a_n} \leq 1 \wedge \frac{a_{n-2}}{a_{n-1}} \leq 1 \wedge \ldots \wedge \frac{a_0}{a_1} \leq 1 \wedge\\ &(\frac{a_{n-1}}{a_n} = 1 \vee \frac{a_{n-2}}{a_{n-1}} = 1 \vee \ldots \vee \frac{a_0}{a_1} = 1)\end{aligned}$$

Then f has no real root of absolute value greater one.

Constraint 4.71 (Theorem 4.57)**.** *Let $m < n$ be two natural-numbered constants and let $f(x) = (\sum_{i=0}^{m} a_i x^i) + a_n x^n$ be a polynomial built from real numbers satisfying the formula (if satisfiable)*

$$\exists a_0 \exists a_1 \ldots \exists a_m \exists a_n \exists r.$$
$$a_n > 0 \wedge r > 0 \wedge r \leq 1 \wedge r^{1-n} \cdot |\frac{a_0}{a_n}| + r^{2-n} \cdot |\frac{a_1}{a_n}| + \ldots + r^{m+1-n} \cdot |\frac{a_m}{a_n}| \leq 1$$

Then f has no real root with absolute value greater one.

5 Real Root Counting

The majority of theorems in this chapter have been formally proven by Li and Paulson [57]. For a gentle introduction to the topic of real root isolation we point to the works of Elgyütt [32] and Biagioli [15].

This chapter discusses the problem of constraining the roots of a polynomial to a specific interval from a different perspective than the previous chapter. While in the previous chapter we had a look at a multitude of theorems giving very specific bounds on the value of the roots of a polynomial, this chapter will instead focus on the two major root isolation procedures. We will evaluate whether we can employ the algorithms for our own purpose, which rather than the search for roots is the confinement of roots to a specific interval. To this end we extract the root counting subroutines from the root isolation algorithms and transform these to logical formulae that codify our demand on the absolute value of the polynomial roots.

Being a topic with a vast array of practical applications, real root isolation has been an area of active research in the discipline of both mathematics and computer science. The foundation for plenty of the root isolation algorithms known today can be traced back to a theorem published as early as 1637 by René Descartes in his book 'La Géométrie' [25]. In the 19th century two theorems, by Charles-François Sturm and A. J. H. Vincent, laid the foundation for the first complete real root counting procedures. The ubiquity of computing devices coupled with the accessibility and power of modern computer algebra systems has made the algorithms derived from some of these early 19th-century theorems a practical tool for mathematicians and computer scientists alike.

Real root isolation describes the process of separating intervals that each contain exactly one root. This is fundamentally useful for root finding algorithms that rely on the isolation mechanism to check whether all real roots have been found. As many root finding methods rely on it, the desire to find a root isolation process that is as efficient as possible lead to the development of a host of different algorithms that found their way into various computer algebra systems. The most well known of these algorithms are based on either Sturm's (most notably the Bisection method) or Vincent's Theorem. And although Vincent's Theorem has been generalized by Wang to a form that does not demand squarefreeness of the polynomial, as far as we know, the popular algorithms based on Vincent's Theorem such as VAS use squarefree factorization instead of Wang's Theorem. Squarefree factorization works for any input polynomial and there are methods that are not overly computationally expensive. This suggests doing a squarefree factorization as a preprocessing step and invoking the actual Vincent's-Theorem–based algorithm on each factor is most likely more efficient than an algorithm built upon Wang's Theorem. But we need to keep in mind that the goal of this thesis deviates from that of a root isolation procedure. We are not interested in the algorithms' performance on numerical input but

rather in keeping the size and complexity of the logical formulae we can extract from the algorithms as low as possible. In the context of the research question this thesis tries to investigate, Wang's Theorem remains an interesting option to explore.

Indeed, we do not even need the full power of real root isolation algorithms. We are content with counting the number of roots in a given interval. Most real root isolation algorithms can be altered to do just that, but in many cases they can be made considerably less complex in the process. Real root counting is in general the computationally cheaper operation, since many modern root isolation algorithms use either root counting or root bounding methods as subroutines. Often, the distinguishing factor between root isolation algorithms is just the method used to count, or rather bound, the number of roots in an interval. Thus the root counting methods we will present in this chapter build on the same foundations as the isolation procedures do: Vincent's and Sturm's Theorems and variants thereof. Before fully delving into the historical foundations for the root counting procedures we intend to present, we will discuss an alternative approach to limit the value of non-negative roots based on geometric observations that is detailed in Method (B) of [63]. It is this simple arithmetic constraint that we will pitch the root counting algorithms against in a quest to most efficiently bound the non-negative roots to a value of maximum one.

5.1 A direct approach

The following observations only apply to cubic polynomials and are due to [63].

Inspecting the discriminant lets us determine the number of real roots. The following lemmata are well-known.

Lemma 5.1. *Let Δ_2 be the discriminant of a quadratic polynomial f. Then all roots of f are real and f is squarefree if and only if Δ_2 is strictly positive, all roots of f are real and f has a multiple root if and only if Δ_2 is zero, and f has no real root if and only if Δ_2 is strictly negative.*

Lemma 5.2. *Let Δ_3 be the discriminant of a cubic polynomial f. Then all roots of f are real and f is squarefree if and only if Δ_3 is strictly positive, all roots of f are real and f has a multiple root if and only if Δ_3 is zero, and f has one real root if and only if Δ_3 is strictly negative.*

Naturally the most interesting case for us is $\Delta_3 = 0$ since we desire to have at least one multiple root so that we can derive complexity bounds of higher degree for a given rewrite system. Let us first discuss the case of a single real root. We demand that the polynomial at position negative one, times the polynomial's leading coefficient is smaller equal zero and that the polynomial at position one, times the polynomial's leading coefficient is greater equal zero. Then the only sign change and thus the only root of the polynomial lies in the desired interval $[-1; 1]$. Now if we have three (not-necessarily distinct) real roots, we know by observing the function's geometrical form that the first derivative changes its sign in between the roots *and in between the roots only.* And since our aim is to constrain the real roots to the interval $[-1; 1]$ we need to ensure that the first derivative

of the polynomial function does not change its sign outside the interval. Observe that the first derivative is a quadratic function. Opting to constrain its discriminant allows the following case distinction. For a strictly negative discriminant there will be no sign change at all for there are no real roots. If the discriminant is zero the sign of the leading coefficient of the cubic polynomial decides whether the image of the derivative is comprised of purely strictly positive or strictly negative numbers, but then again there will be no sign change. In case of a strictly positive discriminant, we have to further demand that $|a_2| \leq 3$ and that the first derivative at position -1 and at position 1 is greater equal zero (smaller equal zero if the leading term of the characteristic polynomial were strictly negative, i.e. if the characteristic polynomial were not normalized). The rest of the constraint is then the same as if Δ_3 were strictly negative.

There are of course other ways to express the solution for the quadratic, cubic and quartic equations in terms of radicals, as well as various other geometric approaches and iterative methods. A short but thorough overview is given in [59]. We especially want to highlight the relevance of the unified approach of Ungar [72] in this context.

5.1.1 Constraints Based on Geometrical Observations by Neurauter et al.

Constraint 5.3 ([63], Method (B))**.** *Let* $f(x) = x^3 + \sum_{i=0}^{2} a_i x^i$ *be a normalized polynomial built from real numbers satisfying the formula*

$$
\begin{aligned}
&-1 - a_2 - a_1 + a_0 \leq 0 \wedge a_2 + a_1 + a_0 + 1 \geq 0 \wedge \\
&((a_2)^2(a_1)^2 - 4(a_1)^3 - 4(a_2)^3 a_0 - 27(a_0)^2 + 18 a_2 a_1 a_0 < 0 \vee (a_2)^2 - 3a_1 \leq 0 \vee \\
&(-3 \leq a_2 \leq 3 \wedge -a_1 - 3 \leq 2a_2 \leq a_1 + 3))
\end{aligned}
$$

Then f *has no real root with absolute value greater one.*

5.2 Sturm

5.2.1 Fourier's Theorem

The basis of Sturm's method is *Fourier's Theorem for polynomials*, not to be confused with the more widely known Fourier Theorem that is about periodic functions. Fourier had taught his theorem several years before eventually publishing it in 1820 [35]. A contemporary of his, François Budan, independently formulated an equivalent theorem. This eventually lead to a dispute on first-authorship. The formulation of the theorem as stated by Fourier can be found in the literature under various names, including Fourier's Theorem, Fourier-Budan Theorem, Budan-Fourier Theorem, and surprisingly even as Budan's Theorem ([29, 77], as described in [1]). The theorem of Budan and effective equivalence of the two theorems will be topic of a later section.

The importance of Fourier's Theorem cannot be overstated, Sturm himself acknowledged

quite openly that the it was this theorem that sparked the very idea for his results.[1]

Structurally, Fourier's Theorem shows a strong resemblance to Descartes' Rule of Signs (cf. Theorem 4.4), bounding the number of roots in an interval by the number of sign changes, and if there is a discrepancy between the bound and the actual number of roots it has to be even. Fourier's Theorem works with a sequence of derivatives of the original polynomial instead of just the original polynomial as Descartes' Rule of Signs does, which allows us, and this is the big difference to Descartes' Rule of Signs, to bound the number of roots in an arbitrary interval, instead of being constrained to the interval $(0, \infty)$, or $(-\infty, 0)$ respectively. This makes Fourier's Theorem a lot more powerful, and much more widely applicable. As we progress further into this chapter, we will see a theorem that allows us to determine an explicit number for the roots in an interval and not merely an upper bound. I.e., with help of this theorem we will learn how to count roots instead of 'guesstimating' them.

In the preliminaries we defined the process of counting the number of sign variations in a sequence of real numbers. In Section 5.2.2 we will elaborate in detail on how to count the sign variations in a sequence of polynomials. To anticipate, we count the number of sign variations *for a specific point* p by evaluating the polynomials at p, which leaves us with a real sequence, off which we then can read the changes easily. The sequence of polynomials that is of interest for Fourier's Theorem is the initial polynomial expression alongside its multiple iterative derivatives. This simple polynomial sequence allowed Fourier to transform Descartes' rule into a template for a much more powerful algorithmic construct.

Definition 5.4. Let f be a non-constant polynomial in x of degree n. The Fourier sequence of derivatives D_f is the sequence $f, f', \ldots, f^{(n)}$.

Theorem 5.5 (Fourier)**.** *Let f be a non-constant polynomial in x, D_f its corresponding Fourier sequence of derivatives and let $p, q \in \mathbb{R}$ such that $p < q$. Then the number of sign variations of D_f at position p is greater or equal to the number of sign variations of D_f at position q. Furthermore, the number of roots of f in the interval $(p, q]$ cannot exceed this difference. If there is a discrepancy between this bound and the actual number of roots, then the difference has to be even.*

5.2.2 Sign Variations, Sturm Sequences and Root Isolation Algorithms

Before defining the well-known Sturm Sequences we discuss how to extend the definition of sign variations to sequences *of univariate polynomials*. The basic concept to obtain the number of sign variations **at position** n is to evaluate the polynomials for n and then proceed as with a sequence of numbers following the procedure outlined in the preliminaries. So in short, from the sequence of numbers we form a new sequence

[1] In his words: *'C'est en m'appuyant sur les principes qu'il a posés, et en imitant ses démonstrations, que j'ai trouvé les nouveaux théorèmes que je vais énoncer.'*, which translates to *'It is by relying upon the principles he has laid out and by imitating his proofs that I have found the new theorems which I am about to present.'* [14].

excluding the zeroes, and then traverse this new sequence and add one to the result every time the sign changes from positive to negative or vice versa.

