Computing Certificates of Members in Archimedean Quadratic Modules in $\mathbb{A}[X]$ and Certifying the Emptiness in Inconsistent Monogenic Archimedean Quadratic Modules in $\mathbb{A}[\overline{X}]$

by

José Abel Castellanos Joo

B.Tech., Universidad de las Américas Puebla, 2015
M.S., University of New Mexico, 2020

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

May, 2026

Dedication

To my family.

Acknowledgments

First, I would like to express my deepest gratitude to my advisor Prof. Deepak Kapur, for many years of patient guidance, support, and help in shaping the ideas presented in this thesis. Among many things, I appreciate his perseverance and persistent efforts to help me become a mature researcher, cultivate a critical mind to organize my own ideas, and develop resilience while facing challenging research problems.

I am immensely grateful to my committee members, Prof. Chenqi Mou, Prof. Naijun Zhan, and Prof. Shuang Luan, for spending their time reviewing my thesis and patience with me after submitting many drafts to them.

My thesis work could not have been possible without the fruitful collaboration and positive environment of the online Weekly Quadratic Modules seminar. I had the pleasure of meeting Prof. Chenqi Mou and Weifeng Shang during these discussions in 2022. Since then, I have learned so much during enlightening discussions and been involved in many research projects.

I want to thank my friends in Albuquerque for making me feel at home away from home: Jon David, Joonyeop Baek, Franklin Pezzuti Dyer, Andrew Geyko, Abigail Pribisova, Jeb Kilfoyle, Warren (Hoss) Craft and his wife Karen Yu, and Sarika Kumar.

Finally, I want to thank my family, especially my mother Narda Oliva Joo Ayar, my father José Abel Castellanos Toledo, and my sister Narda Castellanos Joo, for their constant support and unconditional love. *Muchas gracias, siempre, siempre.*

Computing Certificates of Members in Archimedean Quadratic Modules in $\mathbb{A}[X]$ and Certifying the Emptiness in Inconsistent Monogenic Archimedean Quadratic Modules in $\mathbb{A}[\overline{X}]$

by

José Abel Castellanos Joo

B.Tech., Universidad de las Américas Puebla, 2015
M.S., University of New Mexico, 2020

Ph.D., Computer Science, University of New Mexico, 2026

Abstract

Polynomials have been found to be a powerful tool over hundreds of years for modeling problems in numerous applications in science, engineering, medicine, and other domains. In the context of formal methods, polynomials arise in modeling in aerospace software and robotics, cyber-physical and hybrid systems, autonomous vehicles and controllers based on neural networks.

A quadratic module is a linear combination of polynomials in a set of generators (including the constant 1) with sum of squares polynomials as multipliers. The membership problem for a finitely generated quadratic module can be decided; however, computing a certificate exhibiting why it is nonnegative under the assumption that the generators are nonnegative, can be nontrivial.

A new symbolic algorithm is presented to compute sums of squares multipliers (certificates) to witness the membership of univariate polynomials in Archimedean quadratic modules. An algorithm is presented to compute a certificate for -1 in inconsistent quadratic modules in $\mathbb{A}[\overline{X}]$.

Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

Introduction

Hilbert's 17th problem asks whether a nonnegative multivariate polynomial can be represented as a sums of squares of rational functions. This problem was solved in the affirmative by Artin [2]. A natural generalization of Hilbert's problem extends to check the nonnegativity of a given polynomial over semialgebraic sets, which are the solution set of a system of polynomial inequalities. In [40, 85] Krivine and Stengle independently provided an algebraic characterization of this problem in the celebrated Positivstellensatz theorem. Ever since Krivine-Stengle's work, there has been considerable interest in finding algorithmic implementable solutions for special cases. Noteworthy are the results by Schmüdgen for preorderings [78], Putinar for quadratic modules [72], developing representations of positive and nonnegative polynomials on a semialgebraic set defined by a finite set of generators.

The complexity results in [63, 79] gave upper bounds on the degree of multipliers that appear in the representations of a polynomial in terms of the original generators. However, there has been limited progress in developing implementable algorithmic solutions for computing certificates exhibiting membership of a polynomial in a quadratic module; a notable exception is `RealCertify`, a package in

`Maple`, which is capable of computing certificates for strictly positive polynomial over Archimedean quadratic modules.

The importance of certificates that exhibit the details of a result produced by a computer algebra system has long been emphasized, especially by Kahan, Fateman, [29] and recently Davenport [28], among others. Certificate computation has received particular attention in the SMT (Satisfiability Modulo Theories) [13, 14], theorem proving [30], and more recently, in deep learning networks [39, 83, 84] due to the ability of such software to solve very large problems in numerous applications in engineering, hybrid systems, and other safety-critical systems. The dependence on the reliable results produced by such software in life-critical applications further emphasizes the need for certificates exhibiting the correctness of the results.

1.1 Motivation

Formal verification employs algorithms to provide high-level assurance of hardware and software systems by validating specifications, checking correctness properties, and ruling out undesirable critical states. Typically, these algorithms use logical or algebraic methods or a combination of both. How can *we* be sure that the algorithms used for this purpose do not contain errors? This fairly innocuous trust issue can be ameliorated by extending verification algorithms with checkable properties or certificates that can be processed by simpler mechanisms which, in principle, should be easier to verify.

Logical-based methods have been successful in producing certificates. Techniques such as proof logging [59, 62, 64], proof checking [32, 90], and others allow efficient extraction of certificates to validate the results obtained by these algorithms.

On the other hand, most algebraic-based methods have not been able to replicate

a similar situation in producing certificates. An exception of this is the theory of Gröbner bases as it is able to produce algebraic certificates of membership in ideals or infeasibility proofs in systems of equations via Nullstellensatz [38, 73]. Arguably, workflows involving computer algebra systems are limited to finding solutions to diverse problems, and the algorithms developed for these tasks are reasonably more complex, making it harder to replicate the same level of success using similar methods to logical methods. However, recent interest in integrating computer algebra techniques in reasoning tasks requires a better adaptation of the algebraic method to validate this combination. Our proposed work aims to contribute to closing the bridge for these requirements.

Existing solutions, as mentioned in 2.2.6, use SDP solvers. Although this approach offers good performance approximating results, projection and rounding routines [31, 36, 68, 74] are often needed to obtain exact results that are based on various conditions. Additionally, the dependency on SDP solvers can impact the implementation since it is necessary to find a suitable configuration to many parameters in these solvers. The proposed methods in our work avoid using SDP solvers in order to find exact certificates using symbolic techniques.

Similarly, the certificates problem can be adapted in proof-producing satisfiability modulo theory (SMT) solvers. An SMT solver aims to either find a satisfiable solution or a proof of unsatisfiability to a logical formula in some language like equality theory, theory of integers/reals, theory of arrays, etc. Typically, it combines a logic engine and several theory solvers, which effectively communicate in order to solve or refute a given logical query. For efficiency purposes, some theory solvers do not provide fully constructive proofs, as it is acknowledged that some facts must be included in the theories involved [60]. The methods addressed in this thesis leverage this situation by providing fully constructive methods.

1.2 Preliminaries

Let $\mathbb{A} \subseteq \mathbb{R}$ be the set of real algebraic numbers, and $\mathbb{A}[\overline{X}] = \mathbb{A}[X_1, \ldots, X_n]$. Given a polynomial $f \in \mathbb{A}[\overline{X}]$, we denote with $\mathcal{Z}(f)$ the real zeros of f. Let $\sum \mathbb{A}[\overline{X}]^2$ denote $\left\{\sum_{i=1}^n f_i^2 \mid f_i \in \mathbb{A}[\overline{X}]\right\}$ which is the set of sums of squares in $\mathbb{A}[\overline{X}]$. The semialgebraic set of $G = \{g_1, \ldots, g_s\} \subseteq \mathbb{A}[\overline{X}]$, denoted by $\mathcal{S}(G)$, corresponds to a subset of $\mathbb{R}^n$ defined by $\mathcal{S}(G) := \{\boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) \geq 0, 1 \leq i \leq s\}$. Given $S \subseteq \mathbb{R}^n$, let $\mathrm{Pos}(S)$ (resp. $\mathrm{Pos}^+(S)$) denote the set of nonnegative polynomials (resp. strictly positive) over S, i.e., $\mathrm{Pos}(S) := \{f \in \mathbb{A}[\overline{X}] \mid f(\boldsymbol{x}) \geq 0, \forall \boldsymbol{x} \in S\}$.

Definition 1.1 A **quadratic module** in $\mathbb{A}[\overline{X}]$ is a subset that is closed under addition, closed under multiplication with squares in $\mathbb{A}[\overline{X}]$, and contains 1. Given a set of polynomials $G = \{g_1, \ldots, g_s\} \subseteq \mathbb{A}[\overline{X}]$, the quadratic module generated by G (abbreviated $\mathrm{QM}(G)$) is the set

$$\left\{\sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i \mid \sigma_i \in \sum \mathbb{A}[\overline{X}]^2, 0 \leq i \leq n\right\}$$

A **preordering** in $\mathbb{A}[\overline{X}]$ is a subset that is closed under addition, closed under multiplication, and contains the sums of squares $\sum \mathbb{A}[\overline{X}]^2$ of $\mathbb{A}[\overline{X}]$. The preordering generated by a set of polynomials G (abbreviated $\mathrm{PO}(G)$) is the set

$$\left\{\sum_{(e_1,\ldots,e_s)\subseteq\{0,1\}^s} \sigma_{(e_1,\ldots,e_s)} \cdot g_1^{e_1} \cdots g_s^{e_s} \mid \sigma_{(e_1,\ldots,e_s)} \in \sum \mathbb{A}[\overline{X}]^2\right\}$$

Definition 1.2 A quadratic module Q is called **Archimedean** if Q contains a polynomial of the form $N - \sum_{i=1}^n X_i^2$ where $N \in \mathbb{N}^+$.

Now, we focus our definitions on the context of $\mathbb{A}[X]$. A compact semialgebraic set S is a closed and bounded semialgebraic set. In the univariate case, compact

semialgebraic sets can be represented as a **finite ordered union** [1] of **closed intervals**, distinguishing two particular kinds, **regular intervals** $[a_i, b_i]$, $a_i < b_i$ as well as **isolated points** represented as $[a_i, a_i]$, i.e., $S = \bigcup_{i=0}^{k} [a_i, b_i]$. Given a semialgebraic set S, we denote by S_{isol} the set of isolated points of S.

The semialgebraic set of a given univariate polynomial f, $\mathcal{S}(f)$, is a union of intervals in $\mathbb{R}$. The endpoints of these intervals are the real roots of f; thus, to compute $\mathcal{S}(f)$ is enough to use real root isolation algorithms [15]. To determine the multiplicity of a root of a given polynomial, we use the fact that a root $a \in \mathbb{R}$ of f has multiplicity m if and only if $f(a) = \frac{df}{dX}(a) = \cdots = \frac{d^{m-1}f}{dX^{m-1}}(a) = 0$ and $\frac{d^m f}{dX^m}(a) \neq 0$; the latter can be tested by computing the formal derivatives of the polynomial f and checking if the evaluation of a in the derivatives of f is zero. Motivated from these observations, we formalize them in the following definition as these concepts will be used in the thesis.

Definition 1.3 Let f be a polynomial in $\mathbb{R}[X]$. Given a point $a \in \mathbb{R}$, the **order** of f at a, denoted $\mathrm{ord}_a(f)$, is the minimum natural number n such that $\frac{d^n f}{dX^n}(a) \neq 0$. The **epsilon-sign** of f at a, denoted $\epsilon_a(f)$, is defined as 1 if $\frac{d^n f}{dX^n}(a) > 0$ where $n = \mathrm{ord}_a(f)$ and -1 otherwise.

1.2.1 Problem statement

We refer to the sums of squares σ_i's in Definition 1.1 as the **certificates** for members of a quadratic module. The **certificate problem for quadratic modules** is, given a set of generators G and a member $f \in \mathrm{QM}(G)$, to find the sums of squares multipliers $\sigma_0, \sigma_1, \ldots, \sigma_s$ such that $f = \sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i$. We refer to the polynomial f as the **input polynomial**. Similarly, the **certificate problem for preorderings** seeks to find the sums of squares multipliers σ_e where $e = (e_1, \ldots, e_s) \in \{0, 1\}^s$ such

[1] An ordered union of intervals satisfies $b_i < a_{i+1}$ for every $0 \leq i \leq n-1$.

that $f = \sum_{e \in \{0,1\}^s} \sigma_e \cdot g_1^{e_1} \cdots g_s^{e_s}$. This thesis is concerned with the computational aspects of finding certificates for members in Archimedean quadratic modules in $\mathbb{A}[X]$.

By definition, preorderings are closed under multiplication, whereas quadratic modules are generally not. In case an Archimedean quadratic module is a preordering also, which is the case of univariate structures, the certificate problem for a given polynomial in such a structure has a subtle distinction, depending upon whether the certificate is being sought using its representation in a quadratic module or a preordering.

Algorithmic solutions to problems involving quadratic modules for the univariate case have been discussed in the literature. To name a few, we have the following:

- The membership problem: Given a polynomial f and a quadratic module Q, as defined above, decide whether $f \in Q$ or not.
- Equivalent generators of a quadratic module: Given a set of generators G_1 of polynomials, find another set G_2 with fewer generators such that $\mathrm{QM}(G_1) = \mathrm{QM}(G_2)$.

Finding an equivalent set of generators has been discussed in the literature within different contexts for preorderings [2] and quadratic modules in the univariate case. Natural generators were introduced in [41, 42] to provide conditions for whether a preordering is saturated or not. If the associated semialgebraic set is compact, then it is possible to compute an equivalent quadratic module using three generators [4], two generators [81], or generalized natural generators [3].

The decision procedure for testing membership in an Archimedean quadratic module provides several insights about invariants and the rich structure of this al-

[2] A preordering is a quadratic module that is also closed under multiplication.

gebraic structure; however, it does not provide a constructive approach to finding a sum of squares certificate witnessing the membership of a given polynomials. These certificates have several applications in verification, where explicit constructions are often needed to verify goals and assertions. Although most applications rely on the multivariate case, understanding the basic structure of the quadratic module for univariate polynomials can provide insights into the former.

Other lines of research cover similar cases and consider different assumptions to our problem in consideration; in [55], the authors find weighted sums of Hermitan squares decomposition of nonnegative trigonometric polynomials with Gaussian coefficients over the unit circle using complex root isolation and semidefinite programming techniques. In [8], the authors address the problem of computing certificates of nonnegative polynomials over finite semialgebraic sets in the context of multivariate polynomials; however, it requires additional assumptions. In particular, the ideal I associated with the equality constraints is zero-dimensional and that the ideals $\langle f \rangle$ and $\langle I : f \rangle$ are coprime where f is the input polynomial.

1.2.2 Motivating example

In this section, we discuss key points of the algorithms developed in this thesis for the certificate computation of members in Archimedean Quadratic modules using an example and contrasting it with methods in the existing literature. The first method is the "naive" approach. Given a member of a quadratic module, this method exhaustively searches for the coefficients of a fixed degree polynomials that are squares in a summand of a fixed number of terms.

The second method is the Extended Gram approach. It utilizes semidefinite programming to find matrices that encode information about the sums of squares multipliers of the certificate of the member. While in theory this approach can

be exact, current implementations can avoid a hybrid approach using numerical solvers for the semidefinite programming parts resulting in approximate and non-exact solutions.

Finally, we describe our method. We use the algorithm SaturatedCert presented in Chapter 4 as the discussed example belongs to the saturated case. Our solution is symbolic and exact, and the degree of the certificates obtained is the same as in the Extended Gram method approach.

Membership check

Let us consider the set of generators $G = \{g_1, g_2\}$ where $g_1 = -(X+2)(X-1), g_2 = -(X+1)(X-2)$ and $f = -(X+1)(X-1)$ as input polynomial. First, we need to check if f belongs to QM(G). We use the decision method in Theorem 2.18 from [4] to check membership. This method requires information about the semialgebraic set of the generator set G, the semialgebraic set of the input polynomial f and the multiplicities of the endpoints in the factors of G and f. As we consider polynomials with rational coefficients, there are algorithms to isolate and represent algebraic numbers, which are the roots for the type of polynomials considered in the certificate problem.

The first step computes $\mathcal{S}(G)$, which is the intersection of the semialgebraic sets of $\mathcal{S}(g_1)$ and $\mathcal{S}(g_2)$. To determine the semialgebraic sets of each polynomial, we use the sign variation method that identifies the sign changes of a polynomial through $\mathbb{R}$. First, we focus on the semialgebraic set of g_1. The roots of g_1 are -2 and -1, both with multiplicity 1, and its leading coefficient is -1. The sign changes of g_1 over $\mathbb{R}$ are negative from $(-\infty, -2)$, 0 at -2, positive from $(2, 1)$, 0 at 1, and negative from $(1, \infty)$; therefore, we have $\mathcal{S}(g_1) = [-2, 1]$. Similarly, we find $\mathcal{S}(g_2) = [-1, 2]$; therefore, the semialgebraic of $\mathcal{S}(G)$ is $[-1, 1]$ as it is the intersection of $[-2, 1]$ and

$[-1, 2]$. Using the sign variation approach, we find $\mathcal{S}(f)$ to be $[-1, 1]$.

The input polynomial f is a nonnegative polynomial over $\mathcal{S}(G)$ and is not strictly positive polynomial, as it shares common zeros at the endpoints -1 and 1 of $\mathcal{S}(G)$. We also need to check conditions involving the multiplicity of endpoints in the factors of the input polynomial.

- The endpoint -1 is a left boundary point of $\mathcal{S}(G)$. We need to compute $\kappa^+_{-1}(G)$ which is defined as the minimum multiplicity of the generators at -1 such that these are odd and the epsilon sign at -1 is 1. The multiplicity of g_1 at -1 is 0. The multiplicity of g_2 at -1 is 1 and its epsilon sign at -1 is 1; hence $\kappa^+_{-1}(G) = 1$.

  Then, we check that $\text{ord}_{-1}(f) - \kappa^+_{-1}(G) = 0$ is even. Hence, the condition in item (i) of Theorem 2.18 [4] is satisfied.

- The endpoint 1 is a right boundary point of $\mathcal{S}(G)$. We need to compute $\kappa^-_1(G)$ which is defined as the minimum multiplicity of the generators at 1 such that these are odd and the epsilon sign at 1 is 1. The multiplicity of g_1 at 1 is 1 and its epsilon sign at 1 is 1. The multiplicity of g_2 at 1 is 0; hence $\kappa^-_1(G) = 1$.

  Then, we check that $\text{ord}_1(f) - \kappa^-_1(G) = 0$ is even. Hence, the condition of item (ii) from Theorem 2.18 [4] is satisfied.

Therefore, f is a member of $\text{QM}(G)$.

Naive approach

Once there is a formal guaranty of the membership of a polynomial in a quadratic module, it is possible to find its certificates using an exhaustive search for the sums of squares involved by instantiating templates of sums of squares with unknown

coefficients and solving a set of equations to find a value for these coefficients. This template controls the number of summands m and the degree d of the polynomials in the sums of squares template, i.e., for our example we have the following.

$$\begin{aligned} f &= \sum_{i=1}^{m} (a_{0,i,0} + a_{0,i,1}X + \cdots + a_{0,i,d}X^d)^2 \\ &+ \sum_{i=1}^{m} (a_{1,i,0} + a_{1,i,1}X + \cdots + a_{1,i,d}X^d)^2 \cdot g_1 \\ &+ \sum_{i=1}^{m} (a_{2,i,0} + a_{2,i,1}X + \cdots + a_{2,i,d}X^d)^2 \cdot g_2 \end{aligned}$$

The weakness of this approach is the dependence on the parameters m and d. Certainly, an enumeration of all possible pairs for these values could guide the search for a certificate. Setting $m = d = 1$, we obtain the template

$$f = (a_{0,1,0} + a_{0,1,1}X)^2 + (a_{1,1,0} + a_{1,1,1}X)^2 \cdot g_1 + (a_{2,1,0} + a_{2,1,1}X)^2 \cdot g_2$$

and would need to solve the following equations:

$$\begin{aligned} 1 - a_{0,1,0}^2 - 2a_{1,1,0}^2 - 2a_{2,1,0}^2 &= 0 \\ -2a_{0,1,0}a_{0,1,1} + a_{1,1,0}^2 - 4a_{1,1,0}a_{1,1,1} - a_{2,1,0}^2 - 4a_{2,1,0}a_{2,1,1} &= 0 \\ -1 - a_{0,1,1}^2 + a_{1,1,0}^2 + 2a_{1,1,0}a_{1,1,1} - 2a_{1,1,1}^2 + a_{2,1,0}^2 - 2a_{2,1,0}a_{2,1,1} - 2a_{2,1,1}^2 &= 0 \\ 2a_{1,1,0}a_{1,1,1} + a_{1,1,1}^2 + 2a_{2,1,0}a_{2,1,1} - a_{2,1,1}^2 &= 0 \\ a_{1,1,1}^2 + a_{2,1,1}^2 &= 0 \end{aligned} \tag{1.1}$$

However, this system of equations has no solution over the reals. Setting $m = 1$ and $d = 2$, we obtain a larger system of polynomial equations, but implementations such as `Mathematica` or `Maple` are not able to find a solution over the reals in a reasonable time. As we shall see later, our approach is able to find a certificate within these constraints in 0.163 seconds.

If we add partial solutions to the systems of equations in (1.1) using the certificate obtained in our proposed method, `Mathematica`'s command, `FindInstance`, still struggles to find solutions to the systems. The table below reports the execution time in seconds, up to a limit of 30 seconds; computations that exceed this limit are aborted.

# of partial solutions	Minimum	Median	Average	Maximum	Successful runs
6	0.015574	0.0191525	0.0192074	0.02764	84/84
5	0.016363	0.0200115	0.0217022	0.060212	126/126
4	0.017195	0.0278745	0.0769109	2.53658	120/126
3	0.01968	0.07222	0.4273	7.32189	72/84
2	0.336124	0.951686	3.46394	12.4142	16/36
1	N/A	N/A	N/A	N/A	0/9

Table 1.1: Execution times of `FindInstance` with added partial solutions for the system (1.1).

Extended Gram matrix approach

This approach uses semidefinite programming (SDP) to find a semidefinite positive matrix that encodes the sums of squares in the certificate of members in quadratic modules together with constraints involving the coefficients of the input polynomial. Similarly to the Naive approach from the previous section, this approach also fixes the degree of the sums of square polynomials; therefore, this method also requires a sequential search to find a suitable degree for the certificates.

SDP solvers are fast, since the majority of implementations [19, 87, 88, 91] use interior point methods that run in polynomial time. However, existing software returns numerical/floating point solutions. A limitation of this approach is that the certificates obtained from these solutions are approximate solutions and quite often non-exact.