Definition 5.6. Let $p \in \mathbb{R} \setminus \{-\infty, \infty\}$ and let $(f_i)_{i<k}$ with $k \in \mathbb{N}$ be a sequence of univariate polynomials over $\mathbb{R}$. Then the number of sign variations at position p is the number of sign variations in the real-numbered sequence $(f_i(p))_{i<k}$.

Example 5.7. Consider the following sequence:

$$(f_0, f_1, f_2, f_3) = (x^2 + 4, 3x^6 - 2x^4 - x^2, -8, 2x^4 - 6x^2)$$

To calculate the number of sign variations at position 2 we first evaluate each polynomial on this position. This gives rise to the sequence of scalars

$$(f_0(2), f_1(2), f_2(2), f_3(2)) = (8, 156, -8, 8)$$

Since only $f_0(2)f_1(2) > 0$, while both $f_1(2)f_2(2) < 0$ as well as $f_2(2)f_3(2) < 0$, we have two sign changes in the original sequence of polynomials for position 2. At position 1, however, we would have a different result – namely merely one sign change –, as we have to consider that $f_3(1) < 0$, while $f_1(1) = 0$ and thus is omitted from the calculation.

We further define sign variations in the sequence consisting of the signs of the leading terms of the polynomials to be the number of sign variations of this sequence of polynomials *at position* ∞. By negating the sign of all leading terms of odd power and calculating the number of sign differences of the sequence of leading terms modified this way, we obtain the number of sign changes for the sequence of polynomials *at position* $-\infty$. The two definitions coincide if all terms are of either purely odd or purely even power. Let us see an example illustrating these definitions:

Example 5.8. We continue Example 5.7. What about the number of sign changes at ∞? As a first step in our attempt to answer this question we have to transform the original sequence into a sequence of the leading terms of the polynomials. For our example, at the end of this step, we have the sequence $(x^2, 3x^6, -8, 2x^4)$. When determining the value of this monomials at position 1 we obtain $(1, 3, -8, 2)$, in other words the leading coefficients of the original polynomials. Now the result is easy to see, we apparently have two sign changes at position ∞. Since f_0, f_1, f_2, f_3 only consist of terms of even degree, the number of sign changes at $-\infty$ coincides with our result. Let us consider another sequence, with terms of both odd and even degree constituting its polynomials. This sequence shall be $(5x^3 + x^2, -2x^2 + 1, 18x + 8)$. The sign of the leading term changes twice over the course of the sequence, which means the number of sign changes at ∞ is 2. For position $-\infty$ we evaluate the leading monomials at the point -1, in other words we take the signs of the monomials and invert them for terms of odd power. This leaves us with no sign change at all.

To formally define the sign of the leading term, we use the signum function *sgn* in conjunction with *leadterm*, a function which given a polynomial extracts the leading monomial. The precise sequence of steps is as follows: Extracting the leading monomial

with *leadterm*, evaluating it at 1 or -1 for the number of sign variations at ∞ and $-\infty$ respectively, and then computing the sign with *sgn*. This is the process by which we will compute the sign of the leading monomial of a polynomial. We will label the corresponding function sgn_{lc}.

Definition 5.9.

$$\begin{aligned}&\operatorname{sgn} : \mathbb{R} \to \{\, x \in \mathbb{R} : |x| = 1 \,\} \cup \{0\} \\ &x \mapsto \operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0, \\ 1 & \text{if } x > 0, \\ 0 & \text{else.} \end{cases}\end{aligned}$$

Definition 5.10.

$$\begin{aligned}&\operatorname{leadterm} : \mathbb{R}[x] \to \mathbb{R}[x] \\ &f \mapsto \operatorname{leadterm}(f) = g, \text{ where } g \text{ is leading term of } f.\end{aligned}$$

Definition 5.11.

$$\begin{aligned}&\operatorname{lc} : \mathbb{R}[x] \to \mathbb{R} \\ &f \mapsto \operatorname{lc}(f) = c, \text{ where } c \text{ is coefficient of the leading term of } f.\end{aligned}$$

Definition 5.12. Let $(f_i)_{i<k}$ with $k \in \mathbb{N}$ be a sequence of k univariate polynomials over $\mathbb{R}$. Let $(g_i)_{i<k} = (leadterm(f_i))_{i<k}$ be the sequence of their leading monomials. Then the number of sign variations at position ∞ is the number of sign variations in the real-valued sequence $(g_i(1))_{i<k}$ and the number of sign variations at $-\infty$ is the number of variations in the sequence $(g_i(-1))_{i<k}$.

Book seven of *The Elements* by Euclid of Alexandria defines a method to find the highest common factor of two numbers. This is commonly known as the Euclidean algorithm. Its main operation is called Euclidean division, which is a division that produces both a whole-numbered quotient as well as a remainder. It is well known that the algorithm works also for polynomials, and it is the Euclidean division used in the algorithm that lays the groundwork to obtain the Sturm Sequence (cf. Figure 5.1).

The Sturm Sequence is a polynomial remainder sequence, i.e. after choosing the two starting elements each new element of the sequence is calculated by applying Euclidean division to obtain the remainder. Different to the classical Euclidean algorithm, though, we always negate the remainder before proceeding. This explains the origin of the name *Signed Remainder Sequence*, which is a generalized form of Sturm sequences for which the second input polynomial does not necessarily have to be the first derivative of the first input argument (cf. Section 5.2.5). We want to remark that in the literature one finds that often a form of pseudo-division is used to avoid fractions as remainders when working with integer polynomials. We call the remainders resulting from pseudo-divisions pseudo-remainders.

```
procedure euclidean_division(f, g):
    q ← 0
    q ← f
    while deg(r) ≥ deg(g) do
        i ← (lc(r)/lc(g)) x^(deg(r)−deg(g))
        q ← q + i
        r ← r − gi
    return (q, r)
```

Figure 5.1: Primitive Euclidean Division

We give a concise overview of the different options one has when calculating the highest common factor for two integer polynomials. We first want to outline the procedure when working in the fraction field and then compare it to the use of pseudodivision. We assume the reader is familiar with a method to compute the highest common factor of integers.

Let f and g be two integer polynomials.

Step one
Calculate the highest common factor in $\mathbb{Q}$ of f and g by repeatedly using the primitive euclidean division (Fig. 5.1).

Step two
Eliminate the denominators and calculate the primitive part in $\mathbb{Z}$ of the rational polynomial obtained in step one.

Step three
Calculate the highest common factor of the contents of f and g.

Then the highest common factor in $\mathbb{Z}$ of f and g is the product of the results of step two and step three.

Let us see the procedure in action:

Example 5.13. Consider the polynomials $f_0(x) = 7x^5 + 5x^2 - 3$ and $f_1(x) = 5x^3 + 1$. First we calculate the highest common factor in $\mathbb{Q}$. We have

$$\texttt{euclidean\_division}(f_0, f_1) = (\frac{7x^2}{5}, \frac{18x^2}{5} - 3 =: f_2)$$
$$\texttt{euclidean\_division}(f_1, f_2) = (\frac{25x}{18}, \frac{25x}{6} + 1 =: f_3)$$
$$\texttt{euclidean\_division}(f_2, f_3) = (\frac{108x}{125} - \frac{648}{3125}, -\frac{8727}{3125})$$

and thus

$$\gcd(f_0, f_1) = -\frac{8727}{3125}.$$

Since the gcd in $\mathbb{Q}$ calculated in step one is just a rational number the primitive part equates to 1: $\operatorname{cont}(-\frac{8727}{3125}) = \frac{\operatorname{cont}(-8727)}{3125} = -\frac{8727}{3125}$ and therefore $\operatorname{prim}(-\frac{8727}{3125}) = \frac{-\frac{8727}{3125}}{\operatorname{cont}(-\frac{8727}{3125})} = 1$. In step three we compute the contents of f_0 and f_1, which are $\gcd(7, 5, -3) = 1$ and $\gcd(5, 1) = 1$, and the gcd thereof (i.e. $\gcd(1, 1) = 1$). Thus the highest common factor in $\mathbb{Z}$ is

$$\gcd(f_0, f_1) = 1.$$

If instead of working with rationals, we decide to use pseudodivision and stay in $\mathbb{Z}$ the procedure is as follows:

Let f and g be two integer polynomials.

Step one
Starting from f and g, repeatedly apply the pseudo remainder operation (cf. Def. 5.16) to the last and second last element of the sequence of polynomials and append the resulting pseudoremainder to the sequence, until the next pseudoremainder would equate zero.

Step two
Calculate the primitive part of the last element of the sequence we computed in step one.

Step three
Calculate the highest common factor of the contents of f and g.

Then the highest common factor in $\mathbb{Z}$ of f and g is the product of the results of step two and step three.

A continuation of Example 5.13 illustrates the rapid coefficient growth inherent to the repeated use of naive pseudodivision.

Example 5.14. Let f_0 and f_1 be defined as in Example 5.13. First we repeatedly calculate pseudoremainders until reaching zero.

$$\begin{aligned}
f_0 \bmod_{\text{ps}} f_1 &= \texttt{euclidean_division}(5^3 f_0, f_1) \\
&= (175x^2, 450x^2 - 375 =: f_2) \\
f_1 \bmod_{\text{ps}} f_2 &= \texttt{euclidean_division}(450^2 f_1, f_2) \\
&= (2250x, 843750x + 202500 =: f_3) \\
f_2 \bmod_{\text{ps}} f_3 &= \texttt{euclidean_division}(843750^2 f_2, f_3) \\
&= (379687500x - 91125000, -248514960937500 =: f_4) \\
f_3 \bmod_{\text{ps}} f_4 &= \texttt{euclidean_division}((-248514960937500)^2 f_3, f_4) \\
&= (-2096844982910156250000x - 50324279589843750000, 0)
\end{aligned}$$

We calculate the primitive part of f_4: $\text{cont}(-248514960937500) = -248514960937500$ and therefore $\text{prim}(-248514960937500) = \frac{-248514960937500}{\text{cont}(-248514960937500)} = 1$. We then compute the contents of f_0 and f_1, which are $\gcd(7, 5, -3) = 1$ and $\gcd(5, 1) = 1$, as well as the highest common factor thereof (i.e. $\gcd(1, 1) = 1$). Thus the highest common factor in $\mathbb{Z}$ is

$$\gcd(f_0, f_1) = 1.$$

One way to reduce the coefficient growth that immediately comes to mind is to strip out the content of the polynomial remainder after each pseudodivision.

Let f and g be two integer polynomials.

Step one
Starting from f and g, repeatedly apply the pseudo remainder operation (cf. Def. 5.16) to the last and second last element of the sequence of polynomials and append the primitive part of the resulting pseudoremainder to the sequence, until the next pseudoremainder would equate zero.

Step two
Calculate the highest common factor of the contents of f and g.

Then the highest common factor in $\mathbb{Z}$ of f and g is the product of the last element of the sequence computed in step one and the factor computed in step two.

Example 5.15. Let f_0 and f_1 be defined as in Example 5.13. First we repeatedly calculate pseudoremainders until reaching zero, at each step dividing through the content of remainder.

$$\begin{aligned}
f_0 \bmod_{\text{ps}} f_1 &= \texttt{euclidean_division}(5^3 f_0, f_1) \\
&= (175x^2, 450x^2 - 375) \text{ and thus } f_2 = prim(450x^2 - 375) = 6x^2 - 5 \\
f_1 \bmod_{\text{ps}} f_2 &= \texttt{euclidean_division}(6^2 f_1, f_2) \\
&= (30x, 150x + 36) \text{ and thus } f_3 = prim(150x + 36) = 25x + 6 \\
f_2 \bmod_{\text{ps}} f_3 &= \texttt{euclidean_division}(25^2 f_2, f_3) \\
&= (150x - 36, -2909) \text{ and thus } f_4 = prim(-2909) = 1 \\
f_3 \bmod_{\text{ps}} f_4 &= \texttt{euclidean_division}(f_3, f_4) \\
&= (25x + 6, 0)
\end{aligned}$$

We calculate the contents of f_0 and f_1, which are $\gcd(7, 5, -3) = 1$ and $\gcd(5, 1) = 1$, and the highest common factor of these numbers (i.e. $\gcd(1, 1) = 1$). Thus the highest common factor in $\mathbb{Z}$ is

$$\gcd(f_0, f_1) = f_4 \cdot 1 = 1.$$

Alas, calculating the content in each step is computationally expensive. Section 5.2.5 explores a procedure that avoids the intermediary calculation of highest common factors of coefficients of the remainders but is still fairly effective at containing coefficient growth.