Using our previous example, we assume the existence of a certificate of degree at most 2, hence the "monomial basis" is $m = \begin{pmatrix} 1 \\ X \\ X^2 \end{pmatrix}$. Let $g_0 := 1$, we can express f as $\sum_{i=0}^{s} (m^T A_i m) \cdot g_i$ for some positive semidefinite matrices $A_i = \begin{pmatrix} a_{i,1,1} & a_{i,1,2} & a_{i,1,3} \\ a_{i,1,2} & a_{i,2,2} & a_{i,2,3} \\ a_{i,1,2} & a_{i,2,3} & a_{i,3,3} \end{pmatrix}$. Using `Mathematica`'s command `SemidefiniteOptimization` method, we ask the matrices A_i to be positive semidefinite with the equation constraints involving the coefficients of the input polynomial

$$\begin{aligned}
a_{0,1,1} + 2a_{1,1,1} + 2a_{2,1,1} &= 1 \\
2a_{0,1,2} - a_{1,1,1} + 4a_{1,1,2} + a_{2,1,1} + 4a_{2,1,2} &= 0 \\
2a_{0,1,3} + a_{0,2,2} - a_{1,1,1} - 2a_{1,1,2} + 4a_{1,1,3} + 2a_{1,2,2} - a_{2,1,1} + 2a_{2,1,2} + 4a_{2,1,3} + 2a_{2,2,2} &= -1 \\
2a_{0,2,3} - 2a_{1,1,2} - 2a_{1,1,3} - a_{1,2,2} + 4a_{1,2,3} - 2a_{2,1,2} + 2a_{2,1,3} + a_{2,2,2} + 4a_{2,2,3} &= 0 \\
a_{0,3,3} - 2a_{1,1,3} - a_{1,2,2} - 2a_{1,2,3} + 2a_{1,3,3} - 2a_{2,1,3} - a_{2,2,2} + 2a_{2,2,3} + 2a_{2,3,3} &= 0 \\
-2a_{1,2,3} - a_{1,3,3} - 2a_{2,2,3} + a_{2,3,3} &= 0 \\
-a_{1,3,3} - a_{2,3,3} &= 0
\end{aligned}$$

The output obtained is

$$A_0 = \begin{pmatrix} 0.333333 & 2.783040 \times 10^{-17} & -0.333333 \\ 2.783040 \times 10^{-17} & 3.871773 \times 10^{-8} & -2.073324 \times 10^{-17} \\ -0.333333 & -2.073324 \times 10^{-17} & 0.333334 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 0.166667 & 0.166667 & -1.197158 \times 10^{-14} \\ 0.166667 & 0.166667 & 2.533843 \times 10^{-8} \\ -1.197158 \times 10^{-14} & 2.533843 \times 10^{-8} & 2.533843 \times 10^{-8} \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0.166667 & -0.166667 & -1.198342 \times 10^{-14} \\ -0.166667 & 0.166667 & -2.533843 \times 10^{-8} \\ -1.198342 \times 10^{-14} & -2.533843 \times 10^{-8} & 2.533844 \times 10^{-8} \end{pmatrix}$$

Since A_i are positive semidefinite matrices of dimension 3×3, we obtain a sum of squares decomposition of $m^T A_i m$ by computing a Cholesky decomposition of $A_i = B_i^T B_i$, so $m^T A_i m = m^T B_i^T B_i m = (B_i m)^T (B_i m) = \sum_{j=1}^{3} \langle B_{ij}, m \rangle^2$ where B_{ij} is the j^{th} row of the matrix B_j and $\langle , \rangle$ denotes the dot product. We obtain

$$B_0 = \begin{pmatrix} 0.57735 & 4.820366 \times 10^{-17} & -0.57735 \\ 0 & 0.000196 & 3.606860 \times 10^{-14} \\ 0 & 0 & 0.000196 \end{pmatrix}$$

$$B_1 = \begin{pmatrix} 0.408248 & 0.408248 & -2.932427 \times 10^{-14} \\ 0 & 0.000253 & 0.000100 \\ 0 & 0 & 0.000123 \end{pmatrix}$$

$$B_2 = \begin{pmatrix} 0.408248 & -0.408248 & -2.935326 \times 10^{-14} \\ 0 & 0.000253 & -0.000100 \\ 0 & 0 & 0.000123 \end{pmatrix}$$

Finally, the certificate obtained using this method is

$$\begin{aligned} f &= \sum_{i=0}^{s} (m^T A_i m) \cdot g_i \\ &\approx (0.57735 + 4.82037 * 10^{-17} X - 0.57735 X^2)^2 \\ &\quad + (0.000196768 X + 3.60686 * 10^{-14} X^2)^2 + (0.000196768 X^2)^2 \\ &+ ((0.408248 + 0.408248 X - 2.93243 * 10^{-14} X^2)^2 + (0.000253093 X + 0.000100115 X^2)^2 \\ &\quad + (0.000123755 X^2)^2) \cdot g_1 \\ &+ ((0.000253093 X - 0.000100115 X^2)^2 + (0.408248 - 0.408248 X - 2.93533 * 10^{-14} X^2)^2 \\ &\quad + (0.000123755 X^2)^2) \cdot g_2 \\ &= 1 + 2.77705 * 10^{-16} X - X^2 \\ &\quad - 9.69776 * 10^{-17} X^3 + 6.85159 * 10^{-8} X^4 - 2.36719 * 10^{-17} X^5 - 5.06769 * 10^{-8} X^6 \end{aligned}$$

The approximation found by the method is "close" to the original input polynomial f, but not exact. Additionally, in the Cholesky decomposition used to obtain the sum of squares decomposition with this method, the matrices A_i need to be positive semidefinite to have positive eigenvalues.

Our approach

Our approach is purely symbolic, does not rely on numerical approaches, and is exact. First, we compute an equivalent intermediate set of generators called the natural generators of G introduced in [42]. In our example, we have $n_1 = X + 1$ and $n_2 = -(X - 1)$. Then, we compute the certificate of f in terms of the preordering generated by the natural generators $\mathrm{PO}(n_1, n_2)$. In this case, we note that f is the product of n_1 and n_2.

To lift the certificate of f from $\mathrm{PO}(n_1, n_2)$ to $\mathrm{QM}(n_1, n_2)$, we compute the certificate of the product of n_1 and n_2 in $\mathrm{QM}(n_1, n_2)$. For the latter, we obtain $n_1 n_2 = \frac{1}{2}(X - 1)^2 \cdot n_1 + \frac{1}{2}(X + 1)^2 \cdot n_2$. Finally, to obtain a certificate of f in the quadratic module generated by G, we obtain certificates of natural generators in terms of $\mathrm{QM}(G)$. This step is done utilizing the Basic Lemma construction as follows:

1. First, we obtain a certificate of n_1 using g_2 applying the Basic Lemma with n_1 and $\frac{g_2}{n_1} = -(X - 2)$. We obtain $\frac{1}{3}n_1 + \frac{1}{3}\frac{g_2}{n_1} = 1$; therefore, multiplying n_1 by the previous equation we have $n_1 = \frac{1}{3}n_1^2 + \frac{1}{3} \cdot g_2$, which is a certificate of n_1 in $\mathrm{QM}(g_2)$.

2. Now, we obtain a certificate of n_2 using g_1 applying the Basic Lemma with n_2 and $\frac{g_1}{n_2} = (X + 2)$. We obtain $\frac{1}{3}n_2 + \frac{1}{3}\frac{g_1}{n_2} = 1$; therefore, multiplying n_2 by the previous equation we obtain $n_2 = \frac{1}{3}n_2^2 + \frac{1}{3} \cdot g_1$, which is a certificate of n_2 in $\mathrm{QM}(g_1)$.

Finally, we lift the certificate of f from $\mathrm{QM}(n_1, n_2)$ to $\mathrm{QM}(G)$ substituting the certificates of n_1 and n_2, respectively. We obtain

$$\begin{aligned}
f &= \frac{1}{2}(X-1)^2 \cdot n_1 + \frac{1}{2}(X+1)^2 \cdot n_2 \\
&= \frac{1}{2}(X-1)^2(\frac{1}{3}n_1^2 + \frac{1}{3} \cdot g_2) + \frac{1}{2}(X+1)^2(\frac{1}{3}n_2^2 + \frac{1}{3} \cdot g_1) \\
&= (\frac{1}{6}((X-1)(X+1))^2 + \frac{1}{6}(X-1)^2 \cdot g_2) \\
&\quad + (\frac{1}{6}((X+1)(X-1))^2 + \frac{1}{6}(X+1)^2 \cdot g_1) \\
&= \frac{1}{3}((X-1)(X+1))^2 + \frac{1}{6}(X+1)^2 \cdot g_1 + \frac{1}{6}(X-1)^2 \cdot g_2
\end{aligned}$$

The certificate obtained requires at most 1 summand, and the polynomials are of degree at most 2.

1.3 Contributions

The key contributions of the thesis are (i) an algorithm for computing a membership certificate of a polynomial in a finitely generated univariate Archimedean quadratic module and (ii) an effective method for computing a certificate of the emptiness of a quadratic module in the general multivariate case.

The algorithm in the univariate case relies on two constructions, which are of independent interest: (i) saturated: a special algorithm for computing a membership certificate for a saturated univariate quadratic module, (ii) monogenic: a special algorithm for computing a membership certificate from a univariate quadratic module generated by a single bounded polynomial. The first construction is used repeatedly in the general algorithm. It can require the compactification step; a technique introduced in Chapter 3 of this thesis. Compactification takes a polynomial f and

generates another polynomial with a bounded semialgebraic set and shares the same zeros as f, and most importantly, a membership certificate for the compactified polynomial can be computed from a certificate of f.

Motivated by the algorithm for a saturated quadratic module, methods are developed to minimize the multiplicities of common factors between an input polynomial and the generators of a quadratic module, which leads to simpler modular constructions of certificates of lower degrees. An algebraic technique has also been developed to effectively ignore common factors of even multiplicities between the input polynomial and generators, which are due to isolated points of the semialgebraic set of the generators.

Another important insight enables distinction between **pure** and **mixed** polynomials with respect to a generator set; this can help in determining whether an input polynomial can be generated by a single polynomial in the quadratic module. In both cases, we keep track of the generators needed for each root of the input polynomial, which allows us to instantiate several independent monogenic certificate subproblems to find the certificate of the nonnegative factors of the input polynomial.

In the last chapter, the thesis reports a method to certify the emptiness of a quadratic module in $\mathbb{A}[\overline{X}]$ by finding a certificate of -1. This is achieved by computing the certificate of -1 using a identity similar to Bézout of -1 that involves globally nonnegative polynomials as multipliers.

1.4 Organization of the thesis

This thesis addresses the problem of computing exact certificates for a polynomial belonging to an Archimedean quadratic module in the univariate case generated by a finite number of generators. These do not necessarily need to be strictly positive over

the semialgebraic set of the generator as previously discussed. Our methods typically introduce intermediate quadratic modules as some set of generators are easier to compute direct certificate; to obtain a certificate in the original set of generators, we rely on the "lifting" operation that composes these certificates. For more details on this operation, the reader should refer to Appendix A.1. The thesis considers two fundamental subcases of the general problem, the saturated and monogenic case. Building upon these results, we obtain a complete method for the general case.

In Chapter 4, an algorithm is given for computing a certificate of a polynomial in the univariate case for an Archimedean saturated quadratic module. The algorithm uses **natural (choice)** generators introduced by Kuhlmann and Marshall [41, 42, 58] as a way to describe generators of a **saturated** quadratic module to include all nonnegative polynomials in $\mathbb{R}[X]$ over a given semialgebraic set $\mathcal{S}(G)$. We emphasize that a key novelty of the approach handles nonnegative polynomials in contrast to other approaches [6, 11, 12, 53, 79] that cover the case for strictly positive polynomials derived by Putinar's Positivstellensatz [72].

To compute a certificate in terms of the natural generators of $\mathcal{S}(G)$, a construction in Augustin's thesis [4] is adapted, which gives a certificate using the preordering representation; to get a certificate in the quadratic module representation, certificates of products of natural generators must be constructed; this is shown in Section 4.2. Section 4.3 describes a method for computing certificates of natural generators in terms of the original set of generators.

The operations in the presented algorithms are effective, as it only manipulates real algebraic numbers which are computable [15, 43]. Algorithms are given to compute the certificates of each element in the split generators in terms of the original generators. Combining these results, an algorithm is given for computing a certificate of a given polynomial in terms of the original generators.

The presented construction uses a lemma from [42, Theorem 3.2 p. 590] to split factors of the members and a theorem from [76, Proposition 4.8 p. 1058] to find sums of squares multipliers that avoid products between generators; We describe algorithms to perform these results constructively in Section 2.1.7. Section 4.5 discusses an implementation in `Maple` and compares several examples with `RealCertify` [50] and `MomentPolynomialOpt` [9], software packages in `Maple` and `Julia` that compute certificates in Archimedean quadratic modules for strictly positive polynomials.

Chapter 5 introduces the MonogenicCert algorithm that computes a certificate of members in Archimedean monogenic quadratic modules by exploiting preordering properties and simplification of both the input polynomial and the generator using the concept of redundant factors. Section 5.1 discusses methods to simplify both the generator and the input polynomial by removing sums of squares factors that are unnecessary in sums of squares certificate. Section 5.3 presents algorithms to compute the certificates of members in an Archimedean monogenic quadratic module $\mathrm{QM}(g)$. The approach "splits" the input polynomial into three factors: a strictly positive over $\mathcal{S}(g)$, a sequence of "in-between" factors, and a nonnegative factor that shares common zeros with the generator. Section 5.3.4 discusses the experimental results obtained by running an implementation in `Maple` of the algorithms presented using different examples.

In Chapter 6, we discuss a method to compute a certificate of members $f \in \mathrm{QM}(G)$. The high level description of the approach is the following:

1. If there exists a generator $g \in G$ such that $f \in \mathrm{QM}(g)$, we use MonogenicCert to obtain a certificate of f.

2. Otherwise, we obtain the strictly positive factor f_{pos} of f and compute an intermediate set of generators $H_{f,G}$ which have the property that a certificate for the products of its members is computable in $\mathrm{QM}(G)$.

3. Multiplying f_{pos} and the nonnegative factor of f from $H_{f,G}$ produces f multiplying a strictly positive polynomial p with respect to $\mathcal{S}(G)$. Finally, to obtain a certificate of f with respect to $\mathrm{QM}(H_{f,G})$, we use the Basic Lemma on f and p.

We also address the problem of computing certificates in the equivalent 2-element set of generators $\{g_1, g_2\}$ from [81]. Given a polynomial $h \in \mathrm{QM}(g_1, g_2)$, we express h as the product of two polynomials: h_{pos} and $h_{non-neg}$ where h_{pos} is strictly positive over $S = \mathcal{S}(g_1, g_2)$ of maximal degree, i.e., there is no other factor h_1 of h that is strictly positive over S and $\deg(h_1) > \deg(h_{pos})$.

We compute a certificate of h_{pos} using the Certificate algorithm in [82]. For the $h_{non-neg}$ polynomial, we use NonNeg to compute nonnegative factors $h_{non-neg,i}$ of $h_{non-neg}$. We compute a certificate for each $h_{non-neg,i}$ using the QMCert algorithm.

Finally, to compute a certificate of h, we combine the certificates of each nonnegative factor and the strictly positive factor multiplying them and rearranging the sums of squares multipliers. Notice that, for the latter, such multiplication might bring the product of the generators g_1, g_2 into the representation. To lift the resulting representation to the quadratic module structure, we replace $g_1 g_2$ with its certificate in $\mathrm{QM}(g_1, g_2)$; hence, the resulting representation of h is lifted to a representation in $\mathrm{QM}(g_1, g_2)$.

Chapter 7 addresses the problem of certifying the emptiness of a monogenic quadratic module in $\mathbb{A}[\overline{X}]$. From Positivstellensatz , this kind of quadratic module contains all the polynomials in the ring of polynomials including the constant -1. We develop a method to find a certificate of -1 in terms of $\mathrm{QM}(g)$.

Chapter 2

Background and Related Work

This chapter contains background information on univariate Archimedean quadratic modules. Section 2.1 provides the notation and basic concepts needed for the thesis work and the constructive results that were used in the methods developed in this thesis. In Section 2.2, we discuss the state of the art in the field of decision procedures and certificate algorithms in quadratic modules.

2.1 Background

This section discusses several aspects of Archimedean quadratic modules in $\mathbb{R}[X]$. Mostly, we review results to decide the membership of a polynomial in a quadratic module and associated definitions for the latter. The discussion about **natural generators** is important especially for Chapter 4 as our method to compute certificates in the saturated case uses as an intermediate step the computation of certificate in the preordering generated by these generators.

2.1.1 Positivstellensatz

We begin this section by stating the Stengle/Krivine's Positivstellensatz [40, 85], a seminal result that laid the foundations of real algebraic geometry by establishing correspondence between semialgebraic sets with algebraic identities. The following version is taken from [57].

Theorem 2.1 *Suppose G is a finite subset of $\mathbb{R}[\overline{X}]$, $S = \mathcal{S}(G)$, $P = \mathrm{PO}(G)$ and $f \in \mathbb{R}[\overline{X}]$. Then*

1. *$f > 0$ on S if and only if there exist $p, q \in P$ such that $pf = 1 + q$.*
2. *$f \geq 0$ on S if and only if there exist an integer $m \geq 0$ and $p, q \in P$ such that $pf = f^{2m} + q$.*
3. *$f = 0$ on S if and only if there exists an integer $m \geq 0$ such that $-f^{2m} \in P$.*
4. *$S = \emptyset$ if and only if $-1 \in P$.*

Subsequent results in the area provided further refinements to the algebraic identities below under additional assumptions. Let $g_1, \ldots, g_s \subseteq \mathbb{R}[\overline{X}]$ and $G = \{g_1, \ldots, g_s\}$. The Schmüdgen's Positivstellensatz theorem [78] states that for any strictly positive polynomial f over a compact $\mathcal{S}(G)$, $f \in \mathrm{PO}(G)$. Putinar further simplified Schmüdgen's representation showing that for any strictly positive polynomial f over a compact $\mathcal{S}(G)$, if $\mathrm{QM}(G)$ is Archimedean then $f \in \mathrm{QM}(G)$. We state these results as follows [1].

Theorem 2.2 *[78] Suppose that the basic closed semialgebraic set $\mathcal{S}(g_1, \ldots, g_s)$ is compact. Then for every polynomial $f \in \mathbb{R}[\overline{X}]$, $f > 0$ on $\mathcal{S}(g_1, \ldots, g_s)$ implies $f \in \mathrm{PO}(g_1, \ldots, g_s)$.*

[1] The versions are taken from [63]

Theorem 2.3 *[72] Let* $g_1, \ldots, g_s \subseteq \mathbb{R}[\overline{X}]$. *Suppose that the quadratic module* $\mathrm{QM}(g_1, \ldots, g_s)$ *is Archimedean. Then for every polynomial* $f \in \mathbb{R}[\overline{X}]$, $f > 0$ *on* $\mathcal{S}(g_1, \ldots, g_s)$ *implies* $f \in \mathrm{QM}(g_1, \ldots, g_s)$.

2.1.2 Membership in Archimedean quadratic modules in $\mathbb{R}[X]$

The functions κ_a, κ_a^+, and κ_a^-, introduced in [4], simplify the membership check in Archimedean quadratic modules by keeping track of the minimal orders and sign conditions from the aforementioned lattice construction.

Definition 2.4 [[4, p. 43]] Let $G := \{g_1, \ldots, g_s\} \subseteq \mathbb{R}[X]$. We define:

$$\kappa_a(G) := \min_{1 \leq i \leq s} \{\mathrm{ord}_a(g_i) \mid \mathrm{ord}_a(g_i) \in 2\mathbb{N}, \epsilon_a(g_i) = -1\}$$

$$\kappa_a^+(G) := \min_{1 \leq i \leq s} \{\mathrm{ord}_a(g_i) \mid \mathrm{ord}_a(g_i) \in 2\mathbb{N}+1, \epsilon_a(g_i) = 1\}$$

$$\kappa_a^-(G) := \min_{1 \leq i \leq s} \{\mathrm{ord}_a(g_i) \mid \mathrm{ord}_a(g_i) \in 2\mathbb{N}+1, \epsilon_a(g_i) = -1\}$$

Each of these functions take the value ∞ if there is no polynomial g_i in G satisfying the conditions.

The following theorem allows us to test the membership of polynomials in Archimedean quadratic modules.

Theorem 2.5 ([4, p. 47]) *Let* $f \in \mathbb{R}[X]$ *and* $G = \{g_1, \ldots, g_s\} \subseteq \mathbb{R}[X]$ *with* $S = \mathcal{S}(G)$ *bounded. Then* $f \in \mathrm{QM}(G)$ *if and only if* $f|_S \geq 0$ *and*

1. *for every left boundary point* a *of* $S \setminus S_{isol}$, *we have* $\mathrm{ord}_a(f)$ *is even or* $\mathrm{ord}_a(f) - \kappa_a^+(G) \in 2\mathbb{N}_0$

2. *for every right boundary point a of $S \setminus S_{isol}$, we have* $\mathrm{ord}_a(f)$ *is even or* $\mathrm{ord}_a(f) - \kappa_a^-(G) \in 2\mathbb{N}_0$

3. *for every isolated point a of S, we have* $\mathrm{ord}_a(f)$ *is even and* $\epsilon_a(f) = 1$ *or*

   Case 1: $\mathrm{ord}_a(f) \geq \kappa_a(G)$ *if* $\kappa_a(G) < \kappa_a^+(G)$ *and* $\kappa_a(G) < \kappa_a^-(G)$

   Case 2: $(\mathrm{ord}_a(f) - \kappa_a^+(G) \in 2\mathbb{N}_0$ *and* $\epsilon_a(f) = 1)$ *or* $\mathrm{ord}_a(f) \geq \min\{\kappa_a(G), \kappa_a^-(G)\}$ *if* $\kappa_a^+(G) \leq \min\{\kappa_a(G), \kappa_a^-(G)\}$

   Case 3: $(\mathrm{ord}_a(f) - \kappa_a^-(G) \in 2\mathbb{N}_0$ *and* $\epsilon_a(f) = -1)$ *or* $\mathrm{ord}_a(f) \geq \min\{\kappa_a(G), \kappa_a^+(G)\}$ *if* $\kappa_a^-(G) \leq \min\{\kappa_a(G), \kappa_a^+(G)\}$

2.1.3 Natural generators associated with $\mathcal{S}(G)$

The saturation $\tilde{Q}$ of a quadratic module Q is defined as the smallest intersection of all quadratic modules, including Q. A quadratic module Q is saturated if $\tilde{Q} = Q$. The following theorem provides a semantic characterization:

Theorem 2.6 *[57, Proposition 2.6.1 p. 33] Let G be a finite subset of $\mathbb{R}[X]$ and $Q := \mathrm{QM}(G)$. Then*

1. $\tilde{Q} = \{f \in \mathbb{R}[X] \mid f \geq 0$ *over* $\mathcal{S}(G)\}$.

2. *Q is saturated if and only if for all $f \in \mathbb{R}[X]$, $f \geq 0$ over $\mathcal{S}(G)$ implies $f \in Q$.*

Kuhlmann and Marshall introduced the concept of natural generators associated with a compact semialgebraic set $\mathcal{S}(G)$; we adapt their terminology.

Definition 2.7 [41] For a given semialgebraic set $S \neq \emptyset$, the natural generators of S, denoted $\mathrm{Nat}(S)$, are the set of polynomials including:

(i) If $a \in S$ and $(-\infty, a) \cap S = \emptyset$, then $X - a \in \mathrm{Nat}(S)$. This is called the left natural generator or simply the **left generator** of $\mathrm{Nat}(S)$.

(ii) If $b, c \in S, b < c, (b, c) \cap S = \emptyset$, then $(X - b)(X - c) \in \mathrm{Nat}(S)$. This is called a **quadratic generator** of $\mathrm{Nat}(S)$.

(iii) If $d \in S$ and $(d, \infty) \cap S = \emptyset$, then $-(X - d) \in \mathrm{Nat}(S)$. This is called the right natural generator or simply the **right generator** of $\mathrm{Nat}(S)$.

(iv) $\mathrm{Nat}(S)$ has no other elements except these.

If $S = \emptyset$, then the set of natural generators is $\{-1\}$.

This theorem relates the concept of saturated quadratic modules and natural generators, we focus only on the compact case:

Theorem 2.8 *[42, Theorem 3.1, p. 590 item (b)] If $\mathcal{S}(G)$ is compact, the following are equivalent:*

- $\mathrm{QM}(G)$ *is saturated.*
- $\mathrm{QM}(G)$ *contains the natural generators of $\mathcal{S}(G)$.*

From the latter, we can see that if $\mathcal{S}(G)$ is empty, the quadratic module $\mathrm{QM}(G)$ is saturated. The following theorem from [42] gives a practical procedure to determine whether $QM(G)$ is saturated.

Theorem 2.9 *[42, Theorem 3.2]* $\mathrm{QM}(G)$ *is saturated if and only if $\mathcal{S}(G)$ satisfies the following two conditions:*

(i) for each left endpoint a_j in $\mathcal{S}(G)$, there exists $i \in \{1, \ldots, s\}$ such that $g_i(a_j) = 0$ and $\frac{dg_i}{dX}(a_j) > 0$,

(ii) for each right endpoint b_j in $\mathcal{S}(G)$, there exists $i \in \{1, \ldots, s\}$ such that $g_i(b_j) = 0$ and $\frac{dg_i}{dX}(b_j) < 0$.

A method for generating a certificate of f in $\mathrm{PO}(\mathrm{Nat}(S))$ is extracted from the inductive argument about the degree of f in [4]; it involves computing certificates of factors of f that are nonnegative over S. Since preorderings are closed by multiplication, the certificate of f is obtained by multiplying and rearranging the certificates of each of its factors appropriately in the preordering structure.

Example 2.10 Consider $G = \{g_1\}$ where $g_1 = -(X+1)(X-1)$. We will prove $\mathrm{QM}(G)$ to be saturated using Theorem 2.9. The semialgebraic set of G is $[-1, 1]$, so we need to check the endpoints -1 and 1:

- For the endpoint -1: the generator g_1 evaluates to 0 at -1 and $\frac{dg_1}{dX}(-1) = -2(-1) > 0$.
- For the endpoint 1: the generator g_1 evaluates to 0 at 1 and $\frac{dg_1}{dX}(1) = -2(1) < 0$.