We will skip the original definition of Sturm Sequences and from the very beginning work with pseudo-remainders. The pseudo-remainder operation is defined as follows:

Definition 5.16. $f \bmod_{\text{ps}} g := \texttt{euclidean\_division}(\text{lc}(g)^{\deg(f)-\deg(g)+1} f, g)[1]$, where lc extracts the leading coefficient.

The goal is to obtain polynomial pseudo-remainder sequences that allow one to count the number of roots in the same sense polynomial remainder sequences do in the original Sturm Theorem. To achieve this one has to pay special attention to the signs. We define a modified pseudo-remainder operation that can directly substitute the remainder operation in the construction of the Sturm Sequence.

Definition 5.17. $f \bmod_{\text{ps1}} g := -\texttt{euclidean\_division}(|\text{lc}(g)|^{\deg(f)-\deg(g)+1} f, g)[1]$, where lc extracts the leading coefficient.

Multiplying the first argument with some number before each and every division leads to a rapid growth of input size when calculating longer sequences. It is thus desirable to reduce the size of the remainders after every step. For this we divide by the coefficients-reduction factor c_i, as described in [4].

Definition 5.18. Let s be a Sturm Sequence as defined by Definition 5.19 and let n be the length of this sequence. We define the coefficient-reduction factors $c_2, c_3, \ldots, c_{n-1}$ as follows:

$$
\begin{aligned}
&c_0 = 1 \\
&c_1 = 1 \\
&c_2 = (-1)^{d_2} \\
&\forall_{i>2}\ c_i = -\text{lc}(s_{i-2}) \cdot h_i^{d_i-1} \\
&h_2 = -1 \\
&\forall_{i>2}\ h_i = \frac{(-\text{lc}(s_{i-2}))^{d_{i-1}-1}}{h_{i-1}^{d_{i-1}-2}} \\
&\text{where } d_i = \deg(s_{i-2}) - \deg(s_{i-1}) + 1 \text{ and } s_{i-1} \text{ is} \\
&\text{the } i\text{-th element of the Sturm Sequence } s.
\end{aligned}
$$

Definition 5.19. Let f be a non-constant polynomial of degree n and f' its first derivative. The Sturm Sequence of f is the sequence $f, f', f \bmod_{\text{ps1}} f', (f \bmod_{\text{ps1}} f') \bmod_{\text{ps1}} f'' \ldots g$ where g is the highest common factor of f and f'.

Remark. The Sturm Sequence consists of at most n elements.

Sturm's Sequence allows us to determine the number of roots in a given interval. This is captured by the Sturm Theorem. The original Sturm Theorem requires the input polynomial to be squareefree, but many authors in the literature accompany the theorem with a variant that does without this restriction. The difference between the proofs of these two theorems is explained well in [71].

Theorem 5.20 (Sturm, [66])**.** *Let f be a non-constant, squarefree polynomial and let m and n be two distinct real numbers. Without loss of generality let $m < n$. Then the number of roots of f in the interval $(m, n]$ is the the same as the number of sign changes of the Sturm Sequence of f at position n subtracted from the number of sign changes of the Sturm Sequence of f at position m.*

Definition 5.21. Let f be a non-constant polynomial of degree n and f' its first derivative. The *Extended Sturm Sequence of f* is the sequence $(f_i)_{i<m}$ that is defined as

$$
\begin{aligned}
&f_0 = f \\
&f_1 = f' \\
&\forall_{i>1} \quad f_i = \begin{cases} f_{i-2} \bmod_{\text{ps1}} f_{i-1} & \text{if } f_{i-2} \bmod_{\text{ps1}} f_{i-1} \neq 0, \\ f'_{i-1} & \text{if } f_{i-2} \bmod_{\text{ps1}} f_{i-1} = 0 \\ & \quad \text{and } f'_{i-1} \neq 0, \\ f_{i-1} \text{ was the last element of the sequence.} & \text{else.} \end{cases}
\end{aligned}
$$

Theorem 5.22. *Let f be a non-constant polynomial and let m and n be two distinct real numbers. Without loss of generality let $m < n$. Further, let $f(m) \cdot f(n) \neq 0$. Then the number of (not necessarily distinct) roots of f in the interval $(m, n]$ is the the same as the number of sign changes of the Extended Sturm Sequence of f at position n subtracted from the number of sign changes of the Extended Sturm Sequence of f at position m.*

5.2.3 Constraints Based on Sturm's Theorem

Constraint 5.23 (Theorem 5.20)**.** *Let f be a polynomial over the reals and let $(f_i)_{i<m}$ be the Extended Sturm Sequence of the polynomial. Let $f(1) \neq 0$, and let the number of sign changes of $(f_i)_{i<m}$ at position 1 be the same as the number of sign changes of $(f_i)_{i<m}$ at position ∞. Then f has no real root with absolute value greater one.*

5.2.4 On the Relationship of Some Constraints

When focussing on characteristic polynomials of degree less than four, the similarity of the constraints of [63, Method (B)] and the Sturm constraints we discussed beforehand is immediately apparent. In this section we thus take a closer look at these seemingly similar logical formulae and sketch a constructive equivalence proof for characteristic polynomials of degree 3.

If, following the example of [63, Method (A)], we explicitly calculate the roots for normalized polynomials of degree two we get the following constraint as a result:

Constraint 5.24 ([63], Method (A))**.** *Let $f(x) = x^2 + \sum_{i=0}^{1} a_i x^i$ be a normalized polynomial built from real numbers satisfying the formula*

$$a_1 + 2 \geq 0 \wedge a_1 + a_0 + 1 \geq 0$$

Then f has no real root with absolute value greater one.

Let us contrast this with the constraints based on the original Sturm Theorem, instantiated by a normalized polynomial of degree two.

Constraint 5.25. *Let $f(x) = x^2 + \sum_{i=0}^{1} a_i x^i$ be a squarefree normalized polynomial built from real numbers satisfying the formula*

$$a_1 + 2 \geq 0 \wedge a_1 + a_0 + 1 \geq 0$$

Then f has no real root with absolute value greater one.

Observe that the only difference is that as pre-requirement to using Sturm's Theorem the polynomial in question must be squarefree.

We will now compare Constraint 5.3 with the Sturm-based constraints instantiated by a characteristic polynomial of degree three.

Constraint 5.26. *Let $f(x) = -x^3 + \sum_{i=0}^{2} a_i x^i$ be a squarefree polynomial built from real numbers satisfying the formula*

$$
\begin{aligned}
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 > 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 > 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g < 0) \vee \\
&(a_2 + a_1 + a_0 - 1 > 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 > 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 > 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 > 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 > 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g > 0) \vee \\
&(a_2 + a_1 + a_0 - 1 > 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 = 0) \vee \\
&(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 = 0) \vee
\end{aligned}
$$

$$
\begin{aligned}
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 < 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \land -(a_2)^2 - 3a_1 = 0) \lor \\
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 = 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \land -(a_2)^2 - 3a_1 = 0) \lor \\
&(a_2 + a_1 + a_0 - 1 = 0 \land 2a_2 + a_1 - 3 < 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \land -(a_2)^2 - 3a_1 = 0) \lor \\
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 < 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \land -(a_2)^2 - 3a_1 = 0) \lor \\
&(a_2 + a_1 + a_0 - 1 > 0 \land 2a_2 + a_1 - 3 > 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \land -(a_2)^2 - 3a_1 < 0 \land g = 0) \lor \\
&(a_2 + a_1 + a_0 - 1 = 0 \land 2a_2 + a_1 - 3 > 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \land -(a_2)^2 - 3a_1 < 0 \land g = 0) \lor \\
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 > 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \land -(a_2)^2 - 3a_1 < 0 \land g > 0) \lor \\
&(a_2 + a_1 + a_0 - 1 > 0 \land 2a_2 + a_1 - 3 = 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \land -(a_2)^2 - 3a_1 < 0 \land g = 0) \lor \\
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 = 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \land -(a_2)^2 - 3a_1 < 0 \land g > 0) \lor \\
&(a_2 + a_1 + a_0 - 1 = 0 \land 2a_2 + a_1 - 3 < 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \land -(a_2)^2 - 3a_1 < 0 \land g > 0) \lor \\
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 < 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \land -(a_2)^2 - 3a_1 < 0 \land g > 0) \lor \\
&(a_2 + a_1 + a_0 - 1 > 0 \land 2a_2 + a_1 - 3 > 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \land -(a_2)^2 - 3a_1 < 0 \land g = 0) \lor \\
&(a_2 + a_1 + a_0 - 1 = 0 \land 2a_2 + a_1 - 3 > 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \land -(a_2)^2 - 3a_1 < 0 \land g = 0) \lor \\
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 > 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \land -(a_2)^2 - 3a_1 < 0 \land g > 0) \lor \\
&(a_2 + a_1 + a_0 - 1 > 0 \land 2a_2 + a_1 - 3 = 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \land -(a_2)^2 - 3a_1 < 0 \land g = 0) \lor \\
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 = 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \land -(a_2)^2 - 3a_1 < 0) \lor \\
&(a_2 + a_1 + a_0 - 1 = 0 \land 2a_2 + a_1 - 3 < 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \land -(a_2)^2 - 3a_1 < 0) \lor \\
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 < 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \land -(a_2)^2 - 3a_1 < 0) \lor \\
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 = 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \land -(a_2)^2 - 3a_1 < 0) \lor \\
&(a_2 + a_1 + a_0 - 1 = 0 \land 2a_2 + a_1 - 3 < 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \land -(a_2)^2 - 3a_1 < 0) \lor \\
&(a_2 + a_1 + a_0 - 1 < 0 \land 2a_2 + a_1 - 3 < 0 \land -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \land -(a_2)^2 - 3a_1 < 0)
\end{aligned}
$$

where $g = 4(a_2)^3a_0 - 4(a_1)^3 - (a_2)^2(a_1)^2 + 27(a_0)^2 + 18a_2a_1a_0 = -\Delta_3$. *Then* f *has no real root with absolute value greater one.*

Conjecture 5.27 (Sturm ⇒ Method (B) [63])**.** *Constraint 5.26 implies Constraint 5.3.*

We sketch a possible starting point for a constructive proof of the conjecture.

First we try to further simplify Constraint 5.3. To this end we note that Constraint 5.3 as originally introduced in [63, Method (B)] is based on the normalized form of the characteristic polynomial (i.e. ignoring the leading sign), so for sake of consistency with Constraint 5.26 we alter the constraint so that it is based on the characteristic polynomial with leading sign included. If we choose a_2, a_1, a_0 such that $f(x) = -x^3 + a_2x^2 + a_1x + a_0$ then the *discriminant* Δ_3 is $-4(a_2)^3a_0 + 4(a_1)^3 + (a_2)^2(a_1)^2 - 27(a_0)^2 - 18a_2a_1a_0$. The characteristic polynomial at positions 1 and -1 changes to $f(-1) = a_2 - a_1 + a_0 + 1$ and $f(1) = a_2 + a_1 + a_0 - 1$. Furthermore, due to the introduction of the leading sign, the geometric shape of the function is flipped upside down. This means, if $\Delta_3 < 0$, the single real root is in $[-1, 1]$ if and only if $f(-1) \geq 0$ and $f(1) \leq 0$. If $\Delta_3 \geq 0$ then we have the additional requirement that the extrema of f have to be in $[-1, 1]$, which can be modelled as follows: $f'(x) \leq 0$ for all $x \in \mathbb{R}$ with $|x| \geq 1$. We will now construct a constraint that reflects exactly these requirements.