Hence, $\mathrm{QM}(G)$ as we fulfill the conditions of Theorem 2.9. In fact, $\mathrm{QM}(G)$ contain the natural generator $X+1$ and $-X+1$ since $\pm X+1 = \frac{1}{2}(X \pm 1)^2 + \frac{1}{2} \cdot g_1 \in \mathrm{QM}(G)$.

Example 2.11 Consider $G = \{g_2\}$ where $g_1 = -(X+1)^3(X-1)^3$. We will prove $\mathrm{QM}(G)$ to be not saturated using Theorem 2.9. The semialgebraic set of G is $[-1, 1]$, so we need to check the endpoints -1 and 1:

- For the endpoint -1: the generator g_2 evaluates to 0 at -1 but $\frac{dg_2}{dX}(-1) = -6(-1)((-1)^2 - 1)^2 = 0$.
- For the endpoint 1: the generator g_2 evaluates to 0 at 1 but $\frac{dg_2}{dX}(1) - 6(1)((1)^2 - 1)^2 = 0$.

Hence, $\mathrm{QM}(G)$ is not saturated.

Theorem 2.12 *[4, Theorem 1.6] For a basic closed semialgebraic set $S \subseteq \mathbb{R}$ we have* $\mathrm{Pos}(S) = \mathrm{PO}(\mathrm{Nat}(S))$.

The certificates in Theorem 2.12 are obtained using an inductive proof on the degree of the given polynomial. The base case is when the polynomial is nonnegative [2] over $\mathbb{R}$; the inductive case is when the given polynomial evaluates to a negative value, say $f(c) < 0$ for some $c \in \mathbb{R}$, and $\deg(f) > 0$. This is divided into three cases:

- Case 1: If $c \leq a_0$, then there exists c_0 such that $c < c_0 \leq a$ such that $f(c_0) = 0$. Hence, $f = (X - c_0)g$ for some g that is nonnegative over S.
- Case 2: If $c \geq b_k$, then there exists c_0 such that $b_k \leq c_0 < c$ such that $f(c_0) = 0$. Hence, $f = (-(X - c_0))g$ for some g that is nonnegative over S.
- Case 3: If $b_i \leq c \leq a_{i+1}$, then there exists c_0, c_1 such that $b_i \leq c_0 \leq c_1 \leq a_{i+1}$ and $f(c_0) = f(c_1) = 0$. Hence, $f = (X - c_0)(X - c_1)g$ for some g that is nonnegative over S.

For Case 1 (resp. Case 2), we can compute a certificate in $\mathrm{PO}(\mathrm{Nat}(S))$ of the factor $X - c_0$ (resp. $-(X - c_0)$) since $X - c_0 = (a_0 - c_0) + 1 \cdot (X - a_0)$ (resp. $-(X - c_0) = (c_0 - b_k) + 1 \cdot (-(X - b_k))$). For Case 3, the construction relies on Lemma 2.21.

2.1.4 Signatures of quadratic modules in $\mathbb{A}[X]$

The authors in [81] introduced the concept of *signature* of a quadratic module in order to generalize the construction of "complete vector of orders" of generalized

[2] This is because, in the univariate case, nonnegative polynomials correspond to sums of squares polynomials which are members of any quadratic module.

natural generators [4] and simplify the algorithm description of the membership test of polynomial in Archimedean quadratic modules in one variable. We recall the definition as we use the signature in our algorithms:

Definition 2.13 [81] Given a finite polynomial set $G \subseteq \mathbb{R}[X]$ with $\mathcal{S}(G)$ bounded, let $\mathcal{S}(G) = \bigcup_{i=0}^{k} [a_i, b_i]$ be such that $b_i < a_{i+1}$ for $i = 0, \ldots, k-1$. For any finite polynomial set $F \subseteq \mathbb{R}[X]$ such that $\mathcal{S}(G) \subseteq \mathcal{S}(F)$, we assign a tuple $(\kappa_i, \kappa_i^+, \kappa_i^-)$ to each interval $[a_i, b_i]$ for $i = 0, \ldots, k$ as follows:

1. If $a_i < b_i$, set

   $$\kappa_i := \infty, \kappa_i^+ := \kappa_{a_i}^+(F), \kappa_i^- := \kappa_{b_i}^-(F)$$

2. (Type A) If $\kappa_a(F) < \kappa_a^+(F)$ and $\kappa_a(F) < \kappa_a^-(F)$, set

   $$\kappa_i := \kappa_a(F), \kappa_i^+ := \kappa_a(F) + 1, \kappa_i^- := \kappa_a(F) + 1$$

3. (Type B) If $\kappa_a(F) > \kappa_a^+(F)$ and $\kappa_a(F) > \kappa_a^-(F)$, set

   $$\kappa_i := \max(\kappa_a^+(F), \kappa_a^-(F)) + 1, \kappa_i^+ := \kappa_a^+(F), \kappa_i^- := \kappa_a^-(F)$$

4. (Type C) If $\kappa_a^+(F) < \kappa_a(F) < \kappa_a^-(F)$, set

   $$\kappa_i := \kappa_a(F), \kappa_i^+ := \kappa_a^+(F), \kappa_i^- := \kappa_a(F) + 1$$

5. (Type D) If $\kappa_a^-(F) < \kappa_a(F) < \kappa_a^+(F)$, set

   $$\kappa_i := \kappa_a(F), \kappa_i^+ := \kappa_a(F) + 1, \kappa_i^- := \kappa_a^-(F)$$

6. (Type E) If $\kappa_a(F) = \kappa_a^+(F) = \infty$, set

   $$\kappa_i := \kappa_i^+ := \infty, \kappa_i^- := \kappa_a^-(F)$$

   or if $\kappa_a^-(F) := \kappa_a(F) := \infty$, set

   $$\kappa_i := \kappa_i^- := \infty, \kappa_i^+ := \kappa_a^+(F)$$

The *signature* of F, denoted by $\omega_S(F)$, is defined as the sequence of the tuples above for each of the intervals in S, i.e.,

$$\omega_S(F) := (\kappa_0, \kappa_0^+, \kappa_0^-, \ldots, \kappa_k, \kappa_k^+, \kappa_k^-)$$

When $S = \mathcal{S}(F)$, the subscript S in $\omega_S(F)$ is omitted.

2.1.5 Monogenic quadratic modules in $\mathbb{A}[X]$

This section discusses a simplification of Theorem 2.5 that allows us to test the membership of polynomials in Archimedean monogenic quadratic modules. Additionally, we discuss a characterization theorem of [4] of Archimedean monogenic quadratic modules. Since monogenic quadratic modules are generated by a single polynomial, there is a single type of isolated points that can occur in the semialgebraic set associated with these quadratic modules as there cannot be more than one generator involved Definition 2.13.

Definition 2.14 [[4, p. 48]]

Let $G = \{g_1, \ldots, g_s\} \subseteq \mathbb{R}[X], S = \mathcal{S}(G)$ and $a \in S_{isol}$. We say that a is an isolated point of type A if

$$\kappa_a(G) < \kappa_a^+(G) \text{ and } \kappa_a(G) < \kappa_a^-(G) \tag{2.1}$$

Theorem 2.15 *[4] Let $f, g \in \mathbb{R}[X]$ and $S = \mathcal{S}(g) \subseteq \mathbb{R}$ be bounded. Then $f \in \mathrm{QM}(g)$ if and only if $f \geq 0$ over $\mathcal{S}(g)$ and*

1. *for every boundary point of $S \setminus S_{isol}$ we have that $\mathrm{ord}_a(f)$ is even and $\epsilon_a(f) = 1$ or $\mathrm{ord}_a(f) - \mathrm{ord}_a(g) \in 2\mathbb{N}$ and $\epsilon_a(f) = \epsilon_a(g)$.*

2. *for every isolated point a of S we have that* $\mathrm{ord}_a(f)$ *is even and* $\epsilon_a(f) = 1$ *or* $\mathrm{ord}_a(f) \geq \mathrm{ord}_a(g)$.

Augustin provides in [4] the following Corollary to identify Archimedean monogenic quadratic modules $\mathrm{QM}(G)$ by checking if all isolated points of $\mathcal{S}(G)$ are of type A.

Theorem 2.16 (Corollary 2.29[4, p. 58]) *Let $G \subseteq \mathbb{R}[X]$ be finite and $Q = \mathrm{QM}(G)$ such that $\mathcal{S}(G) = \bigcup_{i=0}^{k} [a_i, b_i]$. Then the following are equivalent:*

- *Q is generated by one polynomial*
- *Every isolated point of $\mathcal{S}(G)$ is of type A*

If one of the equivalent conditions is satisfied, then

$$Q = \mathrm{QM}\left(- \prod_{\substack{i=0 \\ a_i<b_i}}^{m} (X-a_i)^{\kappa^+_{a_i}(G)} (X-b_i)^{\kappa^-_{b_i}(G)} \prod_{\substack{i=0 \\ a_i=b_i}}^{m} (X-a_i)^{\kappa_{a_i}(G)} \right) \tag{2.2}$$

This corollary tells us the form of the single generator of the quadratic module using information from the interval representation of the semialgebraic set of its generators.

Since the set of generators for the quadratic modules considered is fixed, we will consider the notation $\kappa_{a_i}, \kappa^+_{a_i}, \kappa^-_{a_i}$ to refer to $\kappa_{a_i}(G), \kappa^+_{a_i}(G), \kappa^-_{a_i}(G)$, respectively. Unless stated otherwise, we will use g to denote the single generator associated with such quadratic module of Theorem 2.16 and f to denote the input polynomial.

We focus on finding certificates of membership of f belonging to some monogenic quadratic module and denote s_0, s_1 for the sums of squares part and sums of squares multiplier of g, respectively, in a representation of f.

2.1.6 Gram matrix method and sums of squares decomposition

The connection between sums of squares decompositions and positive semidefinite matrices is due to the following proposition.

Proposition 2.17 *[69] Suppose $f \in \mathbb{R}[\overline{X}]$ has degree $2d$, let $N = \binom{n-1+d}{d}$ and let V be the $N \times 1$ vector of all monomials in $\mathbb{R}[\overline{X}]$ of degree at most d. Then f is a sums of squares if and only if there exists an $N \times N$ symmetric positive semidefinite matrix A such that $f = V^T A V$.*

Example 2.18 Consider $f = X^2 + Y^2 - 4XYZ + X^2Z^2 + Y^2Z^2$. We will prove that f is a sum of squares. Consider $A = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$ and $V = \begin{pmatrix} X \\ XZ \\ Y \\ YZ \end{pmatrix}$. We can verify $f = V^T A V$. To obtain a sums of squares decomposition we factorize $A = B^T B$ where $B = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$. The sums of squares decomposition of f is obtained by computing $\langle BV, BV \rangle$, which gives $(X - YZ)^2 + (XZ - Y)^2$.

In [68], the authors introduce a methodology for computing sums of squares decompositions (SOS) using SDP solvers and projection to rational solutions. The advantages of this approach are the efficiency of the SDP solvers while the rational projection allows them to integrate their solution for reasoning tasks. This algorithm and variations of it have been implemented in several software packages such as `SOSTOOLS` [65, 66] and `SumsOfSquares` [25, 26].

SDP solvers are efficient in practice. However, their dependence on numerical methods (interior point algorithms) can lead to precision errors that allow only approximate solutions. A line of research focuses on fixing the SDP solver inaccuracies to provide exact certificates, particularly for sums of squares decompositions. In [68], the authors compute sums of squares decompositions with rational coefficients using a rounding and projection technique by rationalizing the coefficients of the obtained representation and minimizing a least squares error between the coefficients of the original polynomial and the rationalized approximate representation. This procedure depends on geometrical conditions related to the distance between the obtained representation and the space of feasible sums of squares representations. In [36], the authors proposed another method to find exact certificates of sums of squares decompositions by including, in addition to the least squares projection, a "refinement" step involving the Gauss-Newton iteration to find a feasible solution of the SDP which is strictly contained in the interior of the affine space of positive semidefinite matrices.

2.1.7 The Basic Lemma

The authors in [42] provide constructive proofs of the results in this section. We use them in our constructions, so we adapt some terminology to fit our context in quadratic modules in the univariate case:

Theorem 2.19 *[42, Basic Lemma (Lemma 2.1)] Suppose $f, g \in \mathrm{Pos}(\mathcal{S}(G))$ are relatively prime. There exist strictly positive polynomials σ, τ over $\mathcal{S}(G)$ such that $1 = \sigma f + \tau g$.*

This theorem is proved constructively by explicitly computing both σ and τ. As an observation, by Schmüdgen's Positivstellensatz , both σ and τ belong to the preordering $\mathrm{PO}(G)$. Under additional conditions, τ and σ become sums of squares.

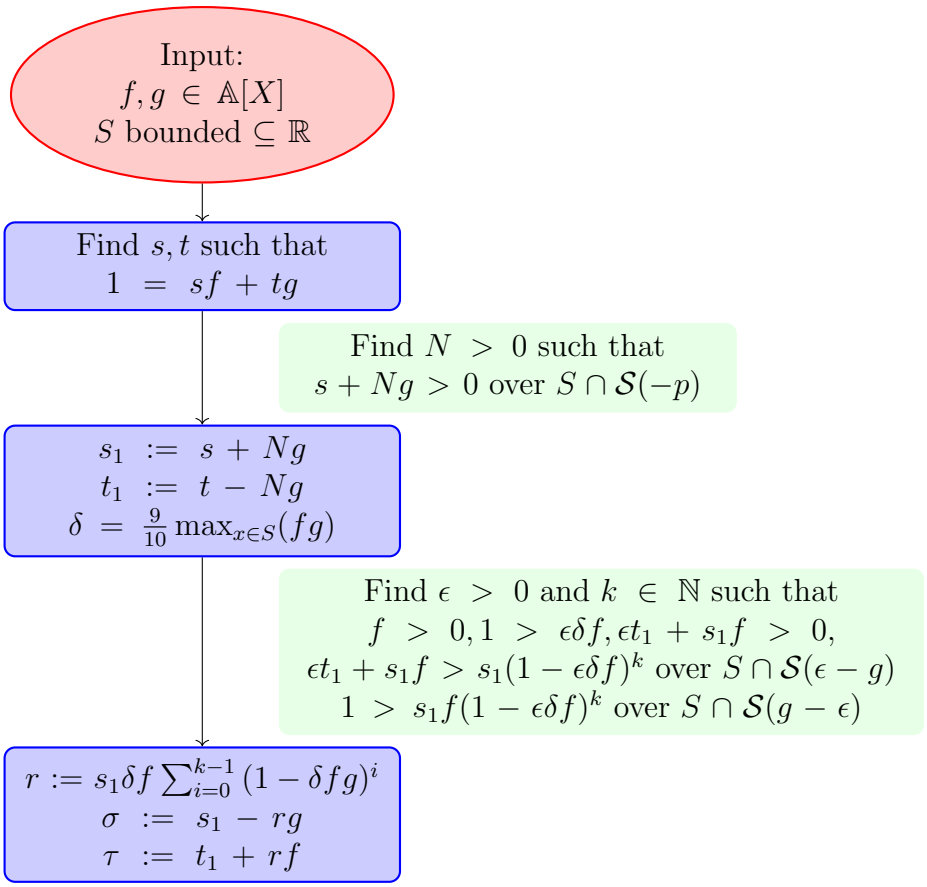

Figure 2.1: High-level description of the Basic Lemma construction

Theorem 2.20 *[42, Corollary 2.3 (i)] Given $f, g \in \mathrm{Pos}(\mathcal{S}(fg))$ being relatively prime, where $\mathcal{S}(fg)$ is compact, there exists $\sigma, \tau \in \sum \mathbb{R}[X]^2$ such that $1 = \sigma f + \tau g$.*

This theorem is used to obtain certificates in quadratic modules of the factors of members satisfying the above conditions. The fact that σ, τ can be made to be sums of squares, allows us to avoid products in the preordering structure. Similarly to this result, there is another constructive approach in [76, Proposition 4.8] using Weierstrass-approximation that computes σ, τ as sums of squares under equivalent conditions. In Section 2.1.7 we discuss the BasicLemma algorithm (resp. SOSBasicLemma) to compute σ, τ in Theorem 2.19 (resp. Theorem 2.20.)

The rest of this section provides an algorithmic description of the Basic Lemma. We have implemented this algorithm in a `Maple` package [3]. The Algorithm 4 describes a procedure for effectively constructing certificates for Theorem 2.20.

Algorithm 1: Algorithm to find ϵ in Basic Lemma

FindEps$(f, g, s_1, t_1, S, \delta)$

Input: $f, g, s_1, t_1 \in \mathbb{A}[X]$, $S \subseteq \mathbb{R}$, $\delta \in \mathbb{Q}$

Output: $\epsilon_{curr} \in \mathbb{Q}$

Requires: S is a non-empty finite union of intervals with algebraic numbers as end points.

`/* Find positive rational` ϵ `so that` $f > 0, 1 > \epsilon\delta f$, `and` $\epsilon t_1 + s_1 f > 0$ `over` $L_2 = \{x \in S \mid g(x) \leq \epsilon\}$ `*/`

$\epsilon_{top} := \lceil \max_{x \in S}(g) \rceil$

$\epsilon_{curr} := \frac{\epsilon_{top}}{2}$

while *true* **do**

  $cond_1 := isEmpty(S \cap \mathcal{S}(\epsilon_{curr} - g) \cap \mathcal{S}(-f))$

  $cond_2 := isEmpty(S \cap \mathcal{S}(\epsilon_{curr} - g) \cap \mathcal{S}(\epsilon_{curr}\delta f - 1))$

  $cond_3 := isEmpty(S \cap \mathcal{S}(\epsilon_{curr} - g) \cap \mathcal{S}(-(\epsilon_{curr} t_1 + s_1 f)))$

  if $cond_1$ *and* $cond_2$ *and* $cond_3$ **then**

    return ϵ_{curr}

  $\epsilon_{curr} = \frac{\epsilon_{curr}}{2}$

[3] This package is freely available on the following GitHub repository: https://github.com/typesAreSpaces/BasicLemmaStrictlyPositiveCert

Algorithm 2: Algorithm to find k in Basic Lemma

$\mathsf{FindK}(f, g, s_1, t_1, S, \delta)$

Input: $f, g, s_1, t_1 \in \mathbb{A}[X]$, $S \subseteq \mathbb{R}$, $\delta \in \mathbb{Q}$

Output: $k \in \mathbb{N}$

Requires: S is a non-empty finite union of intervals with algebraic numbers as end points.

$\epsilon := \mathsf{FindEps}(f, g, s_1, t_1, S, \delta)$

$L_2 := S \cap \mathcal{S}(\epsilon - g)$

$L_3 := S \cap \mathcal{S}(g - \epsilon)$

if L_2 *is non-empty* **then**

  while *true* **do**

    $x^* := \arg\max_{x \in L_2}(s_1 f(1 - \epsilon\delta f)^k)$

    $value := s_1 f(x^*)(1 - \epsilon\delta f(x^*))^k$

    $test := \epsilon t_1(x^*) + s_1 f(x^*)$

    if $value < test$ **then**

      break

    else

      $k := \left\lceil \frac{\log(\frac{test}{s_1(x^*)f(x^*)})}{\log(1-\epsilon\delta f(x^*))} \right\rceil$

if L_3 *is non-empty* **then**

  while *true* **do**

    $x^* := \arg\max_{x \in L_3}(s_1 f(1 - \epsilon\delta f)^k)$

    $value := s_1 f(x^*)(1 - \epsilon\delta f(x^*))^k$

    if $value < 1$ **then**

      break

    else

      $k := \left\lceil \frac{\log(\frac{1}{(s_1(x^*)f(x^*))})}{\log(1-\epsilon\delta f(x^*))} \right\rceil$

return k

Algorithm 3: Implementation of Basic Lemma

BasicLemma(f, g, G)

Input: $f, g \in \mathbb{A}[X]$, $G \subseteq \mathbb{A}[X]$

Output: $\sigma, \tau \in \mathbb{A}[X]$

Requires: $\mathcal{S}(G)$ is a union of bounded intervals $\bigcup_{i=0}^{k} [a_i, b_i]$; f, g are nonnegative over $\mathcal{S}(G)$; f, g are relatively prime.

Ensures: σ, τ are strictly positive over $\mathcal{S}(G)$ and $1 = \sigma f + \tau g$

1 $S := \mathcal{S}(G)$

2 Find $s, t \in \mathbb{A}[X]$ such that $1 = sf + tg$ using the extended Euclidean algorithm

3 **if** S *is empty* **then**

4 **return** s, t

5 $L_1 := S \cap \mathcal{S}(-s)$

/* Find N such that $s + Ng > 0$ over L_1 */

6 **if** $isEmpty(L_1)$ **then**

7 $N := 0$

8 **else**

// The polynomial g is positive over L_1

9 $N := (1 + \frac{1}{100}) \left\lceil \max_{x \in L_1}(-\frac{s}{g}) \right\rceil$

10 $s_1 := s + Ng$

11 $t_1 := t - Nf$

/* Find positive rational δ such that $\delta fg < 1$ over S */

12 $\delta := \frac{1}{\lceil \max_{x \in S}(fg) \rceil}$

13 $k := \mathsf{FindK}(f, g, s_1, t_1, S, \delta)$

14 $r := s_1 \delta f \sum_{i=0}^{k-1} (1 - \delta fg)^i$

15 $\sigma := s_1 - rg$

16 $\tau := t_1 + rf$

17 **return** σ, τ

Algorithm 4: Implementation of SOS variant of Basic Lemma

SOSBasicLemma(f, g)

Input: $f, g \in \mathbb{A}[X]$

Output: $\sigma, \tau \in \mathbb{A}[X]$

Requires: f, g are relatively prime; f, g are nonnegative over $\mathcal{S}(fg)$; $\mathcal{S}(fg)$ is bounded

Ensures: σ, τ are sums of squares and $1 = \sigma f + \tau g$

```
σ1, τ1 := BasicLemma(f, g, {fg})
if σ1 is a sums of squares then
    σ1,0, σ1,1 := σ1, 0
else
    σ1,0, σ1,1 := Certificate({fg}, σ1)
if τ1 is a sums of squares then
    τ1,0, τ1,1 := τ1, 0
else
    τ1,0, τ1,1 := Certificate({fg}, τ1)
σ := σ1,0 + τ1,1 g^2
τ := τ1,0 + σ1,1 f^2
return σ, τ
```

2.1.8 Berg's Lemma

This lemma provides a constructive proof for computing a certificate for a parabola in the monogenic quadratic module generated by another parabola containing the endpoints of the input polynomial. This construction is used in the saturated case (Theorem 2.12, Section 4.1) to compute a certificate of nonnegative factors of degree two using the preordering of natural generators associated with the semialgebraic set of the original generators and in different constructions of Appendix B.

Lemma 2.21 *[4, Lemma 1.5] If $a, b, c, d \in \mathbb{R}$ with $a \leq c \leq d \leq b$ then*

$$(X - c)(X - d) \in \mathrm{QM}((X - a)(X - b)) \tag{2.3}$$

Moreover, there exists a nonnegative number $\gamma \in \mathbb{R}$ depending on a, c, d, b such that $(X - c)(X - d) = \sigma_0 + \gamma \cdot (X - a)(X - b)$ where $\sigma_0 \in \sum \mathbb{R}[X]^2$.

It should be noted that the proof of Lemma 2.21 is constructive and finds an arithmetical expression[4] for γ. Hence, if the parameters a, b, c, d are computable, γ above is computable as well.

The following algorithmic description of Berg's lemma allows us to compute certificates. The constructive results of the proof are taken from [4].

Algorithm 5: Certificates of quadratic polynomials in Berg's Lemma

Input: $a, b, c, d \in \mathbb{A}$

Output: $\sigma_0, \sigma_1 \in \mathbb{A}[X]$

Requires: $a \leq c \leq d \leq b$

Ensures: $(X - c)(X - d) = \sigma_0 + \sigma_1 \cdot (X - a)(X - b)$ and $\sigma_0, \sigma_1 \in \sum \mathbb{A}[X]^2$

1 **if** $a + b < c + d$ **then**
2 $\quad \sigma_1 := \frac{2}{b-a}(b - \frac{c+d}{2})$
3 **if** $a + b > c + d$ **then**
4 $\quad \sigma_1 := \frac{2}{b-a}(\frac{c+d}{2} - a)$
5 **if** $a + b = c + d$ **then**
6 $\quad \sigma_1 := \frac{(d-c)^2}{(b-a)^2}$
7 $\sigma_0 := (X - c)(X - d) - \sigma_1 \cdot (X - a)(X - b)$
8 **return** σ_0, σ_1

[4]The set of arithmetical expressions is a set closed by the arithmetic operators (addition, subtraction, multiplication, division, and exponentiation.)

2.1.9 Certificates of strictly positive polynomials over Archimedean quadratic modules

The Certificate algorithm proposed in [82] computes a certificate for strictly positive polynomials in Archimedean quadratic module. We used Certificate whenever certificates are needed for such polynomials. It is based on a modification and extension of the following result.

Lemma 2.22 *[6, Lemma 7 p. 4] Let $G = \{g_1, \ldots, g_s\}$, $S := \mathcal{S}(G)$, and let $f \in \mathbb{A}[X]$ be strictly positive on S. Let B be a compact subset of $\mathbb{R}^d$. Then there exists $g \in \mathrm{QM}(G)$ such that $f - g$ is strictly positive on B.*

The algorithm for the univariate case combines two ideas: first, it computes a polynomial g in $\mathrm{QM}(G)$ in order to avoid the inclusion of a polynomial of the form $N - X^2$ in the set of generators; second, it uses a constructive result of Lemma 7 in [6] (denoted as the Averkov algorithm) and a generalization result (denoted as the ExtAverkov algorithm) of the latter, which produces a polynomial of the form $f - \tilde{g}$ that is strictly positive over $\mathbb{R}$ where $\tilde{g} \in \mathrm{QM}(G)$. The latter is a sums of squares in the univariate case. We state the generalization of the Averkov's lemma used in this algorithm.

Theorem 2.23 *[82, Proposition 5.1] Let $f, g \in \mathbb{A}[X]$ be two polynomials with $\mathcal{S}(g)$ bounded and $f > 0$ on $\mathcal{S}(g)$. Then there exists a sum of squares δ such that $f - \delta \cdot g$ is strictly positive on $\mathcal{S}(g)$ and has a lower bound over $\mathbb{R}$.*

We implemented a `Maple` package [5] of the Certificate algorithm for the univariate case. We use the Certificate algorithm from [82] to compute certificates of strictly

[5]The package is freely available in the following GitHub repository: https://github.com/typesAreSpaces/StrictlyPositiveCert

positive polynomials over bounded semialgebraic sets. This algorithm does not introduce new constants into the base field $\mathbb{A}$, as intermediate polynomials and parameters use rational numbers. We describe a specialized version of the Certificate algorithm for the univariate case as follows:

Algorithm 6: Algorithm for solving the certificate problem for strictly positive polynomials

$\mathsf{Certificate}(G, f)$

Input: A polynomial set $G = \{g_1, \ldots, g_s\} \subseteq \mathbb{A}[X]$ and a polynomial $f \in \mathbb{A}[X]$ such that $f > 0$ over $\mathcal{S}(G)$

Output: A sequence $(\sigma_0, \sigma_1, \ldots, \sigma_s)$ of sum-of-squares multipliers such that $f = \sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i$

for $i = 1, \ldots, s$ **do**
  $\sigma_i := 0$; $c_i := 0$;
if $\exists g_i \in G$ *such that* $\mathcal{S}(g_i)$ *is bounded* **then**
  $g := g_i; c_i := 1$;
else
  Find g_i and g_j of odd degrees and leading coefficients of opposite signs such that $\deg(g_i) \geq \deg(g_j)$;
  if $\deg(g_i) > \deg(g_j)$ **then**
    $c_i := \frac{1}{|\operatorname{lc}(g_i)|}$; $c_j := \frac{(X-a)^{\deg(g_i)-\deg(g_j)}}{|\operatorname{lc}(g_j)|}$;
  if $\deg(g_i) = \deg(g_j)$ **then**
    $c_i := \frac{X^2}{|\operatorname{lc}(g_i)|}$; $c_j := \frac{(X-a)^2}{|\operatorname{lc}(g_j)|}$;
  $g := c_i \cdot g_i + c_j \cdot g_j$ choosing a value for a such that $\operatorname{lc}(g) < 0$;
$(\sigma_1, \ldots, \sigma_s) := \mathsf{Averkov}(G, f, \mathcal{S}(g))$;
$\tilde{f} := f - \sum_{i=1}^{s} \sigma_i g_i$;
if $\tilde{f}$ *does not have a global lower bound* **then**
  Let $\delta := cX^{2\lfloor \frac{m}{2} \rfloor + 2}$; $\tilde{f} := \tilde{f} - \delta \cdot g$ where $m = \deg(f) - 1$ and $c < \frac{1}{2} \min_{x \in \mathcal{S}(g)} \frac{f(x)}{X^{2\lfloor \frac{m}{2} \rfloor + 2} g(x)}$;
$p := \mathsf{ExtAverkov}(g, \tilde{f}, \mathbb{R}^n)$;
$\sigma_i := \sigma_i + (p + \delta)c_i$; $\sigma_j := \sigma_j + (p + \delta)c_j$;
$\sigma_0 := f - \sum_{i=1}^{s} \sigma_i \cdot g_i$;
return $(\sigma_0, \sigma_1, \ldots, \sigma_s)$;

We illustrate the steps of Certificate using the following examples; we compute a certificate of an Archimedean polynomial for monogenic quadratic modules.

Example 2.24 Let us consider the generator $g_1 = -X^4 - X^2Y^2 + 2X^2Y + XY^2 - Y^2$ and let $a_1 = 3 - X^2 - Y^2$. Since a_1 is strictly positive over $\mathcal{S}(g_1)$, we want to compute a certificate for the Archimedean polynomial $a_1 \in \mathrm{QM}(g_1)$.

1. First, we compute a sums of squares multiplier $\delta_1 = \frac{1}{65}(X^2+Y^2)^4$ in Line 15 of the Certificate algorithm because a_1 does not have a global lower bound. Then, we set $\hat{a_1} := a_1 - \delta_1 \cdot g_1$.
2. Then, we compute the parameters γ, ϵ and N. We obtain $\gamma = \epsilon = \frac{1}{25}$ and $N = 0$. We verify that $p_1 := \hat{a_1} - (\frac{g_1 - \gamma}{\gamma + \epsilon})^{2N} \cdot g_1$ is nonnegative.
3. We use the Gram matrix method (Proposition 2.17) to check if p_1 is a sums of squares. Since it is, we are done with this step.
4. Finally, the certificate obtained for a_1 is $a_1 = p_1 + (\delta_1 + (\frac{g_1 - \gamma}{\gamma + \epsilon})^{2N}) \cdot g_1$.

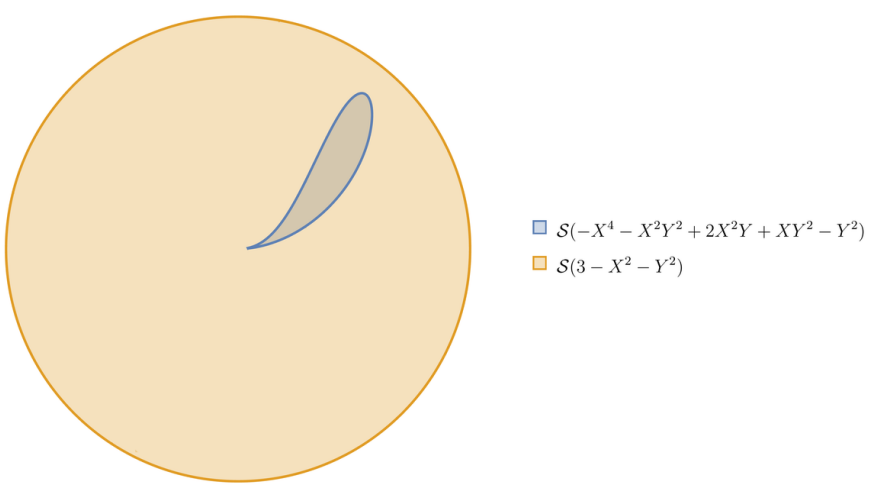

Figure 2.2: Semialgebraic set of $g_1 = -X^4 - X^2Y^2 + 2X^2Y + XY^2 - Y^2$

Example 2.25 Consider the generator $g_2 = 1 + XY - X^2 - Y^2 - Z^2$ and let $a_2 = 3 - X^2 - Y^2 - Z^2$. Since a_2 is strictly positive over $\mathcal{S}(g_2)$, we want to compute a certificate for the Archimedean polynomial $a_2 \in \mathrm{QM}(g_2)$.

1. First, we compute a sums of squares multiplier $\delta_2 = (X^2 + Y^2 + Z^2)^4$ in Line 15 of the Certificate algorithm as a_2 does not have a global lower bound. Then, we set $\hat{a_2} := a_2 - \delta_2 \cdot g_2$.

2. Then, we compute the parameters γ, ϵ and N. We obtain $\gamma = 1, \epsilon = \frac{1}{8}$ and $N = 3$. We verify that $p_2 := \hat{a_2} - (\frac{g_2 - \gamma}{\gamma + \epsilon})^{2N} \cdot g_2$ is nonnegative.

3. Since p_2 is a sums of squares using the Gram matrix check, we are done. Hence, the certificate obtained for a_2 is $a_2 = p_2 + (\delta_2 + (\frac{g_2 - \gamma}{\gamma + \epsilon})^{2N}) \cdot g_2$.

2.1.10 Extended Lasserre algorithm

The authors in [82] provide an algorithm for computing a certificate for a given strictly positive polynomial f over Archimedean quadratic modules $\mathrm{QM}(G)$. This procedure modifies a result from [44] that approximates nonnegative polynomials as sums of squares; the goal of the extended result is to lift a certificate of a nonnegative polynomial as a member of a monogenic quadratic module.

Theorem 2.26 *Let $f \in \mathbb{R}[\overline{X}]$ be a positive polynomial with a global minimum $f^* = \inf_{\bar{x} \in \mathbb{R}^n} f(\bar{x}) > 0$ and $g \in \mathbb{R}[\overline{X}]$ with $\mathcal{S}(g)$ bounded. Then there exist $\epsilon > 0$ and $r \in \mathbb{N}$ such that*

$$f_\epsilon = f - \epsilon \sum_{k=0}^{r} \sum_{j=1}^{n} \frac{X_j^{2k}}{k!} \cdot g \tag{2.4}$$

is a sum of squares.

We implemented a `Julia` package [6] of the ExtLasserre algorithm for the multivariate case. This algorithm follows a symbolic-numeric, as it uses a SDP solver to check whenever the resulting approximation is a sums of squares using the Gram matrix method. We use ExtLasserre in Chapter 7 to handle nonnegative polynomials in the algorithm to compute the certificate of -1 when $\mathcal{S}(g)$ is empty.

2.2 Related work

Real algebraic geometry is a branch of mathematics that is concerned with studying real solutions of polynomial inequalities known as semialgebraic sets. Recent developments in optimization methods [17, 36, 67] and hybrid numeric-symbolic algorithms [52, 54, 56] have enabled researchers to obtain either explicit certificates (existence) or infeasibility refutations (non-existence) of certain geometric properties which decide problems in the language of real closed fields; to name a few we have computing the sums of squares decomposition of nonnegative polynomials over the reals [36, 68, 71], find degree bounds of representations in several variations of the Positivstellensatz theorem including Hilbert's [49], Pòlya's [79], Schmüdgen's [46] and Putinar's [10].

The following lines of research relate to the problems addressed in the thesis. We divide them by symbolic and hybrid symbolic-numeric approaches. The first category relies on theoretical methods including constructive proofs in real algebraic geometry, which provide solutions with prohibitively complexity, triangular decomposition, which presents simpler representations for semialgebraic sets, and algorithms to decide the membership of polynomials in quadratic modules using valuation theory. The hybrid symbolic-numeric approaches rely on semidefinite programming methods and projection/rounding methods to provide rational representations.

[6] The package is freely available in the following GitHub repository: https://github.com/typesAreSpaces/globalStrictPositive

2.2.1 Constructive real algebraic geometry

The effective positvstellesätz problem asks for degree bounds for the sums of squares multipliers for some representation of a member of a quadratic module. Given this kind of information, it is, in principle, possible to compute certificates by setting template polynomials with fixed degree bounded by the aforementioned degrees.

A system of sign conditions, $[\mathcal{F}_{\neq}, \mathcal{F}_{\geq}, \mathcal{F}_{=}]$, is a triplet of sets of polynomials that encodes a conjunction of disequalities, inequalities, and equalities. A system of sign conditions is said to be **incompatible** if there exists an equation $N + S + Z = 0$ where N is a monoid from $\mathcal{F}_{\neq}$, S an element of a preordering generated by elements in $\mathcal{F}_{\geq}$, and Z is an element of the ideal generated by elements in $\mathcal{F}_{=}$. By the Positivstellensatz , a system of sign conditions is inconsistent if and only if the system of sign conditions is incompatible. In [49], the authors developed a technique using **weak inferences** and **weak existence** for which they extract algebraic identities witnessing the validity of the derivations in an inconsistency proof of a system of sign conditions. Various applications to the Positivstellensatz in this setting lead to the finding of certificates of nonnegativity over semialgebraic sets. In addition, the authors provide elementary recursive bounds [7] for the degree of the incompatibility polynomial, improving previous work providing primitive recursive bounds for the latter [48]. Although this line of research provided an improvement to the previous aforementioned complexity bounds, the approach shows a method to find certificates with polynomial degrees bounded by $2^{2^{\left(2^{\max\{2,d\}^{4^k}+s^{2^k}}\max\{2,d\}^{16^k bit(d)}\right)}}$ where d is a bound on the degrees of the polynomial system, and k is the number of variables of the input polynomial.

[7] The class of elementary recursive functions contains the addition, multiplication, difference, quotient, and projection functions over integers and are closed under projections, finite sums, and finite products

2.2.2 Complexity degree bounds for effective Positivstellensatz

A line of research involving the representation of elements in preorderings focuses only on finding bounds for the degree of the sums of squares multipliers. In [80], the author discusses bounds involving a norm in the real vector space $\mathbb{R}[X_1, \ldots, X_n]$ of polynomials for the degree of the term $\sigma_e \cdot g_1^{e_1} \cdots g_s^{e_s}$ in some representation of a strictly positive polynomial over the semialgebraic set of $\{g_1, \ldots, g_s\}$. His approach relies on using Pòlya forms and bounds from [70], guaranteeing conditions for such forms to be strictly positive over some simplex set. The degrees of the multipliers and generators $\sigma_e g_1^{e_1} \ldots g_s^{e_s}$ are bounded as follows: for a non-empty semialgebraic set $S := \mathcal{S}(g_1, \ldots, g_s) \subseteq (-1, 1)^n$, there is some $c \in \mathbb{N}$ such that for every $f \in \mathbb{R}[X_1, \ldots, X_n]$ of degree d with $f^* := \min\{f(x) \mid x \in S\} > 0$ we have

$$\deg(\sigma_e \cdot g_1^{e_1} \ldots g_s^{e_s}) \leq cd^2 \left(1 + \left(d^2 n^d \frac{||f||}{f^*}\right)^c\right)$$

where $||f||$ denotes the maximum norm of a f on $[-1, 1]^n$.

In [63], the authors extended the results from [80] for Archimedean quadratic modules using the Łojasiewicz inequality from semialgebraic geometry [18]. These bounds find application in optimization, especially providing conditions for hierarchies like Lassere's [45] to determine an optimal convergence. The degrees of the multipliers and generators $\sigma_e \cdot g_1^{e_1} \ldots g_s^{e_s}$ are bounded as follows: for a non-empty semialgebraic set $S := \mathcal{S}(g_1, \ldots, g_s) \subseteq (-1, 1)^n$, there is some $c \in \mathbb{N}$ such that for every $f \in \mathbb{R}[X_1, \ldots, X_n]$ of degree d with $f^* := \min\{f(x) \mid x \in S\} > 0$ we have

$$\deg(\sigma_e \cdot g_1^{e_1} \ldots g_s^{e_s}) \leq c \exp\left(\left(d^2 n^d \frac{||f||}{f^*}\right)^c\right)$$

2.2.3 Characterization of semialgebraic sets using triangular sets

Since semialgebraic sets and quadratic modules are dually related, understanding the former object can provide helpful insights about the certificates in quadratic modules. Semialgebraic sets are the fundamental objects of study in real algebraic geometry. For a long time, the Collins' Cylindrical Algebraic Decomposition algorithm [27] has been used to study polynomial systems of equalities and inequalities. Although an expensive method, this algorithm does not provide checkable information for a user to verify the correctness of the result.

Work in [23] characterizes semialgebraic sets by extending the definition of Triangular sets for systems of sign conditions. The decomposition of a semialgebraic set is presented using finitely many regular semialgebraic systems. Full (eager) and lazy versions of the decomposition algorithm are discussed. The lazy decomposition provides a preliminary decomposition with the highest dimensional components and unevaluated recursive calls, which provides a full triangular decomposition after being evaluated. According to the authors, the information encoded in the decompositions with the highest dimension is enough for most practical problems. The triangular structure comes primarily from the system of polynomial equations, and the introduced **border polynomial** allows one to check for good specialization to maintain desirable regularity properties in the decomposition. The triangular decomposition method relies on CAD algorithms to check the emptiness of semialgebraic sets and quantifier elimination to test the satisfiability of a particular existentially quantified formula. The latter can be solved more efficiently as an instance of the so-called **real root classification** problem.

In follow-up work [22], the authors improved the theoretical aspects of semialgebraic decomposition, particularly the theory of the **border polynomial** and an

optimized decomposition algorithm using relaxations to simplify the output of the algorithm. Effectively, relaxations replace strict inequalities with non-strict inequalities. This relaxation enhances the decomposition algorithm by avoiding unnecessary recursive calls. A criterion for whether a relaxation does not change the zero set of the semialgebraic set is discussed. Motivated by the problem of checking redundant components, the authors developed set-theoretical operations, including checking whether the zero sets of two different components are subsets of each other and the difference and intersection of zero sets. Experimental results show the feasibility of the current implementation in the `RegularChains` library [47] for the Maple computer algebra system outperforming the number of problems solved to the well-known implementation of the cylindrical algebraic decomposition. However, these approaches do not provide direct information about explicit certificates for verification purposes, as mentioned above. In combination with Positivstellensatz results, this approach is suitable for working with a reasonable membership checking algorithm in quadratic modules by checking strict positivity over semialgebraic sets in the proposed representation.

2.2.4 Membership checking using valuation theory

The certificate and membership problem are deeply interrelated. Our method proposed in this thesis relies on checking initially the membership of a polynomial in some quadratic module to then proceed with the computation of its certificates. The line of research described in this section addresses the membership problem involving valuation theory.

In Jacobi's Ph.D. thesis [34], the Archimedean condition was used to extend Schmüdgen's Positivstellensatz [78] for quadratic modules. The Archimedean condition for quadratic modules is a generalization of the Archimedean property for the

reals, which states that every $r \in \mathbb{R}$ is bounded by a natural number $N \in \mathbb{N}$; a quadratic module M is Archimedean if and only if for every $f \in \mathbb{R}[X_1, \ldots, X_n]$ we have $N - f \in M$ for some $N \in \mathbb{N}$. An equivalent definition of the Archimedean property for quadratic modules is that M is Archimedean if and only if $N - \sum_{i=1}^{m} X_i^2 \in M$ for some $N \in \mathbb{N}$. Using valuation theory, Jacobi and Prestel [35] provided a theoretical criterion to decide the Archimedean property in quadratic modules in the multivariate case by checking if the residue forms over the residue fields are weakly isotropic. This abstract method has provided connections between real algebra and valuation theory, but practical algorithms for the general case have not been developed from this approach.

In her Ph.D. thesis [21], Canto Cabral provided an algorithm to check whether a quadratic module with two indeterminates is Archimedean or not by instantiating Jacobi-Prestel's criterion [35]. In particular, the setting for two indeterminates simplifies the conditions to check for such a criterion, since it is only necessary to consider two kinds of real valuations. Wagner's Ph.D. dissertation [89] proved the decidability of the Archimedean property of a quadratic module for the multivariate case modifying Jacobi-Prestel's criterion using particular valuations known as prime divisors. An explicit algorithm for the latter is still missing. The algorithms mentioned above provide decision methods to determine the Archimedean property for different cases of quadratic modules. However, it is not known whether current applications of these approaches can find explicit certificates of members of these quadratic modules.

2.2.5 Membership checking of quadratic modules in dimension one

As mentioned earlier, the membership and certificates problem are deeply interrelated. The line of work in this section describes an algorithm to check the membership

of polynomials in the univariate setting using local information of the polynomial at the endpoints of the generators of the quadratic modules. This information is presented as signs and orders of both the generators and the polynomial given for the membership test.

A quadratic module $\mathrm{QM}(g_1, \ldots, g_n)$ is **stable** [58] if there exists a function $\ell : \mathbb{Z}^+ \to \mathbb{Z}^+$ that bounds the degree of each term in a member f of the quadratic module by $\deg(\sigma_i g_i) \leq \ell(\deg(f))$. If the number of indeterminates is one and the associated semialgebraic set of a stable quadratic module is unbounded, then it contains a cone for which the bound function ℓ is the identity function, thus it is possible to use an SDP solver to solve a Gram matrix problem to compute certificates for members of this kind of quadratic module [4] or find a parametric representation of the sums of squares multipliers checking the emptiness of the associated semialgebraic set [81].

In her Ph.D. thesis [4], Augustin introduced a procedure to decide the membership of a given polynomial f in an Archimedean quadratic module Q for the univariate case. The method checks if a polynomial is nonnegative over the associated semialgebraic set of Q and verifies the orders of f at the endpoints of the aforementioned semialgebraic set. This result has already been discussed previously in Section 2.1.2, mainly in Theorem 2.5. The sign and order conditions have been studied and introduced as a data structure in [81], which computes a minimal equivalent set of generators for a given Archimedean quadratic module.

Additionally, in [4] the author proves that several cases of quadratic modules are stable. If the semialgebraic set is a collection of isolated points or a collection of regular intervals, a quadratic module Q is stable if and only if Q is saturated. Hence, using the Extended Gram method is possible to determine the membership of polynomials in these quadratic modules; however, as we notice in Section 1.2.2, the certificates obtained using this method are approximation and thus not exact.

2.2.6 Certificates computation using symbolic-numeric approaches

The computing of certificates using semidefinite programming (SDP) solvers and projecting rational solutions has been a popular technique to compute certificates for quadratic modules [52, 68]. The advantages of this approach benefit from the efficiency of the SDP solvers, while rational projection allows them to integrate their solution for reasoning tasks.

The Archimedeaness property of a quadratic module is an important assumption in many representation results [52, 63, 79, 80]. Finding explicit representation of the auxiliary polynomials entailed by the Archimedean property remains an open problem towards a fully constructive proof in the certificates problem for quadratic modules with associated bounded semialgebraic set.

Researchers have developed algorithms to approximate certificates in the multivariate case [52] using representation theorems like Schmüdgen's and Putinar's Positivstellensatz . The approach usually relies on variations of the extended Gram method utilizing SDP solvers. These numerical solutions, while efficient in practice, their numerical errors limit their correctness.

The check of the emptiness of a semialgebraic set or the feasibility of a semidefinite program can decide the membership problem for stable quadratic modules [57, 81]. Under certain conditions, SDP techniques can be used to compute certificates for members in a stable quadratic module [52] using perturbation methods [24] and bounds from Pòlya representations [79]. For example, if the generators of the quadratic module have an unbounded semialgebraic set, then such a quadratic module is a stable quadratic module, and SDP methods can be used to test membership. However, the effectiveness of these approaches is frequently hindered by numerical instabilities arising in interior-point methods and by the challenge of determining

suitable parameter settings for the SDP solvers to achieve reliable convergence.

Although SDP solvers can be efficient in practice, their dependence on numerical methods (interior point algorithms) can lead to precision errors that allow only approximate solutions. A line of research focuses on fixing the SDP solver inaccuracies to provide exact certificates, particularly for sums of squares decompositions. In [68], the authors compute sums of squares decompositions with rational coefficients using a rounding and projection technique by rationalizing the coefficients of the obtained representation and minimizing a least squares error between the coefficients of the original polynomial and the rationalized approximate representation. This procedure depends on geometrical conditions related to the distance between the obtained representation and the space of feasible sums of squares representations. In [36], the authors proposed another method to find exact certificates of sums of squares decompositions by including, in addition to the least squares projection, a "refinement" step involving the Gauss-Newton iteration to find a feasible solution of the SDP which is strictly contained in the interior of the affine space of positive semidefinite matrices.

In order to ensure good performance of the implementations relying on SDP solvers, most authors perturb the polynomials with the intention of ensuring a feasible semidefinite programming problem. While the approach is still able to recover certificates under perturbation, this approach limits the scope of polynomials for which the approach can find certificates only for strictly positive polynomials over the semialgebraic sets of the generators.