The discriminant of f' is $\Delta_2 = 4((a_2)^2 + 3a_1)$ which we simplify to $(a_2)^2 + 3a_1$ since we are only interested in its sign. To illustrate the origin of our next constraint, we will do a case distinction on Δ_2. First, assume $\Delta_2 < 0$. This means f' has no root. Since this implies that f cannot have local extrema, it directly follows that f has exactly one (not necessarily distinct) real root. Due to the properties of the discriminant of $f(x)$, Δ_3, we have $\Delta_3 \leq 0$. The same reasoning can be applied for the converse direction. Since we

already have branched into the case of $\Delta_3 \geq 0$ due to the earlier case distinction we can reason that $\Delta_2 < 0 \Leftrightarrow \Delta_3 = 0$. And since f' has no real root, $f'(x) < 0$ and thus $f'(x) \leq 0$ holds for all $x \in \mathbb{R}$. Next, assume $\Delta_2 = 0$. In this case, we have $f'(x) = -3(x - \frac{a_2}{6})^2$ and $\frac{a_2}{6}$ as the multiple root of f'. Since f' is zero at position $\frac{a_2}{6}$ and less than zero everywhere else we fulfil the requirement. The only case that requires more thought is when $\Delta_2 > 0$. Visualising the geometric shape of f' we can see that we have to demand that $f'(-1) \leq 0$ and $f'(1) \leq 0$ as well as $-3 \leq a_2 \leq 3$.

Due to working with non-negative matrices and Theorem 2.1 we arrive at the following simplification of Constraint 5.3:

Constraint 5.28. *Let $f(x) = -x^3 + \sum_{i=0}^{2} a_i x^i$ be a polynomial built from real numbers satisfying the formula*

$$a_2 + a_1 + a_0 - 1 \leq 0 \wedge (\Delta_3 < 0 \vee (a_2)^2 + 3a_1 \leq 0 \vee (a_2 \leq 3 \wedge 2a_2 + a_1 - 3 \leq 0))$$

Then f has no real root with absolute value greater one.

The arguments are the same: e.g., in the case of $(a_2)^2 + 3a_1 > 0$ and $2a_2 + a_1 - 3 \leq 0$, the added constraint $a_2 \leq 3$ ensures that the right root is in the interval $[0; 1]$. Neurauter et al. [63] reference the Perron-Frobenius Theorem but not in its weak form stated in Theorem 2.1 that allows us to do this simplification.

This takes care of the leading sign and simplifies the constraint a bit. We observe that we now already are a step closer to Constraint 5.26. Moreover, Constraint 5.26 also can be simplified a bit. If (and only if) f is squarefree then $\Delta_3 \neq 0$ and thus $g = -\Delta_3 \neq 0$, and thus we can drop the disjuncts that require g to be equal to zero. So the Sturm constraints look as follows:

Constraint 5.29. *Let $f(x) = -x^3 + \sum_{i=0}^{2} a_i x^i$ be a squarefree polynomial built from real numbers satisfying the formula*

$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 > 0) \vee \tag{5.1}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 > 0) \vee \tag{5.2}$$
$$(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 > 0) \vee \tag{5.3}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 > 0) \vee \tag{5.4}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 > 0) \vee \tag{5.5}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \tag{5.6}$$
$$(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \tag{5.7}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \tag{5.8}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g < 0) \vee \tag{5.9}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \tag{5.10}$$
$$(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \tag{5.11}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 > 0 \wedge g > 0) \vee \tag{5.12}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g > 0) \vee \tag{5.13}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g > 0) \vee \tag{5.14}$$
$$(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g > 0) \vee \tag{5.15}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g > 0) \vee \tag{5.16}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 = 0 \wedge g > 0) \vee \tag{5.17}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 = 0) \vee \tag{5.18}$$

$$(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 = 0) \vee \tag{5.19}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 = 0) \vee \tag{5.20}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 = 0) \vee \tag{5.21}$$
$$(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 = 0) \vee \tag{5.22}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 = 0) \vee \tag{5.23}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 < 0 \wedge g > 0) \vee \tag{5.24}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 < 0 \wedge g > 0) \vee \tag{5.25}$$
$$(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 < 0 \wedge g > 0) \vee \tag{5.26}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 > 0 \wedge -(a_2)^2 - 3a_1 < 0 \wedge g > 0) \vee \tag{5.27}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 > 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 < 0 \wedge g > 0) \vee \tag{5.28}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 < 0) \vee \tag{5.29}$$
$$(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 < 0) \vee \tag{5.30}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 = 0 \wedge -(a_2)^2 - 3a_1 < 0) \vee \tag{5.31}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 = 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 < 0) \vee \tag{5.32}$$
$$(a_2 + a_1 + a_0 - 1 = 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 < 0) \vee \tag{5.33}$$
$$(a_2 + a_1 + a_0 - 1 < 0 \wedge 2a_2 + a_1 - 3 < 0 \wedge -2(a_2)^2 - 6a_1 - a_2a_1 - 9a_0 < 0 \wedge -(a_2)^2 - 3a_1 < 0) \tag{5.34}$$

where $g = 4(a_2)^3a_0 - 4(a_1)^3 - (a_2)^2(a_1)^2 + 27(a_0)^2 + 18a_2a_1a_0 = -\Delta_3$. *Then* f *has no real root with absolute value greater one.*

We can show that compatibility with Constraint 5.3 does not imply squarefreeness of the polynomial, and therefore Constraint 5.3 does not imply Constraint 5.26. For this we note that a polynomial $f(x) = -x^3 + \sum_{i=0}^{2} a_i x^i$ is squarefree if either $f(x) = (x-a)(x-b)(x-c)(-1) = -x^3 + (a+b+c)x^2 - (ab+ac+bc)x + abc \wedge a \neq b \neq c$ for some real numbers a, b, c or if f factors into an irreducible quadratic and a linear factor, i.e. $f(x) = (x^2 - ax - b)(x-c)(-1) = -x^3 + (a+c)x^2 + (b-ac)x - bc \wedge \Delta_2^{(x^2-ax-b)} = a^2 + 4b < 0$ for some real numbers a, b, c. Suppose Constraint 5.3 implied squarefreeness. Let $f(x) = -x^2$. Then f is compatible with Constraint 5.3 but not with the conditions for squarefreeness laid out above, which is a contradiction.

Next we try to prove that the converse direction does actually hold, which means Constraint 5.26 implies Constraint 5.3. For this we note that all disjuncts of Constraint 5.26 imply the conjunct $a_2 + a_1 + a_0 - 1 \leq 0$ since $a_2 + a_1 + a_0 - 1 < 0 \Rightarrow a_2 + a_1 + a_0 - 1 \leq 0$ and $a_2 + a_1 + a_0 - 1 = 0 \Rightarrow a_2 + a_1 + a_0 - 1 \leq 0$. Hence what remains to be proven is that each disjunct of Constraint 5.26 implies (at least) one of the disjuncts of the formula $\Delta_3 < 0 \vee (a_2)^2 + 3a_1 \leq 0 \vee (a_2 \leq 3 \wedge 2a_2 + a_1 - 3 \leq 0)$. In that regard, the disjuncts on lines (5.1) to (5.23) imply $(a_2)^2 + 3a_1 \leq 0$. The disjuncts of lines (5.24) to (5.28) contain the conjunct $-\Delta_3 > 0$ which implies $\Delta_3 < 0$. That the remaining disjuncts ((5.29) to (5.34)) imply one of the above disjuncts (presumably $a_2 \leq 3 \wedge 2a_2 + a_1 - 3 \leq 0$) is more difficult to prove. We presume one has to use the precondition of squarefreeness to prove the implications. The disjuncts on lines (5.29) to (5.34) readily imply $2a_2 + a_1 - 3 \leq 0$ but that they also imply $a_2 \leq 3$ remains a conjecture.

5.2.5 Signed Subresultant PRS

We will see that from matrices of a certain kind we can construct polynomial sequences that can act as a substitute for the formerly defined Sturm Sequences. These sequences

are named Signed Subresultant Sequences, the corresponding matrices are various variants of the Sylvester matrix. They have found plenty of applications in computer algebra programs, yielding better performance in practice than their Sturm polynomial remainder sequence counterparts.

The Signed Subresultant Sequence can be classified as a pseudo-remainder sequence. As mentioned when defining the Extended Sturm Sequences in the section before, pseudo-remainder sequences are the sequences of remainders one obtains when replacing the `euclidean_division(f, g)[1]` operation in Euclid's algorithm with $\frac{f \bmod_{\mathrm{ps}} g}{n}$ where n is chosen such that it divides the coefficients of $f \bmod_{\mathrm{ps}} g$. Trivial choices for n include 1 or $f \bmod_{\mathrm{ps}} g$, and we have seen division by the content in Procedure 5.2.2. The Signed Subresultant polynomial remainder sequence represents one such way to choose appropriate numbers for n.

Before delving deeper into this alernative to Sturm Sequences we introduce a slightly altered definition of sign variations (cf. [11, pp. 332]). Instead of dropping all zeroes before counting, we have to account for some patterns that can be encountered if we do not cull them. Each occurrence of a suitable pattern will further increase the number of sign variations. For now just keep in mind that counting the sign changes present in the sequence after dropping zeroes alone is not sufficient this time.

There are alternatives (e.g. discussed in [18]), but we focus on creating the Signed Subresultant polynomial remainder sequence by linking the coefficients of the remainders to determinants of some specific submatrices.

Definition 5.30. Let $f(x) = \sum_{i=0}^{n} a_i x^i$ and $g(x) = \sum_{i=0}^{m} b_i x^i$ be polynomials and $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_m$ their roots. The number

$$(a_n)^m (b_m)^n \prod_{i=1}^{n} \prod_{j=1}^{m} (\alpha_i - \beta_j)$$

is the *resultant* of f and g.

Sylvester [68, 67] details how to construct a matrix whose determinant corresponds to the resultant of the polynomials:

Definition 5.31. Let $f(x) = \sum_{i=0}^{n} a_i x^i$ and $g(x) = \sum_{i=0}^{m} b_i x^i$ be polynomials of degree n and m. The $(m+n)\times(m+n)$ matrix whose entries correspond to the function

$$\mathbf{A}_{ij} = \begin{cases} a_{n+i-j} & \text{if } i \leq m \text{ and } i \leq j \leq n+i, \\ b_{i-j} & \text{if } i > m \text{ and } i-m \leq j \leq i, \\ 0 & \text{otherwise.} \end{cases}$$

is called the *Sylvester Matrix* of f and g.

Theorem 5.32. *Let f and g be polynomials and $\mathbf{A}$ their Sylvester Matrix. The determinant of $\mathbf{A}$ corresponds to the resultant of f and g.*

Definition 5.33 ([69])**.** Let $f(x) = \sum_{i=0}^{n} a_i x^i$ and $g(x) = \sum_{i=0}^{m} b_i x^i$ be polynomials of degree n and m, and without loss of generality let $n \geq m$. The $2n \times 2n$ matrix whose entries correspond to the function

$$\mathbf{A}_{ij} = \begin{cases} a_{n-j+1} & \text{if } \lceil \frac{i}{2} \rceil \leq j \leq n + \lceil \frac{i}{2} \rceil \text{ and } i \text{ is odd,} \\ b_{n-j+1} & \text{if } n - m + \lceil \frac{i}{2} \rceil \leq j \leq n + \lceil \frac{i}{2} \rceil \text{ and } i \text{ is even,} \\ 0 & \text{otherwise.} \end{cases}$$

we call the *Sylvester Matrix of 1853* of f and g.