2.2.7 Certificates computation using Gröbner bases

The idea of Gröbner bases was introduced by Bruno Buchberger in [20]. In this seminal work, Buchberger developed an algorithmic approach to handle polynomial

ideals, providing a constructive foundation for computational ideal theory. Given an ideal generated by a finite number of polynomials, the original algorithm produces a new set of generators, properly called a basis using a monomial ordering with rich mathematical and computational properties. To produce such bases, the algorithm iterates a reduction step taking "S-pairs" until no more of these can be added to the output of the algorithm. These techniques have been heavily applied in other parts of computational algebra; to name a few, we have ideal membership, radical membership, and elimination procedures [16, 61]. Applied domains have also benefited from the Gröbner basis algorithm; for example, in automated theorem proving [37], robotics and kinematics [33], coding theory and cryptography [75], etc.

Buchberger's algorithm can be studied as a rewriting system [7] that allows effective tracking of the multipliers involved in the certificate problem for ideals. In [79], the author uses the Gröbner basis algorithm to translate new variables that encode the nonnegativity of a set of generators in a simplex domain.

Chapter 3

Compactification of members in Archimedean Quadratic Modules

The Basic Lemma (Theorem 2.19) introduced in [42] provides a constructive foundation for studying quadratic modules with associated compact semialgebraic sets. It was developed with the objective of providing alternative proofs of key abstract results of Scheiderer [76, 77] in a compact setting. As discussed in Section 2.1.7, this lemma is a Bézout-like identity between two polynomials, say f, g, which expresses 1 as a linear combination of the form $1 = \sigma f + \tau g$ where σ, τ are strictly positive over a compact semialgebraic set S. As in the classical Bézout identity, the Basic Lemma requires both f, g to be coprime and nonnegative over S. A specialized version of the Basic Lemma (Theorem 2.20), further assumes that if $S = \mathcal{S}(fg)$ is bounded, then σ, τ can be chosen to be sums of squares [1].

The specialized version of the Basic Lemma is widely used throughout this thesis; however, its requirements, especially the boundedness condition of S, impose

[1] This follows from the application of the Basic Lemma with $G = \{fg\}$ to obtain strictly positive polynomials σ_1, τ_1 over $\mathcal{S}(\{fg\})$ and Putinar's Positivstellensatz (Theorem 2.3) to obtain a certificate of σ_1, τ_1 in $\mathrm{QM}(fg)$.

a restriction that is not applicable in certain situations. To address some of these shortcomings, in this chapter we introduce the concept of **compactification** of a polynomial with respect to the generators G of an Archimedean quadratic module $\mathrm{QM}(G)$. This construction is not limited to any conditions for the set of generators or the input polynomial.

This chapter is divided into two parts. The first part, Section 3.1, introduces the Algorithm 9 to compute the compactification of a given polynomial as well as its certificates if a certificate of the input polynomial is given; this is done in Section 3.1.4. The main algorithm must treat two cases: when the semialgebraic set defined by the input polynomial is unbounded on one side (left or right) and when it is unbounded in both directions. These cases are covered in Section 3.1.1 and Section 3.1.3, respectively. The involution technique discussed in Section 3.1.2 allows us to solve the case when the semialgebraic set of the input polynomial is unbounded to the right by reusing the results for the unbounded to the left case. The second part, Section 3.2, applies the compactification algorithm to solve two related problems. We compute the certificates of nonnegative factors of a member of a quadratic module in Section 3.2.1 and Section 3.2.2 introduce a technique to remove intervals from the semialgebraic set of members of a quadratic module that do not intersect with the semialgebraic of the original generators. We end this chapter with a summary in Section 3.3.

3.1 Compactification

Given a polynomial $f \in \mathrm{QM}(G)$, the compactified polynomial of f, denoted $\mathrm{compact}_G(f)$, is a polynomial such that $\mathcal{S}(\mathrm{compact}_G(f))$ is bounded and $\mathrm{compact}_G(f)$ satisfies the same order and sign conditions at the roots of f. To accomplish the latter, we find an auxiliary strictly positive polynomial h such that fh

satisfies the above properties of $\text{compact}_G(f)$; hence, we define $\text{compact}_G(f) := fh$. Consequently, $\text{compact}_G(f) \in \text{QM}(G)$. We remove the subscript G in compact_G whenever it is clear from the context the set of generators being considered.

If $\mathcal{S}(f)$ is bounded, then $\text{compact}_G(f) = f$; otherwise, there are two cases to consider: the semialgebraic set of the polynomial f is unbounded to the right (resp. left), i.e., $\mathcal{S}(f) = \bigcup_{i=0}^{k} [a_i, b_i]$, with $a_0 \in \mathbb{A}$, and $b_k = \infty$ (resp. $a_0 = -\infty$, and $b_k \in \mathbb{A}$), and when the semialgebraic set f is unbounded from both sides, i.e., $a_0 = -\infty$ and $b_k = \infty$. For the first case, the polynomial h is a monomial of the form $-(X - a)$ (resp. $X - a$) when $\mathcal{S}(f)$ is unbounded to the right (resp. left); for the second case, the polynomial h is of the form $-(X - a)(X - b)$. In all these cases, the polynomial h is always strictly positive over $\mathcal{S}(G)$, hence $h \in \text{QM}(G)$.

In Section 3.1.1 and Section 3.1.3, we introduce algorithms to compute the polynomial h and, given certificates of f in $\text{QM}(G)$, we can compute a certificate of $\text{compact}_G(f)$ in $\text{QM}(G)$. The approach is similar to the Basic Lemma Theorem 2.20 with inputs f and $\epsilon - f$ for some $\epsilon > 0$. We will show that $\epsilon - f = s \cdot h$ where s is a sum of squares and h is the auxiliary polynomial mentioned in the following, so $1 = \frac{1}{\epsilon} f + \frac{1}{\epsilon}(\epsilon - f)$. Hence, we use this equation to obtain a certificate of $\text{compact}_G(f) = fh$ by multiplying fh by both sides; therefore, we obtain $\text{compact}_G(f) = fh = \frac{1}{\epsilon} f^2 \cdot h + \frac{1}{\epsilon}(\epsilon - f) fh = \frac{1}{\epsilon} f^2 \cdot h + \frac{1}{\epsilon} sh^2 \cdot f$. As h is strictly positive over $\mathcal{S}(G)$, we can use Certificate to compute a certificate in $\text{QM}(G)$, so lifting[2] h in the previous equation we obtain a certificate of $\text{compact}_G(f)$ in $\text{QM}(G)$. The advantage of this construction is that we do not use Theorem 2.20 and therefore avoid the use of the SOSBasicLemma algorithm. Finally, we present the compactification algorithm in Section 3.1.4 by putting together the results of the auxiliary sections.

[2]See Appendix A.1

3.1.1 Compactification when $\mathcal{S}(f)$ is unbounded to the right

The intuitive idea is to raise the polynomial $-f$ by a positive constant ϵ such that $\epsilon - f$ only has one real root. The additional roots of $\epsilon - f$ are complex conjugates, so these are sums of squares for the multiplier τ. This ϵ must be chosen so that $\epsilon - f$ is strictly positive over $\mathcal{S}(G)$.

We rely on Cauchy bounds to find an interval that contains all local maxima of f. In the following, we recall the definition of Cauchy bounds and state the key property we need for this purpose.

Lemma 3.1 *[15, Lemma 10.2, page 354] Let $P = a_pX^p + \ldots a_qX^q$, $p > q$ and $a_qa_p \neq 0$ with coefficients in $\mathbb{R}$. Let us denote $C(P) = \sum_{q \leq i \leq p} |\frac{a_i}{a_p}|$.*

The absolute value of any root of P in $\mathbb{R}$ is less than $C(P)$.

Additionally, we state a fundamental theorem in analysis called Rolle's theorem which we use in our construction from [1].

Theorem 3.2 *Let $f : [a, b] \to \mathbb{R}$ be continuous on $[a, b]$ and differentiable on (a, b). If $f(a) = f(b)$, then there exists a point $c \in (a, b)$ where $\frac{df}{dX}(c) = 0$.*

The following constructive proposition tells us how to find such ϵ and prove the intended properties.

Proposition 3.3 *Let $f \in \mathrm{QM}(G)$ such that its semialgebraic set is unbounded to the right and $\mathcal{S}(G) = \bigcup_{i=0}^{k} [a_i, b_i]$ is bounded. There exists a positive constant ϵ such that $\epsilon - f = s \cdot (-(X - a))$ where $-(X - a)$ is strictly positive over $\mathcal{S}(G)$ and s is a sums of squares.*

Proof. Assume the degree of the polynomial f is n. Since $\mathcal{S}(f)$ is unbounded to the right, then n is odd and the sign of the leading coefficient of f is positive. This proof first considers a special case when f has n different real roots $r_1 < \cdots < r_n$; whenever this condition does not hold, the same construction also provides a solution with minor changes.

If f has n different real roots, then we can iterate Theorem 3.2 to show that the interval $[r_1, r_n]$ contains all the roots of the derivative of f. Let $I = [\min(r_1, a_0 - 1), \max(r_n, b_k + 1)]$ and $\epsilon = \max_{x \in I}(f)$. First, we prove that $\epsilon - f$ is strictly positive over $\mathcal{S}(G)$. The interval I strictly contains $\mathcal{S}(G)$, so it suffices to prove that $\epsilon - f$ is nonnegative over I; the latter follows from the definition of ϵ, i.e., for every $x \in I$, $\max_{x \in I}(f) - f(x) \geq 0$. Now, we prove that $\epsilon - f$ is of the form $s \cdot (-(X - a))$ where s is a sums of squares; it is enough to prove $\mathcal{S}(f) = (-\infty, a]$.

As $\mathcal{S}(f)$ is unbounded to the right, we have $\mathcal{S}(-f)$ is unbounded to the left, hence $\mathcal{S}(\epsilon - f)$ is also unbounded to the left. The latter means that $\epsilon - f$ has at least a real root, say a, with odd multiplicity. We prove that a is the only real root of $\epsilon - f$ with odd multiplicity.

From the definition of I, it contains the roots of the derivative of f which include all the local maxima of f. Hence, $\epsilon - f$ is strictly decreasing over $\mathbb{R} \setminus I$; thus $\epsilon - f$ has no roots over $(-\infty, \min(r_1, a_0 - 1))$ and a single root a with odd multiplicity in $(\max(r_n, b_k + 1), \infty)$. We proved that $\epsilon - f$ is nonnegative over $I = [\min(r_1, a_0 - 1), \max(r_n, b_k + 1)]$, thus $\mathcal{S}(f) = (-\infty, a]$.

If f does not have n real different roots, we modify the interval I from the aforementioned construction as follows. Let C be the Cauchy bound of the derivative of f and I the interval $[\min(-C, a_0 - 1), \max(C, b_k + 1)]$. The interval I contains the semialgebraic set $\mathcal{S}(G)$ and the reals roots of the derivative of f using [15, Lemma 10.2 p. 354]. Similarly to the original construction, we define $\epsilon = \max_{x \in I}(f)$. We use

the same argument of the previous case to show $\epsilon - f$ is of the form $s(-(X-a))$ as desired. □

Example 3.4 Let us consider $G = \{-(X+1)(X-1)\}$ and $f = (X+2)(X-2)(X-3)$. The semialgebraic set of G is $[-1,1]$ while $\mathcal{S}(f) = [-2,2] \cup [3,\infty)$ so the semialgebraic set of f is unbounded to the right. The input polynomial f has degree 3 and 3 distinct real roots, hence we compute the interval I from the proof of Proposition 3.3 we calculate the Cauchy bound of f, which is 20. Notice that $I = [-20,20]$ contains $\mathcal{S}(G)$. Then we compute the maximum value of f over I, which is 6732 when $x = 20$, that is, at the right end point of I. We have $\epsilon - f = 6720 + 4X + 3X^2 - X^3 = (336(\frac{17}{672}X + 1)^2 + \frac{1055}{1344}X^2) \cdot (-(X-20))$. Hence, $h = -(X-20)$, which is strictly positive over $\mathcal{S}(G)$.

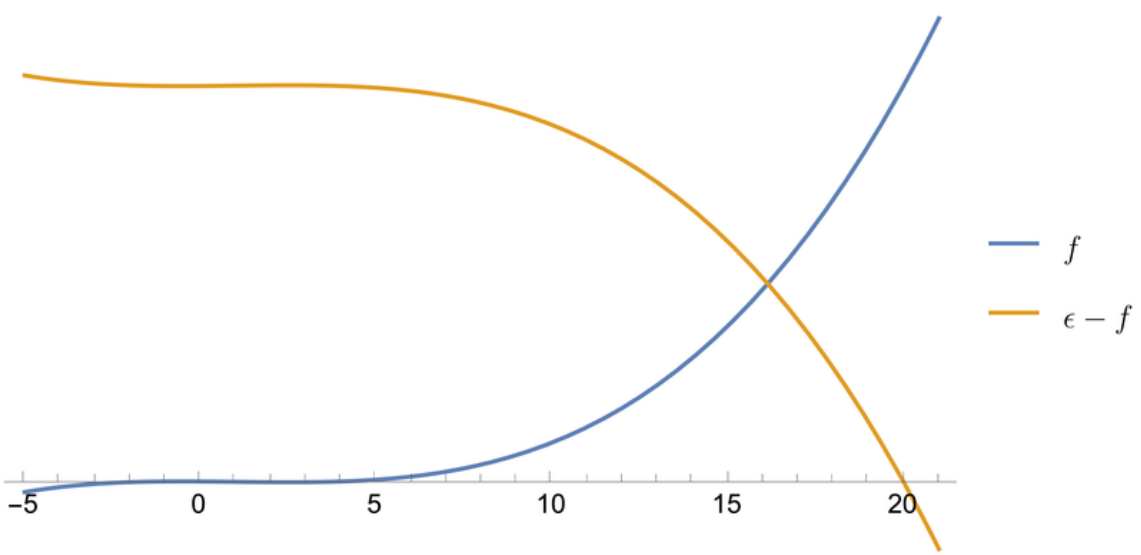

Figure 3.1: Graph of polynomials f $((X+2)(X-2)(X-3))$ and $\epsilon - f$ $(6720+4X+3X^2-X^3)$

We formalize these ideas in Algorithm 7.

Algorithm 7: Computing auxiliary polynomial h when $\mathcal{S}(f)$ is unbounded to the right

Input: $f \in \mathbb{A}[X], bound \in \mathbb{A}$

Output: $\sigma \in \mathbb{A}^+$ and $\tau, h \in \mathbb{A}[X]$

Requires: $\deg(f) = n$ is odd and the leading coefficient of f is positive

Ensures: $1 = \sigma f + \tau h$, $\sigma, \tau \in \sum \mathbb{A}[X]^2$

1 **if** f *has* $r_1 < \cdots < r_n$ *as roots* **then**
2 Let C be the Cauchy bound of f;
3 **else**
4 Let C be the Cauchy bound of $\frac{d}{dX} f$;
5 $I := [-C, \max(bound, C)]$;
6 $\epsilon := \max_{x \in I}(f)$;
7 $\alpha :=$ real root of $\epsilon - f$ of odd degree;
8 $\sigma := \frac{1}{\epsilon}$;
9 $h := -(X - \alpha)$;
10 $\tau := \sigma \cdot quotient(\epsilon - f, h)$;
11 **return** σ, τ, h;

Example 3.5 Let us consider $g_1 = (X+1)(X-2)(X-4)$ and $g_2 = -(X-1)$. The semialgebraic set of $G = \{g_1, g_2\}$ is the interval $[-1, 1]$ so it is bounded, however $\mathcal{S}(g_1)$ is $[-1, 2] \cup [4, \infty)$, which it is unbounded to the right. We use Algorithm 7 to compute the polynomial h such that $\text{compact}(g_1) = g_1 h$. The largest end point of $\mathcal{S}(G)$ of 1, which is used as the bound for Algorithm 7.

The Cauchy bound of g_1 is 16, therefore we compute the maximum value of g_1 between the interval $[-16, 16]$ to be $\epsilon_1 = 2856$. Then we factorize $\epsilon_1 - g_1 = ((X + \frac{11}{2})^2 + \frac{591}{4}) \cdot (-(X - 16))$. Hence, the real root α of $\epsilon_1 - g_1$ of odd degree is 16.

Following the algorithm, $\sigma := \frac{1}{2856}$ is a positive constant. We have $h = -(X - 16)$,

and $\tau = \frac{1}{2856}((X + \frac{11}{2})^2 + \frac{591}{4})$.

3.1.2 Involution technique

In the previous section, we discussed how to compute the auxiliary polynomial h in the compactification of a polynomial f when $\mathcal{S}(f)$ is bounded to the right. The construction for the case where $\mathcal{S}(f)$ is bounded to the left is similar due to symmetric arguments. In this section, we use the involution operation to formalize this observation.

Involution $\iota : \mathbb{R}[X] \to \mathbb{R}[X]$ is a ring automorphism that fixes elements in $\mathbb{R}$ and sends X to $-X$. We prove the following properties:

Proposition 3.6 *Let $f \in \mathbb{R}[X]$ and $f(a)$ be the homomorphism substitution of a that replaces a for every X in f. The following holds:*

- $\iota((X-a)^k) = (-1)^k(X+a)^k$
- $(\iota(f)^{(k)})(a) = (-1)^k f^{(k)}(-a)$
- *If f is a sums of squares, then $\iota(f)$ is also a sums of squares.*

Proof. For the first item, since ι is an automorphism, we have $\iota((X-a)^k) = (\iota(X-a))^k = (-X-a)^k = (-(X+a))^k = (-1)^k(X+a)^k$.

For the second item, the Taylor series expansion of f at $-a$ is $f = \sum_{i=0}^{n} \frac{f^{(k)}(-a)}{k!}(X+a)^k$. Applying the automorphism ι to the latter, we have

$$\begin{aligned} \iota(f) &= \sum_{i=0}^{n} \frac{f^{(k)}(-a)}{k!}\iota((X+a)^k) \\ &= \sum_{i=0}^{n} \frac{f^{(k)}(-a)}{k!}(-1)^k(X-a)^k \end{aligned} \tag{3.1}$$

by the first item of this proposition. The Taylor series expansion of $\iota(f)$ at a is

$$\iota(f) = \sum_{i=0}^{n} \frac{(\iota(f))^{(k)}(a)}{k!}(X-a)^k \tag{3.2}$$

Subtracting Equation (3.1) and Equation (3.2) gives the desired equality.

For the third item, if f is a sums of squares then there exists $f_1, \ldots, f_s$ such that $f = \sum_{i=1}^{s} f_i^2$. Since ι is an automorphism, we have $\iota(f) = \sum_{i=1}^{s} \iota(f_i)^2$. □

The following corollary is obtained directly from the previous proposition.

Corollary 3.7 *Let $f \in \mathbb{R}[X]$. The following holds:*

- $\mathrm{ord}_a(\iota(f)) = \mathrm{ord}_{-a}(f)$
- $\epsilon_a(\iota(f)) = (-1)^{\mathrm{ord}_{-a}(f)}\epsilon_{-a}(f)$

The previous corollary tells us that involution preserves the order at a at the reflected point $-a$. Additionally, the ϵ sign is preserved as well modulo the parity of the order.

Example 3.8 In this example, we combine the involution technique and the Algorithm 7 to compute the auxiliary polynomial h when $\mathcal{S}(f)$ is unbounded to the left.

Consider $G = \{-(X+1)(X-1)\}$ and $f = -(X-2)(X-3)(X-5)$. First, we apply the involution to the input data. In this case $\hat{G} := \iota(G) = \{-(X+1)(X-1)\}$ and $\hat{f} := \iota(f) = (X+5)(X+3)(X+2)$. Notice that the semialgebraic set of $\hat{f}$ is $[-5,-3] \cup [-2,\infty)$, so it is unbounded to the right. Hence, we apply Algorithm 7 to $\hat{f}$.

The Cauchy bound of $\hat{f}$ is 72 so the interval I is $[-72, 72]$, which includes $\mathcal{S}(\hat{G})$. We obtain $\epsilon = 427350$, so $\epsilon - \hat{f} = ((X+41)^2+4254)\cdot(-(X-72))$ hence $\hat{h} = -(X-72)$, which is strictly positive over $\mathcal{S}(\hat{G})$. To obtain h for our original problem, we simply apply involution $\hat{h}$, thus $h = X + 72$. Notice that $1 = \frac{1}{427350} f + \frac{1}{427350}((X-41)^2 + 4254)h$.

3.1.3 Compactification when $\mathcal{S}(f)$ is unbounded in both sides

Similarly to the unbounded case to the right in Section 3.1.1, we raise the polynomial $-f$ by a positive constant. Since f is unbounded to both sides, the polynomial $\epsilon - f$ will have only two real roots, and the rest of its roots are complex conjugates. We present these ideas in Algorithm 8.

Proposition 3.9 *Let $f \in \mathrm{QM}(G)$ be such that its semialgebraic set is unbounded on both sides and $\mathcal{S}(G) = \bigcup_{i=0}^{k} [a_i, b_i]$ is bounded. If f is not a positive constant, there exists a positive ϵ such that $\epsilon - f = s \cdot (-((X-a)(X-b)))$ where $-(X-a)(X-b)$ is strictly positive over $\mathcal{S}(G)$ and s is a sums of squares.*

Proof. We follow a similar approach to the proof of Proposition 3.3; we discuss two cases whether all the real zeros of f are distinct or not. For the former case, we set the interval I as $[\min(r_1, a_0 - 1), \max(r_n, b_k + 1)]$ where r_1 (resp. r_n) is the minimum (resp. maximum) root of f and let ϵ be $\max_{x \in I}(f)$. First, we prove that $\epsilon - f$ is strictly positive over $\mathcal{S}(G)$. This follows from the fact that $\epsilon - f$ is nonnegative over I and $\mathcal{S}(G) \subset I$.

Now, we prove that $\epsilon - f$ is of the form $s \cdot (-(X-a)(X-b))$ where $a < b$ and s is a sums of squares; it is enough to prove $\mathcal{S}(f) = [a, b]$. As $\mathcal{S}(f)$ is unbounded on both sides, $\mathcal{S}(-f)$ is either bounded, hence $\mathcal{S}(\epsilon - f)$ is bounded. This means $\epsilon - f$

has at least two real roots, say a and b, with odd multiplicities. We prove a, b are the only real roots of $\epsilon - f$ with odd multiplicity.

From the definition of I, it contains the roots of the derivative of f which include all the local maxima of f. Hence, $\epsilon - f$ is strictly decreasing over $\mathbb{R} \setminus I$; thus $\epsilon - f$ has a single root a in $(-\infty, \min(r_1, a_0 - 1))$ and a single root b in $(\max(r_n, b_k + 1))$ both with odd multiplicities. We proved that $\epsilon - f$ is nonnegative over $I = [\min(r_1, a_0 - 1), \max(r_n, b_k + 1)]$, thus $\mathcal{S}(f) = [a, b]$.

If f does not have n real different roots, we modify the interval I from the aforementioned construction similarly to the proof of Proposition 3.3. Let C be the Cauchy bound of the derivative of f and I the interval $[\min(-C, a_0 - 1), \max(C, b_k + 1)]$. The construction discussed previously allows us to obtain the results in our statement, as I contains the roots of the derivative of f. $\square$

Example 3.10 Consider $G = \{g_1, g_2, g_3\}$ where $g_1 = X + 3$, $g_2 = (X + 2)(X - 2)$, $g_3 = -(X - 3)$ and $f = (X + 2)(X + 1)(X - 1)(X - 2)$. The semialgebraic set of G is $[-3, -2] \cup [2, 3]$ while $\mathcal{S}(f) = (-\infty, -2] \cup [-1, 1] \cup [2, \infty)$, so the semialgebraic set of f is unbounded to both sides. To compute the interval I from the proof of Proposition 3.3 we calculate the Cauchy bound of f, which is 10. Notice that $I = [-10, 10]$ contains $\mathcal{S}(G)$. Then we compute the maximum value of f over I, which is 9504 at $x = -10$, i.e., at the right end point of I.

We have $\epsilon - f = 9500 + 5X^2 - X^4 = (X^2 + 95) \cdot (-(X + 10)(X - 10))$. Hence, $h = -(X + 10)(X - 10)$, which is strictly positive over $\mathcal{S}(G)$.

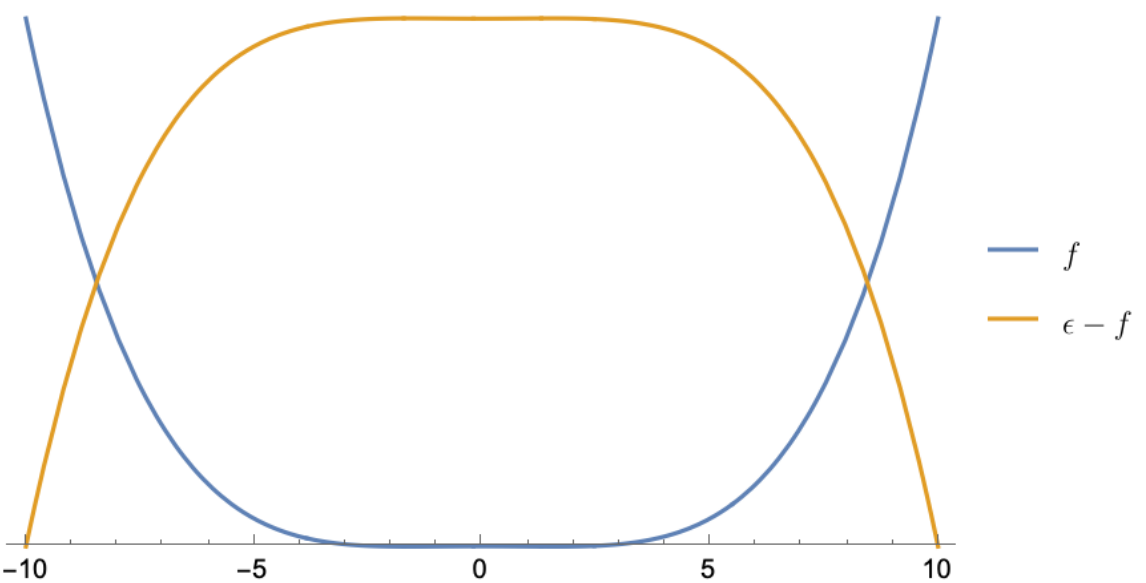

Figure 3.2: Graph of the polynomials f $((X + 2)(X + 1)(X - 1)(X - 2))$ and $\epsilon - f$ $(9500 + 5X^2 - X^4)$

Algorithm 8: Computing auxiliary polynomial h when $\mathcal{S}(f)$ is unbounded from both sides

Input: $f \in \mathbb{A}[X], lowerbound, upperbound \in \mathbb{A}$

Output: $\sigma \in \mathbb{A}^+$ and $\tau, h \in \mathbb{A}[X]$

Requires: $\deg(f)$ is even and leading coefficient of f is positive

Ensures: $1 = \sigma f + \tau h$, $\sigma, \tau \in \sum \mathbb{A}[X]^2$

if *f is a positive constant* **then**
  return $\frac{1}{f}$, *0, 0*;
else
  if *f has $r_1 < \cdots < r_n$ as roots* **then**
    Let C be the Cauchy bound of f;
  else
    Let C be the Cauchy bound of $\frac{d}{dX}f$;
  $I := [\min(lowerbound, -C), \max(upperbound, C)]$;
  $\epsilon := \max_{x \in I}(f)$;
  $\sigma := \frac{1}{\epsilon}$;
  Let a, b be the two real roots of $\epsilon - f$;
  $h := -(X - a)(X - b)$;
  $\tau := \sigma \cdot quotient(\epsilon - f, h)$;
  return σ, τ, h;

3.1.4 Compactification algorithm

We assume that $\mathcal{S}(G) = \bigcup_{i=0}^{k} [a_i, b_i]$. Using Algorithm 7 and Algorithm 8 to compute the auxiliary polynomial h and the sums of squares σ, τ for the identity $1 = \sigma f + \tau h$, we finally present in Algorithm 9 the compactification algorithm. We set the values for the lowerbounds and upperbounds with a_0 and b_k respectively to force h to be strictly positive over $\mathcal{S}(G)$; using the Certificate algorithm, we compute a certificate of h in $\mathrm{QM}(G)$. Finally, we use Algorithm 9 to compute a certificate of compact(f)

in $\mathrm{QM}(G)$.

Algorithm 9: Compactification algorithm $\mathrm{compact}_G(f)$

Input: $f \in \mathbb{A}[X]$, $G = \{g_1, \ldots, g_s\} \subseteq \mathbb{A}[X]$

Output: $\tau_0, \tau_1, \ldots, \tau_s \in \mathbb{A}[X]$

Requires: $f = \sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i \in \mathrm{QM}(G)$ and $\mathcal{S}(G) = \bigcup_{i=0}^{k} [a_i, b_i]$

Ensures: $\mathrm{compact}_G(f) = \tau_0 + \sum_{i=1}^{s} \tau_i \cdot g_i$, $\tau_i \in \sum \mathbb{A}[X]^2$ and $\mathcal{S}(\mathrm{compact}_G(f))$ is bounded

1 **if** $\mathcal{S}(f)$ *is bounded* **then**
2 | **return** $\sigma_0, \sigma_1, \ldots, \sigma_s$
3 **if** $\mathcal{S}(f)$ *is unbounded to the right* **then**
4 | Let σ, τ, h be the output of Algorithm 7 with input f, $b_k + 1$
5 **if** $\mathcal{S}(f)$ *is unbounded to the left* **then**
6 | Let σ, τ, h be the output of Algorithm 7 with input $f[X/-X]$, $a_0 - 1$
7 | $\sigma := \sigma[X/-X], \tau := \tau[X/-X], h := h[X/-X]$
8 **if** $\mathcal{S}(f)$ *is unbounded from both sides* **then**
9 | Let σ, τ, h be the output of Algorithm 8 with input f, $a_0 - 1$, $b_k + 1$
10 Let $\rho_0, \ldots, \rho_s$ be the certificate obtained from the Certificate algorithm with input G, h
11 **return** $\sigma f^2 \rho_0 + \tau h^2 \sigma_0, \ldots, \sigma f^2 \rho_s + \tau h^2 \sigma_s$

Since both Algorithm 7 and Algorithm 8 satisfy $1 = \sigma f + \tau h$, multiplying $\mathrm{compact}(f) = fh$ on both sides of the previous equation gives $fh = \sigma f^2 \cdot h + \tau h^2 \cdot f$. Since h is strictly positive over $\mathcal{S}(G)$, we can use the Certificate algorithm to compute a certificate for it in $\mathrm{QM}(G)$, hence the result.

Example 3.11 We will continue with Example 3.5. We need to compute a certificate of h in $\mathrm{QM}(g_1, g_2)$. Using Certificate, we obtain $h = (\frac{52795}{16542}(X + \frac{19711}{211180})^2 + \frac{22498742519}{13973358240}) + \frac{52795}{33084} \cdot g_1 + \frac{52795}{33084}(X-1)^2 \cdot g_2$. Notice that there is a simpler certificate of h

in $\mathrm{QM}(g_1, g_2)$ using $h = 15 + g_2$ that we will use in this example. Hence, a certificate of the compactified generator g_1 is $\mathrm{compact}_{\{g_1,g_2\}}(g_1) = 15\sigma g_1^2 + h^2\tau \cdot g_1 + \sigma g_1^2 \cdot g_2$. We can verify $\mathcal{S}(\mathrm{compact}_{\{g_1,g_2\}}(g_1)) = [-1, 2] \cup [4, 16]$.

Example 3.12 The following example shows that Algorithm 9 can effectively handle the case when the associated semialgebraic set of the generators is empty. Consider the generators $G = \{g_1, g_2, g_3\}$ where $g_1 := X+5$, $g_2 := -(X+2)(X+1)(X-2)(X-3)$ and $g_3 := (X + 1)(X - 4)$. We want to compactify the generator g_1. Since $\mathcal{S}(g_1)$ is unbounded to the right, we use Algorithm 7 with inputs g_1 and 1 as bound to compute sums of squares σ, τ and an auxiliary polynomial h. The output of this step is $\sigma = \frac{1}{11}, \tau = \frac{1}{11}$ and $h = -(X - 6)$. Then, we compute the certificate of h in $\{g_1, g_2, g_3\}$ as any polynomial is strictly positive over the empty set. For the latter, we use the Certificate algorithm to obtain $h = \rho_0 + \rho_2 \cdot g_2$ where:

$$\rho_0 = \tfrac{576474}{37831}(-\tfrac{1231613}{5764740}X^2 + \tfrac{195161}{1152948}X + 1)^2 + \tfrac{299040382003}{436171757880}(-\tfrac{95421750827}{299040382003}X^2 + X)^2 + \tfrac{1272229014367963}{282824917288887325}(X^2)^2$$

$$\rho_2 = \tfrac{29124}{37831}$$

Finally, we obtain a certificate of $\mathrm{compact}(g_1) = g_1 h$ by combining the certificates of g_1 and h on line 11 of Algorithm 9, we obtain $\mathrm{compact}_G(g_1) = \sigma g_1^2 \rho_0 + \tau h^2 \cdot g_1 + \sigma g_1^2 \rho_2 \cdot g_2$.

3.2 Applications to certificate problems in quadratic modules

In this section, we use the compactification algorithm to solve two problems in the computation of certificates. The first application is discussed in Section 3.2.1 to

compute the certificate of a nonnegative factor of a member of a quadratic module from the certificates of the latter. The second application is discussed in Section 3.2.2; this application becomes crucial in our main procedure of Chapter 4 and Chapter 6 as it allows us to modify the generators to compute certificates of natural generators in terms of the original set of generators using the Basic Lemma.

3.2.1 Certificates of nonnegative factors of members of QM(G)

The calculation of a certificate of natural generators in terms of the original set of generators G of generators is an important step in the algorithms developed in chapters 4, 5, and 6 of this thesis. Since natural generators are nonnegative factors for certain elements of G, it is necessary to develop a method to obtain certificates for such types of polynomials.

Suppose that $f = t_1 t_2 \in \mathrm{QM}(G)$ where both t_1 and t_2 are coprime and nonnegative over $\mathcal{S}(f)$ and assume that we have certificates of f in QM(G), i.e., $f = \sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i$. Our goal is to compute a certificate of t_1 in QM(G) using the certificates $\sigma_0, \ldots, \sigma_s$ of f. We have two cases considering the boundedness of $\mathcal{S}(f)$.

In case $\mathcal{S}(f)$ is bounded, using SOSBasicLemma with inputs t_1 and t_2, we compute sums of squares σ, τ such that $1 = \sigma t_1 + \tau t_2$. The certificate of t_1 is computed by multiplying t_1 by both sides of the previous equation, thus obtaining $t_1 = t_1^2\sigma + \tau \cdot f$. Finally, we obtain a certificate of t_1 in QM(G) by substituting f with its certificate in QM(G).

In case $\mathcal{S}(f)$ is unbounded, we use the compactification algorithm 9 to compactify the polynomial f. Then, the above construction for the bounded case is used, thus

finding a certificate of the factor of t_1 in $\mathrm{QM}(\mathrm{compact}_G(g))$. This certificate is lifted to $\mathrm{QM}(G)$ as Algorithm 9 provides a certificate of $\mathrm{compact}_G(f)$ in $\mathrm{QM}(G)$.

3.2.2 Removing redundant intervals from semialgebraic sets

In this section, we discuss a method that addresses a shortcoming of Section 3.2.1 on the requirement that both factors of a given member f in $\mathrm{QM}(G)$ are nonnegative over $\mathcal{S}(f)$. The latter limits the application of the Basic Lemma (Theorem 2.19). We motivate this construction with the following example.

Example 3.13 Consider the following set of generators $G = \{g_1, g_2\}$ where $g_1 = -(X+4)(X+3)(X+1)(X-1)(X-3)(X-4)$ and $g_2 = (X+2)(X-2)$. The natural generators of $\mathrm{QM}(G)$ are $n_1 = X+4, n_2 = (X+3)(X-3)$ and $n_3 = -(X-4)$. Certificates of the natural generators n_1 (resp. n_3) can be obtained using the method discussed in Section 3.2.1 using g_1 as these polynomials are nonnegative over $\mathcal{S}(\mathrm{compact}_G(g_1))$. However, this method is not applicable to the natural generator n_2 as it is not nonnegative over $\mathcal{S}(\mathrm{compact}_G(g_1))$ nor over $\mathcal{S}(\mathrm{compact}_G(g_2))$.

In general, given a polynomial $f \in \mathrm{QM}(G)$ such that

$$\begin{aligned} \mathcal{S}(f) &= \bigcup_{i=0}^{j} [c_i, d_i] \cup \bigcup_{i=j+1}^{k} [c_i, d_i] \cup \bigcup_{i=k+1}^{l} [c_i, d_i] \\ \mathcal{S}(G) &\cap \bigcup_{i=j+1}^{k} [c_i, d_i] = \emptyset \end{aligned} \tag{3.3}$$

we want to find a polynomial $p \in \mathrm{QM}(G)$ such that $\mathcal{S}(f+p) \cap (d_j, c_{k+1}) = \emptyset$ and $f+p$ have the same order and sign conditions at d_j, c_{k+1} as f. Clearly, if certificates of $f \in \mathrm{QM}(G)$ are given, then it is enough to find certificates of $p \in \mathrm{QM}(G)$ to have

a certificate of $f + p \in \mathrm{QM}(G)$. In this way, we can find a certificate of the natural generator n_2 in Example 3.13 by removing the interval $[-1, 1]$ from the semialgebraic set of g_1 and applying the method in Section 3.2.1.

We describe a procedure to compute a certificate of $f + p$ mentioned above. We assume $f = \sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i$. A high level description is the following:

1. Compute positive constants α, ϵ_1, and ϵ_2 satisfying the following conditions:

   - $d_j + \epsilon_1 < c_{j+1}$ and $d_k < c_{k+1} - \epsilon_2$
   - Let $h = (X - (d_j + \epsilon_1))(X - (c_{k+1} - \epsilon_2))$, $q = (X - d_j)^{\gamma_1}(X - c_{k+1})^{\gamma_2}$ where

$$\gamma_1 = \begin{cases} \mathrm{ord}_{d_j}(f) + 1 & \text{if } \mathrm{ord}_{d_j}(f) \text{ is odd} \\ \mathrm{ord}_{d_j}(f) + 2 & \text{otherwise} \end{cases}$$
$$\gamma_2 = \begin{cases} \mathrm{ord}_{c_{k+1}}(f) + 1 & \text{if } \mathrm{ord}_{c_{k+1}}(f) \text{ is odd} \\ \mathrm{ord}_{c_{k+1}}(f) + 2 & \text{otherwise} \end{cases} \tag{3.4}$$

   - $\mathcal{S}(f + \alpha q \cdot h) \cap (d_j, c_{k+1}) = \emptyset$

2. Since h is strictly positive over $\mathcal{S}(G)$, compute a certificate $\tau_0 + \sum_{i=1}^{s} \tau_i \cdot g_i$ of h using the Certificate algorithm.

3. Return certificate $\sigma_0 + \alpha q \tau_0, \ldots, \sigma_s + \alpha q \tau_s$.

In the next proposition, we prove that the polynomial $\alpha q \cdot h$ above satisfies the required order and sign conditions at d_j, c_{k+1}.

Proposition 3.14 *Let f and $p := \alpha q \cdot h$ be the polynomials described above. We have*

- $\text{ord}_{d_j}(f+p) = \text{ord}_{d_j}(f)$, $\text{ord}_{c_{k+1}}(f+p) = \text{ord}_{c_{k+1}}(f)$
- $\epsilon_{d_j}(f+p) = \epsilon_{d_j}(f)$, $\epsilon_{c_{k+1}}(f+p) = \epsilon_{c_{k+1}}(f)$

Proof. Since $\text{ord}_{d_j}(f) < \text{ord}_{d_j}(p)$ and $\text{ord}_{c_{k+1}}(f) < \text{ord}_{c_{k+1}}(p)$, the polynomial $f+p$ maintains the same sign and order values at d_j and c_{k+1}. Thus, the statement holds. □

Example 3.15 Continuing with Example 3.13, we want to remove the redundant interval $[-1,1]$ from the semialgebraic set of g_1. We find that $\alpha = \frac{175}{216}$ and $\epsilon_1 = \epsilon_2 = 1$ satisfy the conditions mentioned above. We have

$$\begin{aligned} f &= g_1 + \frac{175}{216}((X+3)(X-3))^2 \cdot (X+2)(X-2) \\ &= \frac{41X^2+79}{216} \cdot (-(X+6)(X+3)(X-3)(X-6)) \\ &= g_1 + \frac{175}{216}((X+3)(X-3))^2 \cdot g_2 \end{aligned}$$

The semialgebraic set of the polynomial f is $[-6,-3] \cup [3,6]$. A certificate of f in $\text{QM}(g_1, g_2)$ is straightforward to compute as the strictly positive polynomial h used is g_2. Hence, we can apply Section 3.2.1 to compute a certificate of $(X+3)(X-3)$ in $\text{QM}(g_1, g_2)$ as follows:

1. Let $f_1 = \frac{41X^2+79}{216}(-(X+6)(X-6))$ and $f_2 = (X+3)(X-3)$
2. We use SOSBasicLemma with inputs f_1 and f_2 to find
   - $\sigma = \frac{1}{56} - \frac{239}{13414478} f_1 f_2$
   - $\tau = \frac{41}{12096}X^2 - \frac{257}{3024} + \frac{239}{13414478} f_1^2$

   such that σ, τ are strictly positive over $\mathcal{S}(f)$ and $1 = \sigma f_1 + \tau f_2$.

3. We check σ is nonnegative over $\mathbb{R}$, hence it can be expressed as a sums of squares. On the other hand, τ is not nonnegative over $\mathbb{R}$, in fact $\mathcal{S}(\tau) \cong (-\infty, -2.9755] \cup [2.9755, \infty)$. We use Certificate to compute a certificate of τ in $\mathrm{QM}(f)$. We obtain $\tau = s_0 + s_1 \cdot f$ where

   - $s_1 = (\frac{f-902}{910})^{648}$
   - $s_0 = \tau - s_1 f$

4. Finally, we obtain

$$\begin{aligned}
&(X+3)(X-3)\\
&= \sigma \cdot f + f_2^2 \cdot \tau\\
&= \sigma \cdot f + f_2^2 (s_0 + s_1 \cdot f)\\
&= f_2^2 s_0 + (f_2^2 s_1 + \sigma) \cdot f\\
&= f_2^2 s_0 + (f_2^2 s_1 + \sigma) \cdot (g_1 + \frac{175}{216}((X+3)(X-3))^2 \cdot g_2)\\
&= f_2^2 s_0 + (f_2^2 s_1 + \sigma) \cdot g_1 + \frac{175}{216}((X+3)(X-3))^2 (f_2^2 s_1 + \sigma) \cdot g_2
\end{aligned}$$

Example 3.16 In this example, we modify Example 3.13 by changing the multiplicities of $X+3$ and $X-3$ to $g_1 = -(X+4)(X+3)^3(X+1)(X-1)(X-3)^5(X-4)$. The goal is also to remove the interval $[-1, 1]$ from the semialgebraic set of g_1. Using the same parameters $\alpha, \epsilon_1, \epsilon_2$ as in Example 3.15, we have

$$\begin{aligned}
f &= \frac{41X^2 + 79}{216} \cdot (-(X+6)(X+3)^3(X-3)^5(X-6))\\
&= g_1 + \frac{175}{216}((X+3)^2(X-3)^3)^2 \cdot (X+2)(X-2)
\end{aligned}$$

The semialgebraic set of f is $[-6, -3] \cup [3, 6]$.

Example 3.17 Consider $g_1 = -(X+3)^2(X+1)(X-1)(X-3)^4$ and $g_2 = (X+2)(X-2)$. We want to remove the redundant interval $[-1,1]$ from $\mathcal{S}(g_1)$. Using the parameters $\alpha = \frac{5}{196}, \epsilon_1 = 1, \epsilon_2 = 1$, we have

$$\begin{aligned}
&g_1 + \frac{5}{196}((X+3)^2(X-3)^3)^2 \cdot (X+2)(X-2) \\
&= g_1 + \frac{5}{196}((X+3)^2(X-3)^3)^2 \cdot g_2 \\
&= (\frac{89}{196}(-\frac{169}{890}X^2+1)^2 + \frac{19}{980}(X)^2 + \frac{2277}{249200}(X^2)^2) \\
&\cdot (X+4)(X+3)^2(X-3)^4(X-4)
\end{aligned}$$

The semialgebraic set of the last polynomial is $(-\infty,-4] \cup \{-3,3\} \cup [4,\infty)$.

In order to find values for $\alpha, \epsilon_1, \epsilon_2$ in step 1 in the high level description above, first we compute a value for α fixing the values of ϵ_1, ϵ_2 with $\frac{c_{j+1}-d_j}{2}, \frac{c_{k+1}-d_k}{2}$, respectively, such that $\mathcal{S}(\alpha h) \cap \bigcup_{i=j+1}^{k} [c_i, d_i]$ is empty. The polynomial $p = \alpha q \cdot h$ introduces additional roots in d_j and c_{k+1}, so the semialgebraic set of $\mathcal{S}(f+p)$ might have intervals near these endpoints; to fix the latter, we update the values of ϵ_1, ϵ_2 from $(0, \frac{c_{j+1}-d_j}{2}), (0, \frac{c_{k+1}-d_k}{2})$, respectively, to remove these intervals. We use the following proposition to find a value for α:

Proposition 3.18 *Let $f \in \mathrm{QM}(G)$ with semialgebraic set as in* (3.3)*. Let $p = (X-d_j)^{\gamma_1}(X-c_{k+1})^{\gamma_2} \cdot (X - \frac{c_{j+1}-d_j}{2})(X - \frac{c_{k+1}-d_k}{2})$ where γ_1, γ_2 are defined in* (3.4)*. There exists a positive α such that $\mathcal{S}(f+\alpha \cdot p) \cap \bigcup_{i=j+1}^{k} [c_i, d_i]$ is empty.*

Proof. Let $S = \bigcup_{i=j+1}^{k} [c_i, d_i]$. The semialgebraic set of p is $(d_j, \frac{c_{j+1}-d_j}{2}] \cup [\frac{c_{k+1}-d_k}{2}, c_{k+1})$, and therefore p is strictly negative over S. For each of the intervals $[c_i, d_i]$ in S, we compute the maximum value of p and the maximum value of f, i.e., $\alpha_i = -\frac{\max_{x \in [c_i,d_i]}(f)}{\max_{x \in [c_i,d_i]}(p)} > 0$. We will prove that $\mathcal{S}(f + (\alpha_i + 1) \cdot p) \cap [c_i, d_i]$ is empty. Otherwise,

there exists a point $x \in [c_i, d_i]$ such that $f(x) + (\alpha_i + 1) \cdot p(x) \geq 0$, which implies $f(x) + p(x) \geq -\alpha_i \cdot p(x) = \max_{x\in[c_i,d_i]}(f)\frac{p(x)}{\max_{x\in[c_i,d_i]}(p)}$. Notice that $\frac{p(x)}{\max_{x\in[c_i,d_i]}(p)} \geq 1$ as p is negative over $[c_i, d_i]$ so $f(x)+p(x) \geq \max_{x\in[c_i,d_i]}(f)$, which is a contradiction. In fact, any $\beta > \alpha_i$ guaranties that $\mathcal{S}(f + \beta \cdot p) \cap [c_i, d_i]$ is empty. From this, choose $\alpha := 1 + \max\{\alpha_{j+1}, \ldots, \alpha_k\}$, which satisfies the property of empty intersection with each interval in S. □

Proposition 3.19 *Let the polynomial f and the positive constant α be as in Proposition* 3.18. *Let $p_1 = \alpha(X - d_j)^{\gamma_1}(X - c_{k+1})^{\gamma_2} \cdot (X - (d_j + \epsilon_1))(X - (c_{k+1} - \epsilon_2))$ where γ_1, γ_2 are defined in* (3.4). *There exist $\epsilon_1 \in (0, \frac{c_{j+1}-d_j}{2})$ and $\epsilon_2 \in (0, \frac{c_{k+1}-d_k}{2})$ such that the semialgebraic set of $f + p_1$ intersecting (d_j, c_{k+1}) is empty.*

Proof. Let p be from Proposition 3.18, and $t := \alpha \cdot p$. First, we prove that for every $x \in [\frac{c_{j+1}-d_j}{2}, \frac{c_{k+1}-d_k}{2}]$ we have $p_1(x) \leq t(x)$. By continuity, it suffices to prove the existence of a point $x \in [\frac{c_{j+1}-d_j}{2}, \frac{c_{k+1}-d_k}{2}]$ such that $p_1(x) \leq t(x)$ since both p_1 and t have no additional roots in $[\frac{c_{j+1}-d_j}{2}, \frac{c_{k+1}-d_k}{2}]$. This is satisfied with $x = \frac{c_{j+1}-d_j}{2}$ since $t(x)$ is zero while $p_1(x)$ is a negative value. This implies that the polynomial p_1 also satisfies the property $\mathcal{S}(f + p_1) \cap [\frac{c_{j+1}-d_j}{2}, \frac{c_{k+1}-d_k}{2}]$ is empty.