Definition 5.34 (cf. [4])**.** Let f and g be polynomials and $\mathbf{A}$ their Sylvester Matrix of 1853. The modified resultant of f and g is defined as the determinant of $\mathbf{A}$.

Remark. The modified resultant is identical to the resultant up to possibly the sign and an integer factor.

We are now able to construct the resultant of two polynomials. What are subresultants, then? Simply put, subresultants are determinants of submatrices of a matrix whose determinant is a resultant, e.g. the Sylvester Matrix.

Theorem 5.35 ([28])**.** *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ and $g(x) = \sum_{i=0}^{m} b_i x^i$ be polynomials of degree n and m, without loss of generality let $n \geq m$. For all natural numbers $k < m$ let $\mathbf{A_k}$ be the $(m + n - 2k) \times (m + n - k)$ matrix whose entries correspond to the function*

$$(\mathbf{A_k})_{ij} = \begin{cases} a_{n+i-j} & \text{if } i \leq m - k \text{ and } i \leq j \leq n + i, \\ b_{i-j} & \text{if } i > m - k \text{ and } i + k - m \leq j \leq i + k, \\ 0 & \text{otherwise.} \end{cases}$$

and for all $l \leq m + n$ let $\mathbf{A_{kl}}$ be the $m \times m$ matrix whose entries correspond to

$$(\mathbf{A_{kl}})_{ij} = \begin{cases} (\mathbf{A_k})_{ij} & \text{if } j \leq m - 1, \\ (\mathbf{A_k})_{i,k+j} & \text{otherwise.} \end{cases}$$

Then for natural numbers $k < m$, the kth element of the Subresultant Sequence of f and g is defined as:

$$\sum_{i=0}^{n-m} (x^{n-m-i} |\mathbf{A_{ki}}|)$$

There are several other matrices which can serve as basis for the calculation of the Subresultant Sequence. Among the various alternatives found in the literature (cf. eg. [28]), we want to highlight the one introduced by Sylvester himself, the Bézoutic Square.

Definition 5.36 ([69], also cf. [19])**.** Let $f(x) = \sum_{i=0}^{n} a_i x^i$ and $g(x) = \sum_{i=0}^{m} b_i x^i$ be univariate polynomials of degree n and m, and without loss of generality let $n \geq m$. Let $c_0, \ldots, c_{n^2-1}$ be the coefficients of the bivariate polynomial

$$\sum_{i=0}^{n^2-1} c_i x^{(i \bmod n)} y^{\lfloor \frac{i}{n} \rfloor} := \frac{f(x)g(y) - f(y)g(x)}{x - y}$$

The $n \times n$ matrix whose entries correspond to the function

$$\mathbf{A}_{ij} = c_{i-1+(j-1)n}$$

is then called the *Bézoutic Square* of f and g.

Theorem 5.37 ([28])**.** *Let f and g be polynomials of degree n and m, without loss of generality let $n \geq m$, and let $\mathbf{A}$ be their Bézoutic Square. Let $1 \leq k' \leq m$ be a natural number and let $k = k' + n - m$. Further, let $\mathbf{B_{k,0}}, \mathbf{B_{k,1}}, \ldots, \mathbf{B_{k,n-k}}$ be the sequence of $k \times k$ matrices of the form:*

$$(\mathbf{B_{k,t}})_{ij} = \begin{cases} \mathbf{A}_{n-k+i,n-k-t+1} & \text{if } j = 1 \\ \mathbf{A}_{n-k+i,n-k+j} & \text{otherwise} \end{cases}$$

Then the $(n-k)$th element of the subresultant polynomial sequence can be expressed as:

$$\frac{\mathbf{B_{k,0}} x^{n-k} + \mathbf{B_{k,1}} x^{n-k-1} + \ldots + \mathbf{B_{k,n-k}}}{(-1)^{(k^2-k)/2} \cdot \mathrm{lc}(f)^{n-m}}$$

Remark. One can construct a matrix that transforms a Sylvester Matrix to a Bézoutic Square. For details on the process refer to [21].

We will now state a definition for the number of sign variations in a sequence that is adapted to fit the purpose of allowing us to use Signed Subresultant Sequences to count the number of roots a polynomial has in a given interval.

Definition 5.38. Let $s = (a_i)_{i<n}$ be a finite real numbered sequence and let $(b_i)_{i<m}$ be the sequence consisting of the non-zero elements of s. Further, let k be the number of indices i for which either $a_i > 0 \wedge a_{i+1} = 0 \wedge a_{i+2} = 0 \wedge a_{i+3} > 0$ or $a_i < 0 \wedge a_{i+1} = 0 \wedge a_{i+2} = 0 \wedge a_{i+3} < 0$. The *number of sign variations* in s is defined as

$$\max(0, \frac{b_0 b_1}{-|b_0 b_1|}) + \max(0, \frac{b_1 b_2}{-|b_1 b_2|}) + \ldots + \max(0, \frac{b_{m-2} b_{m-1}}{-|b_{m-2} b_{m-1}|}) + 2k$$

The extension of Definition 5.38 to sequences of polynomials is analogous to Definition 5.6 and Definition 5.12. For the rest of the section we exclusively use the new definition. To spare the reader from ambiguities we will refer to this definition of sign variations as *sign variations (5.38)* throughout the rest of the thesis.

With the help of this new definition of sign variations, the sequence of subresultants can be applied to determine the number of roots in an interval analogous to the (Extended) Sturm Sequence.

Theorem 5.39. *Let f be a non-constant polynomial and let m and n be two distinct real numbers. Without loss of generality let $m < n$. Further, let $f(m) \cdot f(n) \neq 0$. Then the number of (not necessarily distinct) roots of f in the interval $(m, n]$ is the the same as the number of sign variations (5.38) of the Signed Subresultant Sequence of f at position n subtracted from the number of sign variations (5.38) of the Signed Subresultant Sequence of f at position m.*

5.2.6 Constraints Based on Subresultant Sequences

Constraint 5.40 (Theorem 5.39)**.** *Let f be a polynomial over the reals and let $(f_i)_{i<m}$ be the Signed Subresultant Sequence of the polynomial. Let $f(1) \neq 0$, and let the number of sign variations (5.38) of $(f_i)_{i<m}$ at position 1 be the same as the number of sign variations (5.38) of $(f_i)_{i<m}$ at position ∞. Then f has no real root with absolute value greater one.*

6 Vincent's Theorem

Even though consulting Sturm's Theorem may be seen as the standard procedure when attempting to manually isolate real roots, it is not the only theorem of its kind. Quite unaccredited is a theorem created independently by Vincent, a fellow contemporary of Sturm.

This theorem is shrouded in quite some confusion and misunderstandings. First, Vincent published his theorem twice, in 1834 ([75]) and 1836 ([76]). Both publications had a small cryptic addendum instructing to "follow the method of Lagrange", which was *vital to the theorems correctness* but many later authors (Alesina, Galuzzi, Uspensky) failed to recite. Since 2000, the literature knows not of one but two versions of his theorem, the more modern version being Alesina's and Galuzzi's *Bisection* Theorem [5]. To further complicate the case, there is a naming issue for an underlying theorem. Vincent's Theorem is directly built on a theorem due to Budan. Similar to how Sturm modified Fourier's Theorem which became the Sturm Sequence, Vincent thought up an isolation method that employs continued fractions by building on Budan's Theorem. Which is where the confusion comes in: while the two theorems are essentially equivalent, the way they are stated is still important when we want to understand how Sturm and Vincent developed their theorems. Technically speaking, Budan's Theorem was published earlier. According to Arago, Fourier was determined to prove that he had taught his theorem before Budan (cf. [17, p. 164]) . Eventually Fourier prevailed in the priority dispute. Just as the Sturm method was much more covered in literature the years to follow while Vincent's method was completely forgotten, the same happened to the theorems these methods were built upon. As an example, Pierre-Louis-Marie Bourdon, father-in-law of Vincent, published Fourier's version of the theorem in his famous algebra textbook [16, p. 515], although he included a reference to the works of both Budan and Fourier. And whilst almost all literature exclusively stated Fourier's Theorem (until recent efforts to re-establish both Vincent's and Budan's Theorem in literature, most notably by Akritas), it was often simply called "Fourier-Budan Theorem" or sometimes even "Budan's Theorem" [29, 77]. Hence the clear distinction between the two theorems was obscured. There has also been a misattribution with Vincent's Theorem itself. When Uspensky restated Vincent's Theorem with some minor (detrimental) modifications in his book *Theory of Equations* [73] – not to be confused with a book of the same title by Turnbull published just a year earlier –, Collins and Akritas constructed the first algorithm based on the theories presented by Uspensky. In their publication, Collins and Akritas mistakenly attributed the theorem to Uspensky. They later published the aptly named article "There is no Uspensky's method" [2] to clear up any misunderstandings. In light of these historical facts we shall now present Budan's Theorem and a reformulation of Vincent's Theorem.

6.1 Budan's Theorem

Recall Definition 4.4. If we can determine an upper bound for the number of roots in an interval starting at an arbitrary position and ending in $-\infty$ or $+\infty$ we could do so for two different points and subtract one of the results from the other. The absolute value of this number is a bound for the interval that is spanned by these two points. This is known as the method of Budan. Budan also proved that the parity of the bound and the actual number of roots still correlate.

Vincent stated in his work that he was familiar with both the theorems of Budan and Fourier. We will compare the two theorems shortly. Budan's Theorem can be stated as follows:

Theorem 6.1 (Budan, [23])**.** *Let f be a non-constant polynomial in x and let $p, q \in \mathbb{R}$ such that $p < q$. Then the number of sign variations of $f(x+p)$ is greater or equal to the number of sign variations of $f(x+q)$. Furthermore, the number of roots of f in the interval $(p, q]$ cannot exceed this difference. If there is a discrepancy between this bound and the actual number of roots, then the difference has to be even.*

Example 6.2. Consider the polynomial $f = 3x^2 - 5x + 1$. Then we have

$$f(x+1) = 3x^2 + x - 1$$

$$f(x+3) = 3x^2 + 13x + 13$$

and thus there is exactly one root in the interval $(1, 3)$.

We can observe that Budan's Theorem and Fourier's Theorem look strikingly similar. This is no coincidence since the theorems can be shown to be equivalent.

6.1.1 Equivalence of the Theorems by Fourier and Budan

For any constant c we can represent the polynomial $f(x+c)$ as a Taylor series $\sum_{i=0}^{deg(f)} \frac{f^{(i)}(c)}{i!} x^i$, where $f^{(i)}$ denotes the ith derivative of f. Note that up to a strictly positive factor these coefficients match the elements of the Fourier series. Thus the sign sequences are identical.

6.2 From Budan's to Vincent's Theorem

In the following we outline the steps that led from Budan's Theorem to Vincent's. The main idea is as follows: Vincent observed that if the number of sign variations is either zero or one, this corresponds precisely to the number of roots in the interval in question. Based on this fact Vincent proposed the following sequence of transformations which guarantees the number of sign variations to be either zero or one:

Theorem 6.3 (Vincent, [75, 76])**.** *Let f be a non-constant polynomial of degree n with rational coefficients and without multiple roots. Further, let $c_0, c_1, \ldots, c_{h-1}$ be constants greater one. Then when one makes successive transformations of the form*

$$x \leftarrow c_0 + x^{-1}, x \leftarrow c_1 + x^{-1}, \ldots, x \leftarrow c_{h-1} + x^{-1}$$

for sufficiently large h, the resulting expression gives rise to a polynomial that has either zero or one sign variation. This sequence of transformations can be written as a single transformation of the form

$$x \leftarrow [c_0; c_1, \ldots, c_{h-1}, x]$$

which is equivalent to the Möbius transformation

$$x \leftarrow \frac{ax + b}{cx + d}$$

for some non-negative integers a, b, c, and d[1]. By eliminating the divisor in the resulting expression we obtain a polynomial[2]

$$g(x) = (cx + d)^n f\left(\frac{ax + b}{cx + d}\right)$$

whose coefficients have either zero or one sign change. This determines the number of roots of f in the interval $(\frac{a}{c}, \frac{b}{d})$.