Now, we prove the existence of ϵ_1 and ϵ_2 satisfying the main statement. Suppose, on the contrary, that for every $\epsilon_1 \in (0, \frac{c_{j+1}-d_j}{2})$ and $\epsilon_2 \in (0, \frac{c_{k+1}-d_k}{2})$, we have $\mathcal{S}(f + p_1) \cap (d_j, c_{j+1})$ is non-empty. Let $I_1 = (d_j, d_j + \epsilon_1)$, $I_2 = (c_{k+1} - \epsilon_2, c_{k+1})$, and let us choose a value of ϵ_1, ϵ_2 close to 0 such that f strictly decreases over I_1 and strictly increases over I_2. Clearly, $f+p_1$ is non-positive over $[d_j+\epsilon_1, \frac{c_{j+1}-d_j}{2}]$ and non-positive over $[\frac{c_{k+1}-d_k}{2}, c_{k+1} - \epsilon_2]$, so we will prove that $\mathcal{S}(f+p_1) \cap (I_1 \cup I_2)$ is also empty, thus reaching a contradiction.

We prove if $\mathcal{S}(f+p_1) \cap I_1$ is non-empty, then it has at most one single interval. If $\mathcal{S}(f+p_1) \cap I_1$ has more than a single interval, it means that $f+p_1$ should have at least two local maxima and one local minimum. The latter implies that the derivative of

$f+p_1$ has at least three real zeros on I_1. However, since p_1 has a single local maximum in I_1, its derivative has a single real zero in I_1. Additionally, f is strictly decreasing over I_1, so the derivative of f is negative over I_1, therefore the derivative of $f+p_1$ can have at most two real zeros in I_1. Therefore, if $\mathcal{S}(f+p_1) \cap I_1$ is not empty, then it is a single interval. Furthermore, the left endpoint of this interval is d_j as it is a common zero of f and p_1. A similar argument concludes that if $\mathcal{S}(f+p_1) \cap I_2$ is non-empty, then it has at most a single interval with right endpoint c_{k+1}.

Now, we will prove that if $\mathcal{S}(f+p_1) \cap I_1$ is a single interval with left endpoint d_j, we reach a contradiction, proving our main assertion about the existence of a value of ϵ_1. First, we compute the Taylor series expansion of f and p_1 at d_j, i.e., $f = \beta_1(X-d_j)^{\mathrm{ord}_{d_j}(f)} + \dots$ and $p_1 = \alpha\gamma_1(X-d_j)^{\mathrm{ord}_{d_j}(f)+r} + \dots$ where $\beta_1 < 0$ and r is either 1 or 2. Then $f+p_1 = (\beta_1 + (X-d_j)^r(\dots))(X-d_j)^{\mathrm{ord}_{d_j}(f)}$. If we pick a point δ in I_1 close to d_j, then the sign of $f(\delta)+p_1(\delta)$ is the sign of β_1, which is negative, as $(\delta-d_j)^r$ is close to 0 as well as $(\delta-d_j)^{\mathrm{ord}_{d_j}(f)} > 0$. However, this contradicts the fact that $\delta \in \mathcal{S}(f+p_1) \cap I_1$. A similar argument also shows that $\mathcal{S}(f+p_1) \cap I_2$ is empty. Therefore, there exist values of $\epsilon_1 \in (0, \frac{c_{j+1}-d_j}{2})$ and $\epsilon_2 \in (0, \frac{c_{k+1}-d_k}{2})$ such that $\mathcal{S}(f+p_1) \cap (d_j, c_{k+1})$ is empty. □

The Algorithm 10 allows us to compute the parameters $\alpha, \epsilon_1, \epsilon_2$ required by the construction.

Algorithm 10: Removing redundant in-between strictly positive factors

RemoveStrictPosBetween(f)

Input: $f \in \mathbb{Q}[X], d_j, c_{k+1} \in \mathbb{A}$

Output: $h, \alpha q \in \mathbb{A}[X]$

Requires: $\mathcal{S}(f) = \bigcup_{i=0}^{j} [c_i, d_i] \cup \bigcup_{i=j+1}^{k} [c_i, d_i] \cup \bigcup_{i=k+1}^{l} [c_i, d_i]$

Ensures: $\mathcal{S}(f + \alpha q \cdot h) \cap (d_j, c_{k+1})$ is empty

$\epsilon_1 := \frac{c_{j+1} - d_j}{2}$

$\epsilon_2 := \frac{c_{k+1} - d_k}{2}$

$h := (X - (d_j + \epsilon_1))(X - (c_{k+1} - \epsilon_2))$

$q := (X - d_j)^{\gamma_1}(X - c_{k+1})^{\gamma_2}$ where γ_1, γ_2 are defined as in equations (3.4)

$\alpha := 1$

for $i = j + 1$ *to* k **do**

  Let $\alpha_i := -\frac{\max_{x \in [c_i, d_i]}(f)}{\max_{x \in [c_i, d_i]}(q \cdot h)}$

  if $\alpha_i > \alpha$ **then**

    $\alpha := \alpha_i$

$\alpha := 1 + \alpha$

while $\mathcal{S}(f + \alpha q \cdot h) \cap ((d_j, d_j + \epsilon_1) \cup (c_{k+1} - \epsilon_2, c_{k+1})) = \emptyset$ **do**

  $\epsilon_1 := \frac{1}{2}\epsilon_1$

  $\epsilon_2 := \frac{1}{2}\epsilon_2$

return $h, \alpha q$

Finally, we consider another variation of the problem, where we want to remove the intervals from the semialgebraic set of f located to the left of the semialgebraic set of $\mathcal{S}(G)$[3].

$$\begin{aligned} \mathcal{S}(f) &= \bigcup_{i=0}^{j} [c_i, d_i] \cup \bigcup_{i=j+1}^{k} [c_i, d_i] \\ \mathcal{S}(G) &\cap \bigcup_{i=0}^{j} [c_i, d_i] = \emptyset \end{aligned} \tag{3.5}$$

The theoretical approach is the same as the previous problem. We formalize in Algorithm 11 a solution to this problem.

[3]The approach is symmetric in the sense that it can also address the case where the intervals to be removed are on the right of $\mathcal{S}(G)$

Algorithm 11: Removing redundant left strictly positive factors

RemoveStrictPosLeft(f)

Input: $f \in \mathbb{Q}[X], c_{j+1} \in \mathbb{A}$

Output: $h, \alpha q \in \mathbb{A}[X]$

Requires: $\mathcal{S}(f) = \bigcup_{i=0}^{j} [c_i, d_i] \cup \bigcup_{i=j+1}^{k} [c_i, d_i]$

Ensures: $\mathcal{S}(f + \alpha q \cdot h) \cap (-\infty, c_{j+1})$ is empty

$\epsilon := \frac{c_{j+1} - d_j}{2}$

$h := X - (c_{j+1} - \epsilon)$

if $\mathrm{ord}_{c_{j+1}}(f)$ *is even* **then**

  $q := (X - c_{j+1})^{\mathrm{ord}_{c_{j+1}}(f)+2}$

else

  $q := (X - c_{j+1})^{\mathrm{ord}_{c_{j+1}}(f)+1}$

$\alpha := 1$

for $i = 1$ *to* j **do**

  Let $\alpha_i := -\frac{\max_{x \in [c_i, d_i]}(f)}{\max_{x \in [c_i, d_i]}(q \cdot h)}$

  if $\alpha_i > \alpha$ **then**

    $\alpha := \alpha_i$

$\alpha := 1 + \alpha$

while $\mathcal{S}(f + \alpha q \cdot h) \cap (c_{k+1} - \epsilon, c_{k+1}) = \emptyset$ **do**

  $\epsilon := \frac{1}{2}\epsilon$

return $h, \alpha q$

Example 3.20 Let us consider $g_1 = (X+4)(X+3)(X+1)$ and $g_2 = -(X+2)(X-1)$. The semialgebraic set of $\mathcal{S}(g_1) = [-4, -3] \cup [1, \infty)$. We want to remove the redundant interval $[-4, -3]$ from $\mathcal{S}(g_1)$.

First, we fix $\epsilon = 1$ and set $h = X + 2, q = (X + 1)^2$. Then we compute α taking the ratio of the maximum values of g_1 and $q \cdot h$ over $[-4, -3]$. This is $\alpha = 1 -$

$\frac{\max_{x\in[-4,-3]}(f)}{\max_{x\in[-4,-3]}(q\cdot h)} = 1+\frac{1}{54}(10-7\sqrt{7})$. We can check that $f+\alpha q\cdot h = \frac{1}{54}((\sqrt{776-14\sqrt{7}}+\frac{(\frac{1029}{4}+\frac{1}{2}(\frac{111}{2}-21\sqrt{7}))X}{\sqrt{776-14\sqrt{7}}})^2+(118-7\sqrt{7}-\frac{(\frac{1029}{4}+\frac{1}{2}(\frac{111}{2}-21\sqrt{7}))^2}{776-14\sqrt{7}})X^2)\cdot(X+1)$ and $\mathcal{S}(f+\alpha q\cdot h)$ is $[-1,\infty)$.

3.3 Summary

The chapter discusses an algorithm for computing the compactification of a member in a quadratic module and its certificates, given that the certificates of the input polynomial are given. The method is a specialized and efficient version of the Basic Lemma applied to inputs f, $\epsilon - f$ for some $\epsilon > 0$, and a set of generators G.

We also developed the involution technique in Section 3.1.2. This simple observation allows us to reuse previous results about the unbounded left case to the unbounded right case. These symmetries are quite common in the thesis, and this result is useful to avoid repetitive proofs of similar cases.

Finally, in Section 3.2 we discussed two applications that were the motivation of the compactification procedure, to use the Basic Lemma in contexts that were not possible based on the requirements of the Lemma. The first computes certificates of nonnegative factors of a given polynomial. This technique can be seen as an analog to Theorem 2.20, which instead of taking a product, splits the factors of the given polynomial. In Section 3.2.2 we discuss the application of removing factors that introduce intervals that are not part of a given semialgebraic set. This problem is important and used extensively in Chapter 4 as we need to compute certificates of quadratic generators from the set of natural generators; in some cases these come from a single generator that includes additional intervals that need to be removed before we can apply the nonnegative factor certificate algorithm.

Chapter 4

Certificates Computation in Saturated Archimedean Quadratic Modules in $\mathbb{A}[X]$

Saturated quadratic modules have been extensively discussed in the literature [41, 42, 58] as a class of quadratic modules that contain a large number of polynomials, including non-negative polynomials over the semialgebraic set associated with the generators of the quadratic module. The results in the literature involve discussions about characterization, equivalence of an alternative set of generators, etc. However, not much attention is given to the certificate problem. This chapter aims to provide a solution for this problem. The algorithms developed in this chapter are fundamental in Chapter 5 and Chapter 6 of this thesis, as many problems in these chapters involve reductions to the saturated case.

This chapter proposes the SaturatedCert algorithm to compute a certificate for members in an Archimedean saturated quadratic module. Our approach lifts the certificates of an input polynomial f nonnegative over $\mathcal{S}(G)$ produced by Theo-

rem 2.12 from PO(Nat($\mathcal{S}(G)$)) to QM(G). In Section 4.1, we discuss an algorithmic description that allows us to obtain a certificate of the input polynomial f in PO(Nat($\mathcal{S}(G)$)). Section 4.2 focuses on the computation of the certificate for the products of natural generators, so we can lift the certificate in PO(Nat($\mathcal{S}(G)$)) from Section 4.1 to a certificate in QM(Nat($\mathcal{S}(G)$)). In Section 4.3 we describe a procedure to find a certificate in QM(G) of elements in Nat($\mathcal{S}(G)$); this allows us to lift the certificate from QM(Nat($\mathcal{S}(G)$)) to QM(G). Section 4.4 presents the SaturatedCert algorithm combining the results from previous sections in this chapter, discusses several examples, and compares the results with the Shang et. al's general method. Section 4.5 compares the implementation `CertSatQM` of the SaturatedCert algorithm with Shang et. al's general method for nonnegative polynomials. We compare our implementation `CertSatQM` with `RealCertify` and `MomentPolynomialOpt` for strictly positive polynomials. Finally, in Section 4.6 we conclude summarizing the contributions of this chapter.

4.1 Certificates of products in PO(Nat(S))

In Section 2.1.3, the proof of Theorem 2.12 provides a method to identify the nonnegative factors of a nonnegative polynomial over some semialgebraic set S. However, the goal of the theorem is to prove membership, so it does not indicate how to compute a certificate of a member in PO(Nat(S)). The remaining part to obtain a certificate requires a procedure to combine the certificate of the products of the nonnegative factors of the input polynomial. In this section, we provide a solution to this problem.

This method aims to clarify the subtly in handling the sums of squares in preordering representations, as these can be mistaken with generators of the structures. It is important to specify this operation, since some products might introduce square

terms that are not included in the generators associated with the preordering structure.

Proposition 4.1 *Let $\{g_1, \ldots, g_s\}$ be the set of natural generators of some semialgebraic set $S \subseteq \mathbb{R}$ and let $p = \sigma \cdot g_1^{a_1} \cdots g_s^{a_s}$, $q = \tau \cdot g_1^{b_1} \cdots g_s^{b_s}$ where σ, τ are sums of squares and $a_1, b_1, \ldots, a_s, b_s \in \{0, 1\}$. A certificate in $\mathrm{PO}(\mathrm{Nat}(S))$ of the product pq is computable.*

Proof. The product of the common generators of $\mathrm{PO}(\mathrm{Nat}(S))$ between p and q is $r := \prod_{i \in I} g_i$ where $I = \{i \mid 1 \leq i \leq s, a_i = b_i = 1\}$. Clearly, r divides both p and q; furthermore, $\frac{p}{r}$ and $\frac{q}{r}$ are products of generators of $\mathrm{PO}(\mathrm{Nat}(S))$ without common generators, so $\frac{p}{r}\frac{q}{r}$ is of the form $g_1^{c_1} \cdots g_s^{c_s}$ where $c_i = \max(a_i, b_i)$. We can verify that $pq = \sigma\tau r^2 \cdot g_1^{c_1} \cdots g_s^{c_s}$, which is a certificate of pq in $\mathrm{PO}(\mathrm{Nat}(S))$ since $\sigma\tau r^2$ is a non-negative polynomial that can be represented as sums of squares.

□

Corollary 4.2 *Let $G = \{g_1, \ldots, g_s\}$ be the set of natural generators of some set of generators. Let $p = \sum_{i \subseteq \{1,\ldots,s\}} \sigma_i \cdot g_1^{i_1} \cdots g_s^{i_s}$, $q = \sum_{j \subseteq \{1,\ldots,s\}} \sigma_j \cdot g_1^{j_1} \cdots g_s^{j_s} \in \mathrm{PO}(G)$ with $\sigma_i, \sigma_j \in \sum \mathbb{A}[X]^2$. A certificate of $pq \in \mathrm{PO}(G)$ is computable.*

Proof. We distribute the sums in the product of p and q and use Proposition 4.1 to compute the certificates for each of the products $\sigma_i \cdot g_1^{i_1} \cdots g_s^{i_s}$ and $\sigma_j \cdot g_1^{j_1} \cdots g_s^{j_s}$. □

By combining Theorem 2.12 and Corollary 4.2 we obtain an algorithm to find a certificate of f in $\mathrm{Pos}(\mathrm{Nat}(S))$. The construction iterates the inductive step of Theorem 2.12 on the input polynomial f mentioned above until it is a nonnegative polynomial over S and Corollary 4.2 allowing us to combine the certificates for each of the nonnegative factors of f.

The following example illustrates the subtlety that a certificate of a polynomial f in a preordering structure $\mathrm{PO}(G)$ need not serve as a certificate in a quadratic module $\mathrm{QM}(G)$, even though $\mathrm{QM}(G) = \mathrm{PO}(G)$. A certificate of a product of factors is computed by taking the product of certificates of individual factors in a preordering structure, whereas it need not work in the case of generating a certificate in the associated quadratic module structure because a preordering is closed under multiplication. Computing a certificate of a polynomial in a quadratic module requires generating certificates of products of natural generators as well.

Example 4.3 Consider the semialgebraic set $S_1 := [-1, 1] \cup \{2, 3\} \cup [4, 5]$. The natural generators of S_1 are $\mathrm{Nat}(S_1) = \{g_1 : X + 1, g_2 : (X - 1)(X - 2), g_3 : (X - 2)(X - 3), g_4 : (X - 3)(X - 4), g_5 : -(X - 5)\}$.

- Consider a polynomial $f_1 := (X + 2)(X - \frac{3}{2})(X - \frac{4}{3})$, which is nonnegative over S_1. Based on the construction suggested by the inductive proof in Theorem 2.12, f_1 is factored as $(X + 2)$ and $(X - \frac{3}{2})(X - \frac{4}{3})$ and a certificate of each factor is computed.

  - A certificate $X + 2$ is $1 + 1 \cdot g_1$
  - A certificate of $(X - \frac{3}{2})(X - \frac{4}{3})$ is $\frac{1}{6}((X - 1)^2 + 1) + \frac{5}{6} \cdot g_2$

  Hence, a certificate of f_1 is $\frac{1}{6}((X-1)^2+1)+\frac{5}{6} \cdot g_2+\frac{1}{6}((X-1)^2+1) \cdot g_1+\frac{5}{6} \cdot g_1 g_2$.

  Note that while the above certificates of $X + 2$ and $(X - \frac{3}{2})(X - \frac{4}{3})$ belong to $\mathrm{QM}(\mathrm{Nat}(S_1))$ as well as $\mathrm{PO}(\mathrm{Nat}(S_1))$; however, the above certificate of f is only for $\mathrm{PO}(\mathrm{Nat}(S_1))$ and not for $\mathrm{QM}(\mathrm{Nat}(S_1))$ since $(X + 1)(X - 1)(X - 2)$ is not a generator in $\mathrm{QM}(\mathrm{Nat}(S_1))$.

- Consider the polynomial $f_2 := (X - 1)(X - 2)^3(X - 3)^3(X - 4)$, which is nonnegative over S_1. We apply inductive Theorem 2.12 to obtain nonnegative

factors of f_2, which are $(X-1)(X-2), (X-2)(X-3)$ and $(X-3)(X-4)$. Hence, $f_2 = g_3^2 g_2 g_4$. Using Corollary 4.2 we obtain the certificate $f_2 = g_3^2 \cdot g_2 g_4 \in \mathrm{PO}(\mathrm{Nat}(S_1))$.

4.2 Certificate of products of natural generators

In Section 4.1, we described a procedure to compute a certificate of a nonnegative polynomial $f \in \mathbb{Q}[X]$ over a compact semialgebraic set S in $\mathrm{PO}(\mathrm{Nat}(S))$. To lift a certificate of f from $\mathrm{PO}(\mathrm{Nat}(S))$ to $\mathrm{QM}(\mathrm{Nat}(S))$, it is enough to find certificates of the products of natural generators in $\mathrm{QM}(\mathrm{Nat}(S))$; in particular, it is enough to have a certificate of all the products of two generators as we can iteratively multiply and distribute the certificates in $\mathrm{QM}(\mathrm{Nat}(S))$. We formalize this process in the following proposition.

Proposition 4.4 *Let $G = \{g_1, \ldots, g_s\}$ and $f \in \mathrm{PO}(G)$. Assume that we have certificates for each $g_i g_j$ in $\mathrm{QM}(G)$. If we have a certificate of f in $\mathrm{PO}(G)$, then a certificate of f in $\mathrm{QM}(G)$ is computable.*

Proof. We prove by induction on the operations of the preordering that a certificate of f is computable in $\mathrm{QM}(G)$.

- Base case: $f \in G$. A generator is its own certificate in $\mathrm{QM}(G)$.
- Inductive case: $f = f_1 + f_2$ for some $f_1, f_2 \in \mathrm{PO}(G)$. By the inductive hypothesis, if we have certificates of f_1, f_2 in $\mathrm{PO}(G)$, then we have certificates of f_1, f_2 in $\mathrm{QM}(G)$. Adding the certificates of f_1 and f_2 provides a certificate for f in $\mathrm{QM}(G)$.

- Inductive case: $f = f_1 f_2$ for some $f_1, f_2 \in \mathrm{PO}(G)$. By the inductive hypothesis, if we have certificates of f_1, f_2 in $\mathrm{PO}(G)$, then we have certificates of f_1, f_2 in $\mathrm{QM}(G)$, say $f_j = \sigma_{0,j} + \sum_{i=1}^{s} \sigma_{i,j} \cdot g_i$ for $j = 1, 2$. Let $g_0 := 1$. Multiplying f_1 and f_1 we obtain

$$f_i f_j = \sum_{i=0}^{s} \sum_{j=0}^{s} (\sigma_{i,1}\sigma_{j,2}) \cdot g_i g_j \tag{4.1}$$

Lifting the equation (4.1) with the certificates for each $g_i g_j$ in $\mathrm{QM}(G)$ we obtain a certificate for f in $\mathrm{QM}(G)$. □

In this section, all five cases of products of natural generators are considered one by one. By replacing X with $-X$, these cases reduce to the following three cases:

- Case 1: Product of a left natural generator $X - a_0$ and a right natural generator $-(X - b_k)$
- Case 2: Product of a left natural generator $X - a_0$ and a quadratic generator $(X - b_i)(X - a_{i+1})$
- Case 3: Product of two quadratic generators $(X - b_i)(X - a_{i+1})$ and $(X - b_j)(X - a_{j+1})$

For Case 1, we distinguish two cases. If $a_0 = b_k$, the certificate of the product is computed using the following identity $-(X-a_0)^2 = \frac{1}{4}(X-a_0-1)^2 \cdot (X-a_0) + \frac{1}{4}(X - a_0 + 1)^2 \cdot (-(X - a_0))$. If $a_0 < b_k$, the certificates are obtained using the following identity $(X-a_0)(-(X-b_k)) = \frac{1}{b_k - a_0}((X-b_k)^2 \cdot (X-a_0) + (X-a_0)^2 \cdot (-(X-b_k)))$.

Certificates of products of natural generators sharing common zeros of the form $(X - a_0), (X - a_0)(X - a_1)$ [Case 2] and $(X - b_i)(X - a_{i+1}), (X - a_{i+1})(X - a_{j+1})$ [Case 3] are already discussed in the proof of [41, Theorem 2.5]. In [42, Theorem 3.5], the authors proved that $\mathrm{QM}(\mathrm{Nat}(S)) = \mathrm{PO}(\mathrm{Nat}(S))$ when S is compact by showing

the membership of the products of $\mathrm{Nat}(S)$ in $\mathrm{QM}(\mathrm{Nat}(S))$. Thus, to complete the remaining cases, we turn the proof of [42, Theorem 3.5] into an algorithm to compute certificates for Case 2 and Case 3 when the natural generators do not share common zeros. We address each case separately in the following theorems.

Theorem 4.5 *Let* $g_i = X - a_0$ *and* $g_j = (X - b_i)(X - a_{i+1})$ *be the left natural generator and a quadratic generator of* $\mathrm{Nat}(S)$ *such that* $a_0 < b_i$*. There exists a certificate of* $g_i g_j$ *in* $\mathrm{QM}(g_i, g_j, -(X - b_k))$ *and is computable.*

Proof. Choose $\beta \in \mathbb{Q}$ such that $h := -(X - \beta)$ is strictly positive over S. We have $h \in \mathrm{QM}(\mathrm{Nat}(S))$ since $h = (\beta - b_k) + 1 \cdot (-(X - b_k))$.

Applying SOSBasicLemma with g_j and $g_i h$, we obtain $\sigma_1, \tau_1 \in \sum \mathbb{A}[X]^2$ such that $1 = \sigma_1 g_i h + \tau_1 g_j$. Hence, multiplying $g_i g_j h$ by both sides of the previous equation, we obtain the following.

$$\begin{aligned} g_i g_j h &= (g_i h)^2 \sigma_1 \cdot g_j + g_j^2 \tau_1 \cdot g_i h \\ &= (g_i h)^2 \sigma_1 \cdot g_j + g_j^2 \tau_1 \sigma_2 \cdot g_i + g_j^2 \tau_1 \tau_2 \cdot h \end{aligned} \tag{4.2}$$

by Case 1 for some $\sigma_2, \tau_2 \in \sum \mathbb{A}[X]^2$. Notice that $\mathcal{S}(g_i g_j h) = [a_0, b_i] \cup [a_{i+1}, \beta]$. Hence, we again use SOSBasicLemma with $g_i g_j$ and h to obtain the sums of squares σ_3, τ_3 such that $1 = \sigma_3 g_i g_j + \tau_3 h$. Hence, $g_i g_j = g_i^2 g_j^2 \sigma_3 + \tau_3 \cdot g_i g_j h$. Substituting the certificate of $g_i g_j h$ from the equation (4.2) in the previous equation, we obtain a certificate of $g_i g_j$ in $\mathrm{QM}(g_i, g_j, h) \subseteq \mathrm{QM}(g_i, g_j, -(X - b_k))$. □

Theorem 4.6 *Let* $g_i = (X - b_i)(X - a_{i+1})$ *and* $g_j = (X - b_j)(X - a_{j+1})$ *be two different quadratic generators of* $\mathrm{Nat}(S)$ *such that* $a_{i+1} < b_j$*. There exists a certificate of* $g_i g_j$ *in* $\mathrm{QM}(X - a_0, g_i, g_j, -(X - b_k))$ *and is computable.*

Proof. Choose $\alpha, \beta \in \mathbb{Q}$ such that $h_1 := X-\alpha$ and $h_2 := -(X-\beta)$ are strictly positive over S. Both h_1 and h_2 are members of $\mathrm{QM}(\mathrm{Nat}(S))$ since $h_1 = (a_0-\alpha)+1\cdot(X-a_0)$ and $h_2 = (\beta - b_k) + 1 \cdot (-(X - b_k))$.