6.3 A Root Counting Procedure

Based on his theorem, Vincent invented a method to count the number of non-negative real roots as an example use case. In this chapter we are going to study this method and see how we can apply it to get a sufficient and necessary condition that ensures $\rho(\mathbf{A}) \leq 1$. There are multiple variations to this counting method known to modern literature. For an in-depth comparison we want to point to Akritas [3].

Let f be a non-constant polynomial of degree n with rational coefficients and without multiple roots. Initially we are concerned with counting the strictly positive roots only. Using Budan's Theorem, we check whether we can conclude that there is exactly zero or one root in the interval $(0, \infty)$. If not, we try to divide the search space by treating roots smaller, equal and greater than one separately. As a first step we evaluate f at one to test whether one is a root of f. Then we count the remaining roots. First we observe that roots smaller than one can be written as $(c + 1)^{-1}$, roots greater than one as $c + 1$ for some $c > 0$. To investigate the intervals $(0, 1)$ and $(1, \infty)$, we shift f by substituting x with $(x + 1)^{-1}$ and $x + 1$ respectively. To turn the former expression into a polynomial we multiply it with $(x + 1)^n$. Note that this operation is root-preserving. We can now use Budan's Theorem individually on these newly obtained polynomials and

[1] Cf. [65]

[2] Note that we multiply by the factor $(cx + d)^n$. As Biagioli points out in his dissertation [15], this elimination of the denominator is what Vincent meant by the "method of Lagrange".

the corresponding intervals to get an insight on the number of roots they contain. If Budan yields no exact result we once again split the search space, i.e. we substitute the indeterminate and try again, repeating these steps until the algorithm terminates. Using his theorem, Vincent proved that this process is guaranteed to terminate.

Up to this point, we omitted strictly negative roots. Fortunately, counting strictly negative roots is in no way harder than counting strictly positive roots, in fact we can use the very same algorithm. The only required preprocessing step is substituting x with $-x$ in the original polynomial. So to count all the real roots of f, we have to evaluate $f(0)$, run Vincent's root counting procedure on $f(-x)$ and $f(x)$, and finally add up the results. Figure 6.1 depicts Vincent's root counting method in an algorithmic fashion.

```
procedure count_roots(f):
  i ← sign_changes(f)
  if i = 0 or i = 1 then
    return i
  f01(x) ← (x + 1)^deg(f) f(1/(x+1))
  f1∞(x) ← f(x + 1)
  if f(1) = 0 then
    return count_roots(f01) + count_roots(f1∞) + 1
  else
    return count_roots(f01) + count_roots(f1∞)
```

Figure 6.1: Vincent's proposed root counting method (cf. [15]).

Now that we have the theory set, we can investigate if we can employ this new algorithm to check whether $\rho(\mathbf{A}) \leq 1$. In other words, we need to determine whether $\chi_{\mathbf{A}}$ contains no roots in the interval $(1, \infty)$. As we can see, the algorithm lends itself well to this problem: all we need to do is - similarly to what we did to count strictly negative roots - transform $\chi_{\mathbf{A}}$ by the variable replacement $x \leftarrow x + 1$ before calling the actual counting procedure. Since the algorithm is correct and computes the *exact* number of roots the constraints will be both sufficient and necessary to our criterion of $\rho(\mathbf{A}) \leq 1$, provided the characteristic polynomial of $\mathbf{A}$ is squarefree. Alas, there is an infinite number of disjuncts. But to obtain a sufficient criterion it is enough to select any number of disjuncts.

6.3.1 Constraints Based on Vincent's Theorem

Constraint 6.4 (Figure 6.1)**.** *Let $f(x) = \sum_{i=0}^{n} a_i x^i$ be a non-constant, squarefree polynomial built from real numbers satisfying at least one of the disjuncts of the*

non-terminating formula

$$
\begin{aligned}
&(\exists\!\!\!\exists_{i,j>0} f_{i,j}).\, \mathrm{sc}(f(x+1)) = 0 \vee \\
&(f_{1,1} = f(x+1) \wedge \mathrm{sc}(f_{1,1}) \neq 1 \wedge f_{1,1}(1) \neq 0 \wedge (\\
&\quad \mathrm{sc}(f_{2,1}) + \mathrm{sc}(f_{2,2}) = 0 \vee \\
&\quad (f_{2,1} = f_{1,1}(x+1) \wedge f_{2,2} = (x+1)^{\deg(f_{1,1})} \cdot f_{1,1}(\frac{1}{x+1}) \wedge \\
&\quad \mathrm{sc}(f_{2,1}) \neq 1 \wedge \mathrm{sc}(f_{2,2}) \neq 1 \wedge f_{2,1}(1) \neq 0 \wedge f_{2,2}(1) \neq 0 \wedge (\\
&\qquad \ldots \\
&))))
\end{aligned}
$$

where sc *is a function that counts the sign changes of the coefficients of a polynomial. Then f has no real root of absolute value greater one.*

6.4 Thoughts on Squarefreeness

As stated in the previous sections, Vincent's Theorem establishes squarefreeness of the input polynomial as an imperative precondition. Thus also the root counting algorithm devised from this theorem necessarily relies on the input polynomial being squarefree. As we will see in this section this turns out to be quite problematic.

The following lemma provides a sufficient and necessary criterion for the squarefreeness of univariate polynomials over a field of characteristic 0.

Lemma 6.5 (cf. e.g. [40, p. 339])**.** *Let f be a non-constant polynomial, with coefficients in a field of characteristic 0, and let f' be its formal derivative. Then f is squarefree if and only if $gcd(f, f') = 1$.*

By evaluating the primitive Euclidean algorithm on a symbolic polynomial we arrive at the constraints of Corollary 6.6.

Corollary 6.6. *Let $f = \sum_{i=0}^{3} a_i x^i$ be a cubic normalized polynomial with coefficients in a field of characteristic 0. Then f is squarefree if and only if either $4 \cdot (3{a_1}^2 - {a_2}^2 a_1)/(4{a_2}^3 - 15a_2a_1 + 27a_0) \cdot ((9{a_1}^2 - 3{a_2}^2 a_1)/(4{a_2}^3 - 15a_2a_1 + 27a_0) - a_2) + a_1 = 0$ as well as $(4{a_2}^3 - 15a_2a_1 + 27a_0)/27 = 1$ and $6{a_1}^2 - 2{a_2}^2 a_1 = 0$, or $4 \cdot (3{a_1}^2 - {a_2}^2 a_1)/(4{a_2}^3 - 15a_2a_1 + 27a_0) \cdot ((9{a_1}^2 - 3{a_2}^2 a_1)/(4{a_2}^3 - 15a_2a_1 + 27a_0) - a_2) + a_1 = 1$.*

Proof. We synthesize the constraints by applying the steps of the primitive Euclidean algorithm to f and f'. The first iteration of the Euclidean algorithm calculates the remainder of $\frac{f}{f'}$. In general, the algorithm terminates when the remainder becomes zero, in which case the divisor used in the last iteration is the result and thus the highest common factor of the input polynomials. This means that for each iteration of the algorithm we get a new disjunct for our constraints where we demand that the remainder equals zero and the divisor equals one. We now observe the fact that each

run of the algorithm which has the desired outcome of $gcd(f, f') = 1$ consists of at least two iterations. This is due to the fact that the divisor used for the first iteration, f', is a non-constant polynomial and thus cannot equate to 1. Hence we ignore the first iteration when building the constraints and immediately skip to the second iteration. There our divisor is $(4a_2^3 - 15a_2a_1 + 27a_0)/27 + (6a_1^2 - 2a_2^2a_1)/27x$. We can simplify the constraints by observing that for the divisor to become a constant the term $6a_1^2 - 2a_2^2a_1$ has to equate zero. The third iteration leaves us with a remainder that equals zero irrespective of the value of the symbolic constants. Thus for the second disjunct we only have to constraint the corresponding divisor (i. e. the remainder of the last iteration). □

Utilizing the constraints of Corollary 6.6, and constraints for polynomials of higher degree synthesized in analogous manner, one could make use of the Vincent root counting method described in the previous section. In the context of the overarching goal of this thesis, demanding that the characteristic polynomial is squarefree is not sensible in practice. Recall one of the main theorems of [63], introduced in Chapter 2.

Theorem 6.7 ([63])**.** *Let $A \in \mathbb{R}_0^{n\times n}$ be the component-wise maximum matrix of all matrices of a matrix interpretation compatible to a term rewrite system $\mathcal{R}$. If $\rho(\mathbf{A}) \leq 1$, then $dc_{\mathcal{R}}(k) \in \mathcal{O}(k^{d+1})$ where $d := max_\lambda(0, m_\lambda) - 1$ and λ are the eigenvalues with absolute value exactly one.*

Theorem 6.7 makes clear why there is no sense in demanding the characteristic polynomial to be squarefree. For linear bounds we already have the much simpler Constraint 5.24. Additionally, we can see that constraints which limit the eigenvalues to absolute value strictly smaller one imply $\mathcal{O}(1)$ and thus are not of much interest to us either.

The following sections discuss approaches to alleviate the problems that come with the restriction to squarefree polynomials.

6.4.1 Wang's Theorem

Completely unaware of the existence of Vincent's Theorem, Wang formulated a theorem (not to be confused with the Grunwald–Wang Theorem) that subsumes Vincent's Theorem [20]. This more general theorem can be used to isolate (or in our case rather count) the roots of polynomials that do not necessarily have to be squarefree.

6.4.2 Squarefree Decomposition

We still can use Vincent's original theorem by introducing an additional preprocessing step. The idea is to decompose the original polynomial, which may have multiple roots, into factors that have simple roots only. In fact, as can be seen in Theorem 6.7, any polynomial that is of potential interest for us *will* have multiple roots. If we can find factors that are squarefree and can be composed into the original polynomial, we can use Vincent's Theorem on each of these factors individually. Luckily it turns out that given a polynomial, it is always possible to find such factors.

Definition 6.8. Let f be a non-zero polynomial with coefficients in a field F. A *squarefree factorization* is a factorization

$$f = \prod_{i=1}^{k} a_k^k$$

where the polynomials $a_1, a_2, \ldots, a_k$ are all squarefree, and further both $a_k \neq 1$ and $(\forall i, j \leq k, i \neq j)[\gcd(a_i, a_j) = 1]$.

Such a squarefree factorization exists for every non-zero polynomial and it is unique up to multiplication by non-zero constants.

For the implementation of the preprocessing step we can choose among a wide variety of squarefree factorization algorithms. We could use for example the method of Yun [79] to decompose the polynomial into squarefree parts. Then the roots of a factor a_i isolated with the help of Vincent's Theorem are the roots of f with multiplicity i.

The main question that remains is how to know *à priori* which polynomial shall be factorized. Of course we could search for matrices compatible to the TRS, decompose the characteristic polynomials into their squarefree factors and then check for conformance to one of the disjuncts of Constraint 6.4, following a *generate and discard* pattern. Performance-wise, however, this procedure is not sufficient. The likelihood that a compatible matrix has spectral radius smaller equal one out of pure chance is incredibly low. We thus need an approach that allows us to guide the search for compatible matrices towards matrices that comply with our requirements on the spectral radius. Such as augmenting the various constraints we have already seen in this document with an additional conjunct derived from procedures for symbolic factorization, where required. Jordan et al. [49] mention such a procedure originally used in a proof by Kronecker. Another attempt would be to tackle this problem bottom-up. Based on the assumption that compatibility with the TRS is sometimes easier to achieve than compliance with the constraints proposed in this thesis, one could try to find squarefree polynomials (using Lemma 6.6) and multiplying them together. Then one would test for compatibility, discarding combinations that were not compatible with the TRS. But there is still room for improvement. One could now guide the search for polynomials composed of squarefree factors compatible to disjuncts of Constraint 6.4 toward polynomials that would be compatible to the TRS in question. For this we fix the number of factors we want to multiply in order to obtain the final polynomial (note that this inherently decides on the maximum multiplicity and thus puts a limit on the kind of bound we can obtain, which is on the one side bad when it comes to maximizing the number of TRSs we can bound but on the other side can be exploited when specifically searching for lower bounds on the derivational complexity) and encode the multiplication of the factors directly in the constraints. As we alluded to at the beginning of the section, using Vincent's Theorem on quadratic or linear factors will not be better than using e.g. Budan's Theorem since in this case, ignoring squarefreeness, the constraints are equal. But we can use Vincent's Theorem when we decide to tackle matrices of higher dimensions, by decomposing their characteristic polynomials into some factors of degree 3 along the linear and irreducible quadratic factors. For the cubic factors Vincent's

Theorem used in conjunction with Lemma 6.6 could prove to be more efficient than e.g. Constraint 5.3. The idea of factorization is also discussed in [63, Method (C)].

Remark. When deriving constraints from a squarefree factorization algorithm we can limit the constraints to factorizations that admit no more than a certain multiplicity for the root at position one. This allows us to explicitly search for bounds that are tighter than the loosest bound that is normally inferred given the degree of the characteristic polynomial.

7 Experimental Results

We chose to implement Constraint 5.23 and Constraint 6.4 in the open source complexity analyser T$_C$T[1] in order to examine their performance and compare them to earlier techniques found in [63] and [60]. Preliminary tests showed that the squarefreeness prerequisite was severely at odds with the anticipated efficiency of the constraints. We further found that the theorems would not scale well to polynomials of degree four or higher, as the constraints would grow quite big in size. We thus decided to alternatively evaluate the performance of Theorem 6.1 – which, in theory, should be able to scale to polynomials of higher degree rather easily – and empirically test the difference that the simplification of Constraint 5.3 to Constraint 5.28 would make.

As a basis for our evaluation process serves the testbed provided by the Termination Problems Database (TPDB for short).[2] As hinted at by its name, the TPDB provides a collection of term rewrite systems tailored to the evaluation of automatic termination analysers. We are not aware of a database specifically made with complexity proofs in mind, so the TPDB was the best-suited alternative. In our experiments we used version 10.5 of the database. We have conducted the experiments with a time limit of 60 seconds per rewrite system. When the derivational complexity cannot be successfully bounded in this time frame, we mark this as a 'timeout'. It is possible that T$_C$T aborts the proof attempt earlier (i.e. output of T$_C$T is either 'maybe' or 'error'), in which case we flag it as 'failure'. While a 'timeout' has the potential to be a success with a higher execution time and may even be on the verge of the 60 second limit, a 'failure' clearly indicates the rewrite system cannot be bounded with the chosen configuration.

To aid in the evaluation process, we have written a python program that generates constraints for Theorem 6.1 and polynomials of arbitrary degree in Haskell syntax directly compatible to the T$_C$T source code, as well as a C++ program that instantiates Constraint 5.23 for symbolic characteristic polynomials of square matrices of arbitrary size.

For a complexity bound of $\mathcal{O}(k^2)$, both Method (A) [63] and Theorem 6.1 become Constraint 5.24. Further, Constraint 5.23 and Constraint 6.4 differ only in the preconditions. Let us see how well we fare with this constraint when we try to bound the derivational complexity the term rewrite systems of the TPDB 10.5. Out of the 1758 rewrite systems we couldn't prove a linear bound for, on 107 we could impose a quadratic bound. The full results are listed in Table 7.1.

Let us now see how the individual constraints perform on the rest of the rewrite systems: EDA [60] is the most successful and the fastest on average (in respect to successfully bounded TRSs). We hoped for a speed-up by using the EDA criterion directly on the

[1] http://cl-informatik.uibk.ac.at/software/tct/
[2] http://termination-portal.org/wiki/TPDB

Table 7.1: Number of rewrite systems bounded with the shared constraint

		Constraint 5.24
$\mathcal{O}(k)$	51	-
$\mathcal{O}(k^2)$	-	107
Failure	1463	517
Timeout	295	1134

maximum matrix (which would be less precise), but this hurt the performance notably. As expected Theorem 6.1 is faster than the more complex Constraint 5.3. Surprisingly though, Constraint 5.3 is faster on average than Constraint 5.28. This may be due to the additional rewrite systems bounded by Constraint 5.28 probably being harder to analyse than the ones Constraint 5.3 could also bound in complexity. EDA and Theorem 6.1 have longer average run time for rejected examples (i.e. T$_C$T gave up before timing out), but then again they have rejected more rewrite systems and the more complex ones of them may have jacked the numbers up. While not showing as much efficacy as EDA, we found the number of TRSs bounded by Theorem 6.1 to be surprisingly high relative to constraints 5.3 and 5.28. Due to the simplicity of Theorem 6.1 and its modest growth rate for input polynomials of high degree, we expected it to be suitable for proving quartic or higher complexity bounds for the more intricate rewrite systems of the TPDB. But in our tests we found that using Theorem 6.1 with polynomials of degree four or higher was not practicable with a timeout of 60 seconds. By using these constraints which are not significantly bigger in size than the characteristic polynomials they are based on, we tried to estimate how much of a hindrance the size growth innate to symbolic characteristic polynomials really is. It turns out that regardless of the timeout anything beyond characteristic polynomials of degree five is not usable by any means. This entails that we can label polynomials of higher degrees as unusable with Constraint 5.23 and Constraint 6.4 too, since these are considerably more complex in their structure and would thus most likely incur an even longer runtime.

Table 7.2: Number of rewrite systems with proven complexity bounds

	Constraint 5.3	Constraint 5.28	Theorem 6.1	EDA [60]	EDA[a] [60]
$\mathcal{O}(k^3)$	4	6	12	28	18
Failure	94	96	132	367	248
Timeout	1553	1549	1507	1256	1385

[a] (maximum matrix)

In summary, we have found the biggest bottleneck to be the high growth rate of symbolic characteristic polynomials when they are of higher degree, as even plain constraints, e.g. based on the Furier or Budan Theorem, are blown out of proportion in terms of complexity and size and thus are severely restricted in their performance. Further,

Table 7.3: Average runtimes

	Constraint 5.3	Constraint 5.28	Theorem 6.1	EDA [60]	EDA[a] [60]
$\mathcal{O}(k^3)$	23.76s	31.30s	14.66s	13.79s	19.95s
Failure	4.29s	4.34s	9.37s	11.91s	12.63s
Timeout	60.86s	60.85s	60.90s	61.13s	61.02s

[a] (maximum matrix)

squarefreeness poses a significant obstacle for both the Vincent and the Sturm approach. Thus we think the approach of [60, EDA] is the way forward.

8 Conclusion

Let us shortly recap the research question we tried to answer in our thesis:

> "What logical formulas can we formulate for the characteristic polynomial of a non-negative real matrix such that its spectral radius is smaller equal one?"

With our goal in mind, we thus set out to scour the vastness of the mathematical literature for procedures, lemmata and theorems that bound the real roots of a polynomial. Even while excluding the numerical approaches from the get-go, it turned out that the yield of theorems applicable to our research question was encouragingly high. This subsequently allowed us to devise a plethora of logical formulae that met our requirements. For a better overview, all the formulae discussed throughout the foregoing chapters are listed at the end of the corresponding section. It has to be stated, though, that a large share of the formulae are modelling criteria that are not sharp. As a consequence, the theoretical bounds implied by the logical formulae are not necessarily tight. Further discouraging is the weak performance of the new constraints in the experimental evaluation process. Even the theoretical power of those constraints that are sharp does not really translate into adequate performance when it comes to practical evaluation. A disadvantage innate to all constraints based on properties of the characteristic polynomial is the enormous growth in size that characteristic polynomials of higher degree exhibit. This effectively limits the size of the matrix these constraints can support and puts the methods at a direct disadvantage to methods that scale without much effort - such as [60, EDA]. Earlier work evaluating the performance of automata based approaches[1] has proven that the ability to work with matrices of higher order can provide the means to bound the derivational complexity of term rewrite systems where this has not been possible before. Despite the horrendous growth in size inherent to formulae incorporating characteristic polynomials of high degree, we bravely took one of the simplest constraints of this thesis – based on the Budan-Fourier Theorem – and wrote a program that prints the logical formula for polynomials of arbitrary degree. The program, as well as all other software not supplied with the printed document, is available upon request. Although purposely choosing one of the least complex formulas, the performance decreased rapidly with increasing matrix order. With characteristic polynomials of higher degree, the search for matrices compatible to both the constraint and the term rewrite system quickly ended up being capped by the 60 second timeout. Bear in mind that the logical formula based on the Budan-Fourier-Theorem is not much more complex than the characteristic polynomial itself and probably as simple as it gets when it comes to constraints based on the characteristic polynomial. Factorization (cf. Section 6.4.2 and [63, Method (C)]) as

[1] Cf. http://colo6-c703.uibk.ac.at/ttt2/hz/polymatrix/nontriangular/index.php and [63, Lemma 12]

well as constraints not relying on properties of the characteristic polynomial, such as those of Chapter 3, provide alternatives which avoid the issues that come with characteristic polynomials of high degree and may provide a path for future research. We also think that the ideas expressed in [78] might yield new results.

Bibliography

[1] A. G. Akritas. Reflections on a Pair of Theorems by Budan and Fourier. *Mathematics Magazine*, 55(5):292–298, 1982.

[2] A. G. Akritas. There is No 'Uspensky's Method'. In *Proceedings of the Fifth ACM Symposium on Symbolic and Algebraic Computation*, pages 88–90, New York, 1986.

[3] A. G. Akritas. Vincent's Theorem of 1836: Overview and Future Research. *Journal of Mathematical Sciences*, 168:309–325, 2010.

[4] A. G. Akritas. Anna Johnson and Her Seminal Theorem of 1917. Компьютерные инструменты в образовании, 2, 2016.

[5] A. C. Alesina and M. Galuzzi. Vincent's Theorem from a Modern Point of View. *Rendiconti del Circolo Matematico di Palermo*, Serie II Suppl. 64:179–191, 2000.

[6] A. Aziz and Q. G. Mohammad. On the Zeros of a Certain Class of Polynomials and Related Analytic Functions. *Journal of Mathematical Analysis and Applications*, 75(2):495–502, 1980.

[7] A. Aziz and Q. G. Mohammad. Zero-Free Regions for Polynomials and Some Generalizations of Eneström-Kakeya Theorem. *Canadian Mathematical Bulletin*, 27(3):265–272, 1984.

[8] A. Aziz and B. Zargar. Bounds for the Zeros of a Polynomial with Restricted Coefficients. *Applied Mathematics*, 3(1):30–33, 2012.

[9] A. Aziz and B. A. Zargar. Some Extensions of Eneström-Kakeya Theorem. *Glasnik Matematički. Serija III*, 31(2):239–244, 1996.

[10] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[11] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics).* Springer-Verlag, Berlin, Heidelberg, 2006.

[12] P. Batra, M. Mignotte, and D. Ştefănescu. Improvements of Lagrange's Bound for Polynomial Roots. *Journal of Symbolic Computation*, 82:19–25, 2017.

[13] I. Bendixson. Sur les Racines d'une Équation Fondamentale. *Acta Mathematica*, 25:359–365, 1902 (init. pub. 1900).

[14] H. Benis-Sinaceur. Deux Moments dans l'Histoire du Théorème d'Algèbre de Ch. F. Sturm. *Revue d'Histoire des Sciences*, 41(2):99–132, 1988.

[15] E. J. Biagioli. *Methods for Bounding and Isolating the Real Roots of Univariate Polynomials*. D.Sc. thesis, IMPA, March 2016.

[16] M. Bourdon. *Élémens d'Algèbre*. Alexandre de Mat, a la Librairie Classique et Mathematique, Rue de la Batterie, n. 24, 1836.

[17] C. Brezinski. *History of Continued Fractions and Padé Approximants*, volume 12. Springer Science & Business Media, 2012.

[18] W. S. Brown. The Subresultant PRS Algorithm. *ACM Trans. Math. Softw.*, 4(3):237–249, 1978.

[19] A. Cayley. Note sur la Méthode d'Élimination de Bezout. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 53:366–367, 1857.

[20] J. Chen. A new Algorithm for the Isolation of Real Roots of Polynomial Equations. In *Proc. 2nd International Conference on Computers and Applications*, pages 714–719. IEEE Computer Soc. press, 1987.

[21] E.-W. Chionh, R. Goldman, and M. Zhang. Transformations and Transitions from the Sylvester to the Bezout Resultant. Technical report, 1999.

[22] G. E. Collins. Krandick's Proof of Lagrange's Real Root Bound Claim. *Journal of Symbolic Computation*, 70:106–111, 2015.

[23] F. de Budan. *Nouvelle Méthode pour la Résolution des Équations Numériques d'un Degrè Quelconque.* Chez Courcier, Imprimeur-Libraire pour les Mathématiques, quai des Augustins, n. 57, 1807.

[24] N. Dershowitz and J.-P. Jouannaud. Notations for Rewriting. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 1991.

[25] R. Descartes. *La Géométrie*. 1637.

[26] K. Dewan and M. Bidkham. On the Eneström-Kakeya Theorem. *Journal of Mathematical Analysis and Applications*, 180(1):29–36, 1993.

[27] K. Dewan and N. K. Govil. On the Eneström-Kakeya Theorem. *Journal of Approximation Theory*, 42:239–244, 1984.

[28] G. M. Diaz-Toca and L. Gonzalez-Vega. Various New Expressions for Subresultants and Their Applications. *Applicable Algebra in Engineering, Communication and Computing*, 15(3):233–266, 2004.

[29] L. E. Dickson. *First Course in the Theory of Equations.* New York: J. Wiley & Sons, Inc., 1922.

[30] J. Divasón, S. Joosten, O. Kunčar, R. Thiemann, and A. Yamada. Efficient Certification of Complexity Proofs: Formalizing the Perron–Frobenius Theorem (Invited Talk Paper). In *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2018, page 2–13, 2018.

[31] J. Divasón, S. Joosten, R. Thiemann, and A. Yamada. A Perron-Frobenius Theorem for Jordan Blocks for Complexity Proving. In *16th International Workshop on Termination*, pages 30–34, 2018.

[32] A. Elgyütt. Root Isolation of High-Degree Polynomials. Master's Thesis, Masaryk University, Brno, 2017.

[33] J. Endrullis, J. Waldmann, and H. Zantema. Matrix Interpretations for Proving Termination of Term Rewriting. *Journal of Automated Reasoning*, 40:195–220, 2008.

[34] G. Eneström. Härledning af en allmä Formel för Antalet Pensionärer som vid en godtycklig Tidpunkt förefinnas inom en sluten Pensionskassa. *Öfversigt af Kongl. Vetenskaps–Akademiens Förhandlingar*, 50:405–415, 1893.

[35] J. Fourier. Sur l'Usage du Théorème de Descartes dans la Recherche des Limites des Racines. In *Bulletin des Sciences par la Société Philomathique de Paris*, pages 156–165, 181–187. 1820.

[36] G. Frobenius. *Über Matrizen aus positiven Elementen.* Sitzungsberichte der Preußischen Akademie der Wissenschaften zu Berlin, Göttingen, 1908.

[37] C. Fuhs. Transforming Derivational Complexity of Term Rewriting to Runtime Complexity. In *FroCos*, 2019.

[38] R. B. Gardner and N. K. Govil. On the Location of the Zeros of a Polynomial. *Journal of Approximation Theory*, 78(2):286–292, 1994.

[39] R. B. Gardner and N. K. Govil. Eneström-Kakeya Theorem and Some of Its Generalizations. *Current Topics in Pure and Computational Complex Analysis*, pages 171–199, 2014.

[40] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for Computer Algebra.* Kluwer Academic Publishers, 1992.

[41] S. Geršgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Bulletin de l'Académie des Sciences de l'URSS. Classe des Sciences Mathématiques et na*, 6:749–754, 1931.

[42] N. K. Govil and V. K. Jain. On the Eneström-Kakeya Theorem. *Annales Universitatis Mariae Curie-Skłodowska. Sectio A. Mathematica*, 27:13–18, 1973.

[43] N. K. Govil and V. K. Jain. On the Eneström-Kakeya Theorem, II. *Journal of Approximation Theory*, 22(1):1–10, 1978.

[44] N. K. Govil and Q. I. Rahman. On the Eneström-Kakeya Theorem. *Tôhoku Mathematical Journal, Second Series*, 20(2):126–136, 1968.

[45] M. A. Hirsch. Sur les Racines d'une Équation Fondamentale: Extrait d'une Lettre de M. A. Hirsch à M. I. Bendixson. *Acta Mathematica*, 25:367–370, 1902.

[46] D. Hofbauer and J. Waldmann. Termination of String Rewriting with Matrix Interpretations. In *Term Rewriting and Applications*, pages 328–342, 2006.

[47] A. Hurwitz. Über einen Satz des Herrn Kakeya. *Tôhoku Mathematical Journal, First Series*, 4:89–93, 1913.

[48] J. R. Johnson. *Algorithms for Polynomial Real Root Isolation.* PhD thesis, Ohio State University, 1992.

[49] D. E. Jordan, L. C. Clapp, and R. Y. Kain. Symbolic Factoring of Polynomials in Several Variables. *Commun. ACM*, 9(8):638–643, Aug. 1966.

[50] A. Joyal, G. Labelle, and Q. Rahman. On the Location of Zeros of Polynomials. *Canadian Mathematical Bulletin*, 10(1):53–63, 1967.

[51] S. Kahrs. Context Rewriting. In *Proceedings of the Third International Workshop on Conditional Term Rewriting Systems*, CTRS '92, pages 21–35, 1993.

[52] S. Kakeya. On the Limits of the Roots of an Algebraic Equation with Positive Coefficients. *Tôhoku Mathematical Journal, First Series*, 2:140–142, 1912.

[53] J. B. Kioustelidis. Bounds for Positive Roots of Polynomials. *Journal of Computational and Applied Mathematics*, 16(2):241–244, 1986.

[54] J.-L. Lagrange. Sur la Résolution des Équations Numériques. *Mémoires de l'Académie Royale des Sciences et Belles-lettres de Berlin XXIII*, pages 539–578, 1769.

[55] J.-L. Lagrange. *Traité de la Résolution des Équations Numeriques de tous les Degrés, avec des Notes. 4. Éd.* Œuvres de Lagrange. Gauthier-Villars, 1879.

[56] E. Laguerre. Sur la Détermination d'une Limite Supérieure des Racines d'une Équation et sur la Séparation des Racines. *Nouvelles annales de mathématiques : journal des candidats aux écoles polytechnique et normale*, 2e série, 19:97–105, 1880.

[57] W. Li and L. C. Paulson. Counting Polynomial Roots in Isabelle/HOL: A Formal Proof of the Budan-Fourier Theorem. *CoRR*, 2018.

[58] J. M. McNamee. A Bibliography on Roots of Polynomials. *Journal of Computational and Applied Mathematics*, 47(3):391–394, 1993.

[59] J. M. McNamee and V. Y. Pan. Chapter 12 – Low-Degree Polynomials. In *Numerical Methods for Roots of Polynomials - Part II*, volume 16 of *Studies in Computational Mathematics*, pages 527–556. 2013.

[60] A. Middeldorp, G. Moser, F. Neurauter, J. Waldmann, and H. Zankl. Joint Spectral Radius Theory for Automated Complexity Analysis of Rewrite Systems. In *Proceedings of the 4th International Conference on Algebraic Informatics*, CAI'11, pages 1–20, 2011.

[61] A. Mogbademu, S. Hans, and J. A. Adepoju. A Note On Eneström-Kakeya Theorem. *Nonlinear Analysis Real World Applications*, 20:139–145, 2015.

[62] G. Moser, A. Schnabl, and J. Waldmann. Complexity Analysis of Term Rewriting Based on Matrix and Context Dependent Interpretations. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 2 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 304–315, 2008.

[63] F. Neurauter, H. Zankl, and A. Middeldorp. Revisiting Matrix Interpretations for Polynomial Derivational Complexity of Term Rewriting. In *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 550–564, 2010.

[64] N. Obreshkov. *Verteilung und Berechnung der Nullstellen reeller Polynome*. Hochschulbücher für Mathematik. Deutscher Verlag der Wissenschaften, 1963.

[65] V. Sharma. Complexity of Real Root Isolation Using Continued Fractions. *Theor. Comput. Sci.*, 409(2):292–310, 2008.

[66] Ch. F. Sturm. Mémoire sur la Résolution des Équations Numérique. *Bulletin des Sciences de Férussac*, 11:419–425, 1829.

[67] J. J. Sylvester. Part I of a Memoir on the Dyalitic Method of Elimination. *Proceedings of the Royal Irish Academy*, 2:130–138, 1840.

[68] J. J. Sylvester. XXIII. A Method of Determining by Mere Inspection the Derivatives From Two Equations of Any Degree. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 16(101):132–135, 1840.

[69] J. J. Sylvester. On a Theory of the Syzygetic Relations of Two Rational Integral Functions, Comprising an Application to the Theory of Sturm's Functions, and That of the Greatest Algebraical Common Measure. *Philosophical Transactions of the Royal Society of London*, 143:407–548, 1853.

[70] R. Thiemann and A. Yamada. Formalizing Jordan Normal Forms in Isabelle/HOL. In *Proceedings of the 5th ACM SIGPLAN Conference on Certified Programs and Proofs*, CPP 2016, page 88–99, 2016.

[71] J. M. Thomas. Sturm's Theorem for Multiple Roots. *National Mathematics Magazine*, 15(8):391–394, 1941.

[72] A. A. Ungar. A Unified Approach for Solving Quadratic, Cubic and Quartic Equations by Radicals. *Computers & Mathematics with Applications*, 19(12):33–39, 1990.

[73] J. V. Uspensky. *Theory of Equations*. McGraw-Hill New York, 1948.

[74] M. B. Villarino. Quotient Polynomials with Positive Coefficients. *The Mathematical Gazette*, 98(542):250–255, 2014.

[75] A. J. H. Vincent. Mémoire sur la Résolution des Équations Numériques. *Mémoires de la Société Royale des Sciences, de L'Agriculture et des Arts, de Lille*, pages 1–34, 1834.

[76] A. J. H. Vincent. Note sur la Résolution des Équations Numériques. *Journal de Mathématiques Pures et Appliquées*, 1:341–372, 1836.

[77] L. Weisner. *Introduction to the Theory of Equations*. MacMillan, New York, 1938.

[78] L. Yang and B. Xia. Explicit Criterion to Determine the Number of Positive Roots of a Polynomial. *MM Research Preprints*, 15:134–145, 1997.

[79] D. Y. Yun. On Square-free Decomposition Algorithms. In *Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation*, SYMSAC '76, pages 26–35, 1976.