We apply SOSBasicLemma with $h_1 g_i h_2$ and g_j to obtain $\sigma_1, \tau_1 \in \sum \mathbb{A}[X]^2$ such that $1 = \sigma_1 h_1 g_i h_2 + \tau_1 g_j$. Hence, multiplying $h_1 g_i g_j h_2$ by both sides of the previous equation, we obtain the following.

$$\begin{aligned} h_1 g_i g_j h_2 &= \sigma_1 (h_1 g_i h_2)^2 \cdot g_j + \tau_1 g_j^2 \cdot h_1 g_i h_2 \\ &= \sigma_1 (h_1 g_i h_2)^2 \cdot g_j + \tau_1 g_j^2 s_1 \cdot h_1 \\ &+ \tau_1 g_j^2 s_2 \cdot g_i + \tau_1 g_j^2 s_3 \cdot h_2 \end{aligned} \tag{4.3}$$

where the last equation is obtained from a similar derivation of the equation (4.2) for the polynomial $h_1 g_i h_2$. Finally, we apply SOSBasicLemma to $g_i g_j$ and $h_1 h_2$ to obtain $\sigma_2, \tau_2 \in \sum \mathbb{A}[X]^2$ such that $1 = \sigma_2 g_i g_j + \tau_2 h_1 h_2$. We obtain a certificate of $g_i g_j$ by multiplying $g_i g_j$ by both sides of the previous equation and substitute the certificate of $h_1 g_i g_j h_2$ in $\mathrm{QM}(h_1, g_i, g_j, h_2) \subseteq \mathrm{QM}(X - a_0, g_i, g_j, -(X - b_k))$. □

Example 4.7 Consider $G = \{g_1, g_2, g_3, g_4\}$ where $g_1 = X + 3$, $g_2 = (X + 2)(X + 1)$, $g_3 = (X - 1)(X - 2)$ and $g_4 = -(X - 3)$. We want to compute a certificate of $g_2 g_3$ in $\mathrm{QM}(G)$. We apply the procedure in the proof of Theorem 4.6 for the latter:

First, we compute the sums of squares σ_1, τ_1 for the identity $1 = \sigma_1 g_1 g_2 g_4 + \tau_1 g_3$. Using the SOSBasicLemma algorithm with inputs $g_1 g_2 g_4$ and g_3, we obtain the following:

$$\begin{aligned} \sigma_1 &= \tfrac{25651419}{1281797120}(-\tfrac{4415585}{34201892}X^3 - \tfrac{1251755}{102605676}X^2 + X - \tfrac{8011232}{76954257})^2 \\ &+ \tfrac{28331277347353}{2311869023304840}(-\tfrac{29384854743375}{14505614001844736}X^3 - \tfrac{890819840493969}{7252807000922368}X^2 + 1)^2 \\ &+ \tfrac{1994111861811973593029}{1859325425139625729196032 0}(-\tfrac{12854830761748593683 25}{3988223723623947186058}X^3 + X^2)^2 \\ &+ \tfrac{10676681630420278749526 1701}{3271739957028384938317599 0738944}(X^3)^2 + \tfrac{2157}{1595974}((X-1)(X-2))^2 \end{aligned}$$

$$\begin{aligned}
&\tau_1 = \tfrac{697}{2002808}((X+3)(X+2)(X+1)(X-3))^2 \\
&+ \tfrac{7224817}{31919480}(-\tfrac{15161753}{693582432}X^3 - \tfrac{21619799}{231194144}X^2 + \tfrac{7181883}{28899268}X + 1)^2 \\
&+ \tfrac{174487837272007}{1844899213881280}(-\tfrac{302420779830331}{2791805396352112}X^3 + \tfrac{719486242613227}{8375416189056336}X^2 + X)^2 \\
&+ \tfrac{138684576679824814487417}{10265814894269181535690752 0}(\tfrac{428504030089969982462 61}{138684576679824814487417}X^3 + X^2)^2 \\
&+ \tfrac{18295654272054030313659730 9}{39840656144761211125813354648440}(X^3)^2
\end{aligned}$$

Then, using these polynomials, we obtain a certificate of $g_1g_2g_3g_4$ in QM(g_3, $g_1g_2g_4$) as $g_1g_2g_3g_4 = \sigma_1(g_1g_2g_4)^2 \cdot g_3 + \tau_1 g_3^2 \cdot g_1g_2g_4$. In order to lift this certificate to QM(G), we compute two sums of squares σ_2, τ_2 using SOSBasicLemma with inputs g_1g_4 and g_2 such that $1 = \sigma_2 g_1g_4 + \tau_2 g_2$. We obtain

$$\sigma_2 = \tfrac{5077893}{74983750}(-\tfrac{1210241}{3385262}X + 1)^2 + \tfrac{51983202049}{253839639492500}X^2$$

$$\tau_2 = \tfrac{29282713}{149967500}(-\tfrac{22495125}{117130852}X + 1)^2 + \tfrac{116498980322719}{70263284189240000}X^2$$

Hence, $g_1g_2g_4 = \sigma_2(g_1g_4)^2 \cdot g_2 + \tau_2 g_2^2 \cdot g_1g_4 = \sigma_2(g_1g_4)^2 \cdot g_2 + \frac{1}{6}\tau_2(g_4g_2)^2 \cdot g_1 + \frac{1}{6}\tau_2(g_1g_2)^2 \cdot g_4$. The last equation follows from applying Case 1 in this section. Now, we compute the sums of squares σ_3, τ_3 for the identity $1 = \sigma_3 g_1g_4 + \tau_3 g_2g_3$. Using the SOSBasicLemma algorithm with inputs g_1g_4 and g_2g_3, we obtain $\sigma_3 = \frac{1}{40}X^2 + \frac{1}{10}, \tau_3 = \frac{1}{40}$

Thus, multiplying g_2g_3 by the identity $1 = \sigma_3 g_1g_4 + \tau_3 g_2g_3$, we obtain $g_2g_3 = \sigma_3 \cdot g_1g_2g_3g_4 + \tau_3(g_2g_3)^2$. Substituting the certificate of $g_1g_2g_3g_4$ obtained before in the last equation, we obtain a certificate of g_2g_3 in QM(G), i.e.,

$$\begin{aligned}
g_2g_3 = \tau_3(g_2g_3)^2 &+ \frac{1}{6}\sigma_3\tau_1 g_3^2\tau_2(g_2g_4)^2 \cdot g_1 + \sigma_3\tau_1 g_3^2\sigma_2(g_1g_4)^2 \cdot g_2 + \sigma_3\sigma_1(g_1g_2g_4)^2 \cdot g_3 \\
&+ \frac{1}{6}\sigma_3\tau_1 g_3^2\tau_2(g_1g_2)^2 \cdot g_4
\end{aligned}$$

4.3 Certificates of natural generators in terms of a set of generators G

Although natural generators belong to any univariate saturated quadratic module, these might not be included in a given set of generators. This section addresses the problem of computing certificates of natural generators using elements of the original set of generators.

The factors of the natural generators of saturated quadratic modules belong to the factors of the original generators G. This follows as a consequence of Theorem 2.9. From this observation, we develop a method based on the Basic Lemma (Theorem 2.19 and Theorem 2.20) to compute a certificate of nonnegative factors from the given generators. This method computes certificates for the left natural generator $X - a_0$, right natural generator $-(X - b_k)$, and quadratic generator $(X - b_i)(X - a_{i+1})$ that belong to a single generator. When the factors of quadratic generators belong to two different generators in G, we apply a procedure to compute a certificate involving these two generators.

Motivated by the above construction, in this section we describe a procedure to compute certificates nonnegative factors from the original generators. Then, we develop a procedure to compute a certificate of a quadratic generator $(X - b_i)(X - a_{i+1})$ using the certificates of nonnegative factors $(X - b_i)(X - c_{i_1})^{m_{i_1}}$ and $(X - c_{i_2})^{m_{i_2}}(X - a_{i+1})$ from two generators $g_1, g_2 \in G$. Finally, we use these constructions to provide an algorithm to compute certificates of natural generators in terms of a given set of generators G.

4.3.1 Certificates of a left/right natural generator

We compute a certificate for the left natural generator $X - a_0$. Using Theorem 2.9, we know that there exists a polynomial $g \in G$ such that $g(a_0) = 0$ and $\frac{dg}{dX}(a_0) > 0$, that is, $g = (X - a_0)\tilde{g}$ with $\tilde{g}$ is not divisible by $X - a_0$.

We can assume $X - a_0$ is nonnegative over $\mathcal{S}(g)$, as otherwise we use the RemoveStrictPosLeft algorithm in Section 3.2.2 to find a polynomial $\hat{g}$ that preserves the order and sign condition at a_0 and compute its certificates in $\mathrm{QM}(G)$ such that $X - a_0$ is nonnegative over $\mathcal{S}(\hat{g})$. The procedure discussed in Section 3.2.1 allows us to compute a certificate of $X - a_0$ setting $t_1 = X - a_0$ and $t_2 = \frac{g}{t_1}$ in terms of $\mathrm{QM}(G)$. Similarly, the same procedure is used to compute a certificate for the right natural generator $-(X - b_k)$.

4.3.2 Certificates of a quadratic generator

First, we state the following lemma that we will use to describe a method to compute certificates for the quadratic generator $(X - b_i)(X - a_{i+1})$.

Lemma 4.8 *Let $a_0, b_i, c_{i_1}, c_{i_2}, a_{i+1}, b_k \in \mathbb{A}$ be such that $a_0 \leq b_i < c_{i_2} < c_{i_1} < a_{i+1} \leq b_k$ and m_{i_1}, m_{i_2} are odd numbers. We have $(X - b_i)(X - a_{i+1}) \in Q := \mathrm{QM}(X - a_0, (X - b_i)(X - c_{i_1})^{m_{i_1}}, (X - c_{i_2})^{m_{i_2}}(X - a_{i+1}), -(X - b_k))$ and its certificates in Q module are computable.*

Proof. Let g_{nat} denote the natural generator $(X - b_i)(X - a_{i+1}) \in \mathrm{Nat}(S)$ and let $G_{\mathrm{fix}} := \{X - a_0, (X - b_i)(X - c_{i_1})^{m_{i_1}}, (X - c_{i_2})^{m_{i_2}}(X - a_{i+1}), -(X - b_k)\}$. The following is a high-level structure of this proof.

(i) Find a polynomial $q \in \mathrm{QM}(G_{\mathrm{fix}})$ of the form $q = g_{\mathrm{nat}}(\alpha \cdot g_{\mathrm{nat}} + 1)$ with $\alpha \in \sum \mathbb{A}[X]^2$.

(ii) As g_{nat} is nonnegative over $\mathcal{S}(G_{\text{nat}})$, we use the BasicLemma algorithm with inputs $g_{\text{nat}}, \alpha \cdot g_{\text{nat}}+1$ and G_{fix}, compute strictly positive polynomials σ, τ over $\mathcal{S}(G_{\text{fix}})$ such that $1 = \sigma g_{\text{nat}} + \tau(\alpha \cdot g_{\text{nat}} + 1)$.

(iii) Compute a certificate for σ, τ, and $\sigma\tau$ in $\text{QM}(G_{\text{fix}})$ using the Certificate algorithm.

(iv) Using the previous two steps, compute a certificate for τg_{nat} and g_{nat} in $\text{QM}(G_{\text{fix}})$.

Consider

$$\begin{aligned} q_1 := {} & \alpha_1 g_{\text{nat}}^2 \cdot (X - a_0) \\ & + \alpha_2 (X - a_{i+1})^2 \cdot (X - b_i)(X - c_{i_1})^{m_{i_1}} \\ & + \alpha_3 (X - b_i)^2 \cdot (X - a_{i+1})(X - c_{i_2})^{m_{i_2}} \\ & + \alpha_1 g_{\text{nat}}^2 \cdot (-(X - b_k)) \end{aligned} \tag{4.4}$$

with $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{A}^+$, q in step 1 above is constructed as follows. We can assume $m_{j_1} = m_{j_2}$ as otherwise, α_2 and α_3 are replaced by $\alpha_2(X - c_{i_1})^{\max(m_{j_1}, m_{j_2}) - \min(m_{j_1}, m_{j_2})}$ and $\alpha_3(X - c_{i_2})^{\max(m_{j_1}, m_{j_2}) - \min(m_{j_1}, m_{j_2})}$, respectively, since m_{j_1}, m_{j_2} are odd numbers, q_1 can be rewritten as

$$\begin{aligned} q_1 = {} & g_{\text{nat}}(\alpha_1 \cdot g_{\text{nat}}((X - a_0) + (-(X - b_k))) \\ & + \alpha_2 \cdot (X - a_{i+1})(X - c_{i_1})^{m_{i_2}} + \alpha_3 \cdot (X - b_i)(X - c_{i_2})^{m_{i_2}}) \\ = {} & g_{\text{nat}}(\alpha_1 (b_k - a_0) \cdot g_{\text{nat}} \\ & + \alpha_2 \cdot (X - a_{i+1})(X - c_{i_1})^{m_{i_2}} + \alpha_3 \cdot (X - b_i)(X - c_{i_2})^{m_{i_2}}) \end{aligned}$$

Make α_2 be $\frac{1}{(b_i - a_{i+1})(b_i - c_{i_1})^{m_{i_2}}}$ and α_3 be $\frac{1}{(a_{i+1} - b_i)(a_{i+1} - c_{i_2})^{m_{i_2}}}$, which are positive constants, since $b_i < c_{i_1} \leq c_{i_2} < a_{i+1}$. By Lemma *A*.3 from below, $\alpha_2(X - b)(X - c_{j_2})^{m_{j_1}} + \alpha_3(X - a)(X - c_{j_1})^{m_{j_1}}$ is of the form $q_2 \cdot (X - a)(X - b) + 1$ for some $q_2 \in \sum \mathbb{A}[X]^2$. Hence, q_1 is simplified to

$$\begin{aligned} q_1 &= g_{\text{nat}}(\alpha_1(b_k - a_0) \cdot g_{\text{nat}} + q_2 \cdot g_{\text{nat}} + 1) \\ &= g_{\text{nat}}((\alpha_1(b_k - a_0) + q_2) \cdot g_{\text{nat}} + 1) \end{aligned} \tag{4.5}$$

The resulting polynomial q_1 is therefore the required polynomial q above.

Let $q_3 := \alpha_1(b_k - a_0) + q_2 \in \sum \mathbb{A}[X]^2$. Since g_{nat} and $q_3 \cdot g_{\text{nat}} + 1$ are relatively prime, we use BasicLemma with inputs $g_{\text{nat}}, q_3 \cdot g_{\text{nat}} + 1$ and G_{fix} to find σ, τ such that both are strictly positive over $\mathcal{S}(G_{\text{fix}})$ and

$$1 = \sigma g_{\text{nat}} + \tau(q_3 \cdot g_{\text{nat}} + 1) \tag{4.6}$$

Additionally, $\sigma\tau$ is also strictly positive over $\mathcal{S}(G_{\text{fix}})$. The certificates for σ, τ, and $\sigma\tau$ in $\text{QM}(G_{\text{fix}})$ are computed using the Certificate algorithm from [82].

We multiply τg_{nat} on both sides of equation (4.6) to obtain $\tau g_{\text{nat}} = g_{\text{nat}}^2 \cdot \sigma\tau + \tau^2 \cdot q$, for which the certificates are given explicitly. Multiplying both sides of equation (4.6) by g_{nat} we obtain $g_{\text{nat}} = g_{\text{nat}}^2\sigma + g_{\text{nat}}^2 q_3 \cdot \tau + \tau g_{\text{nat}}$, thus the certificates for g_{nat} are given explicitly in $\text{QM}(G_{\text{fix}})$. □

Lemma 4.9 *Let $a_0, b_i, c_{i_1}, c_{i_2}, a_{i+1}, b_k \in \mathbb{A}$ be such that $a_0 \leq b_i < c_{i_1} \leq c_{i_2} < a_{i+1} \leq b_k$ and m_{i_1}, m_{i_2} are odd numbers. We have $(X - b_i)(X - a_{i+1}) \in Q := \text{QM}(X - a_0, (X - b_i)(X - c_{i_1})^{m_{i_1}}, (X - c_{i_2})^{m_{i_2}}(X - a_{i+1}), -(X - b_k))$ and its certificates in Q are computable.*

Proof. Let $G_{\text{fix}} := \{X - a_0, (X - b_i)(X - c_{i_1})^{m_{i_1}}, (X - c_{i_2})^{m_{i_2}}(X - a_{i+1}), -(X - b_k)\}$. We use the results in Section 4.2 to find certificates for the product of $(X - b_i)(X - c_{i_1})$ and $(X - c_{i_2})(X - a_{i+1})$ in $\text{QM}(X - a_0, (X - b_i)(X - c_{i_1}), (X - c_{i_2})(X - a_{i+1}), -(X - b_k))$, i.e., $(X - b_i)(X - c_{i_1})(X - c_{i_2})(X - a_{i+1}) = \sigma_0 + \sigma_1 \cdot (X - a_0) + \sigma_2 \cdot (X - b_i)(X - c_{i_1}) + \sigma_3 \cdot (X - c_{i_2})(X - a_{i+1}) + \sigma_4 \cdot (-(X - b_k))$. Since m_1, m_2 are odd numbers,

we multiply the sums of squares $(X - c_{i_1})^{m_{i_1}-1}(X - c_{i_2})^{m_{i_2}-1}$ by the last equation; thus we obtain a certificate of $p := (X - b_i)(X - c_{i_1})^{m_{i_1}}(X - c_{i_2})^{m_{i_2}}(X - a_{i+1})$ in $\mathrm{QM}(G_{\mathrm{fix}})$.

Since we have certificates for the last expression, we use the RemoveStrictPosBetween algorithm to remove the interval $[c_{i_1}, c_{i_2}]$ from the semialgebraic set of p. Finally, we use the result of Section 3.2.1 to obtain a certificates of $(X - b_i)(X - a_{i+1})$ in $\mathrm{QM}(G_{\mathrm{fix}})$. □

By Theorem 2.9, we know that there exists a polynomial $g_{i_1} \in G$ such that $g_{i_1}(b_i) = 0$ and $\frac{dg_{i_1}}{dX}(b_i) < 0$, i.e., $g_{i_1} = (X - b_i)\tilde{g_{i_1}}$ where $\tilde{g_{i_1}}$ is not divisible by $X - b_i$. If $g_{i_1}(a_{i+1}) = 0$ and $\frac{dg_{i_1}}{dX}(a_{i+1}) > 0$ then $g_{i_1} = (X - b_i)(X - a_{i+1})t$ and t are not divisible by $X - a_{i+1}$. We can assume $(X - b_i)(X - a_{i+1})$ is nonnegative over $\mathcal{S}(g_{i_1})$, as otherwise we use the RemoveStrictPosBetween algorithm to find a polynomial that preserves the sign and order conditions at b_i and a_{i+1}. Hence, we use the procedure discussed in Section 3.2.1 to compute a certificate of $(X - b_i)(X - a_{i+1})$ by setting $t_1 = (X - b_i)(X - a_{i+1})$ and $t_2 = \frac{g_{i_1}}{t_1}$.

If $g_{i_1}(a_i) \neq 0$ or $\frac{dg_{i_1}}{dX}(a_{i+1}) = 0$ then let c_{i_1} be the real root of g_{i_1} closest to the right[1] of b_i and $m_{i_1} := \mathrm{ord}_{c_{i_1}}(g_{i_1})$. We find that $(X - b_i)(X - c_{i_1})^{m_{i_1}}$ divides g_{i_1} and $b_i < c_{i_1} \leq a_{i+1}$. If m_{i_1} is even, we use Proposition $A.4$ to compute a certificates of $(X - b_i)(X - c_{i_1})^{m_{i_1}+1}$ in $\mathrm{QM}(\mathrm{compact}_G(g_{i_1})) \subseteq \mathrm{QM}(G)$. On the other hand, if m_{i_1} is odd, we use the procedure discussed in Section 3.2.1, we obtain a certificate of $(X - b_i)(X - c_{i_1})^{m_{i_1}}$ in $\mathrm{QM}(\mathrm{compact}_G(g_{i_1})) \subseteq \mathrm{QM}(G)$.

Similarly, by Theorem 2.9, there exists a polynomial $g_{i_2} \in G$ such that $g_{i_2}(a_{i+1}) = 0$ and $\frac{dg_{i_2}}{dX}(a_{i+1}) > 0$. Repeating the steps above to g_{i_2} we find certificates for $(X - c_{i_2})^{m_{i_2}}(X - a_{i+1})$ in $\mathrm{QM}(G)$ where c_{i_2} is the real root of g_{i_2} closest to the left of a_{i+1} and m_{i_2} is odd.

[1] i.e., $c_{i_1} - b_i$ is minimal and positive.

Finally, if $c_{i_2} < c_{i_1}$ we use Lemma 4.8 to obtain a certificate of $(X-b_i)(X-a_{i+1})$ in $\mathrm{QM}(X-a_0, g_{i_1}, g_{i_2}, -(X-b_k))$. Otherwise, we use Lemma 4.9 to obtain a certificate of $(X-b_i)(X-a_{i+1})$ in $\mathrm{QM}(X-a_0, g_{i_1}, g_{i_2}, -(X-b_k))$.

4.3.3 Certificates of -1

If the semialgebraic set associated with the set of generators is empty, by definition, the set of natural generators is $\{-1\}$. As the Certificate algorithm excludes the case when $\mathcal{S}(G)$ is empty, we modify the core result in Lemma 2.22 to compute a certificate of -1 to use this construction.

In the proof of Lemma 2.22, the author proves the nonnegativity of $f-g$ over B for a particular $g \in \mathrm{QM}(G)$ constructed in the proof considering two cases, on $B \cap S_1$ and $B \cap S_1^c$ where $S_1 := \{x \in \mathbb{R} \mid g_i(x) + 2\epsilon \geq 0 \text{ for } g_i \in G\}$. For the former case, the author uses the constant $\mu := \min\{f(x) \mid x \in B \cap S_1\}$. If B is empty, then the certificate is trivially a sequence of 0's as sums of squares multipliers. However, if both B and S_1 are non-empty, the construction requires that μ is positive; nonetheless, for the case of an input polynomial to be -1, this cannot be the case. To address this issue, we find an $\epsilon > 0$ such that S_1 is empty. In doing so, the proof of the nonnegativity of $f-g$ for the g constructed is based only on the second case. We construct such an ϵ using the following proposition:

Proposition 4.10 *Let $G = \{g_1, \ldots, g_s\}$ be such that $\mathcal{S}(G)$ is empty. There exists $\epsilon > 0$ such that $\{x \in \mathbb{R} \mid g_i(x) + \epsilon \geq 0, g_i \in G\}$ is empty.*

Proof. Consider the case where G is a singleton. In this case, it suffices to find the maximum value v of $g_1 \in G$ over $\mathbb{R}$ which is negative. Then set $\epsilon := -\frac{v}{2}$.

When G has more than one generator and contains a generator g such that $\mathcal{S}(g)$ is empty, we apply the previous construction. Otherwise, we apply the following

construction. Let $s(x) := \min_{g_i \in G}(g_i(x))$. As the g_i's are polynomials, the function s is continuous. Since $\mathcal{S}(G)$ is empty, we see that $\mathcal{S}(s)$ is also empty. Hence, we find the maximum value v of s over $\mathbb{R}$. Finally, we choose $\epsilon := -\frac{v}{2}$ as in the previous case. We prove that this choice of ϵ makes $\{x \in \mathbb{R} \mid g_i(x) + \epsilon \geq 0, g_i \in G\}$ empty. Otherwise, there exists a point x^* such that all $g_i \in G$ satisfy $g_i(x^*) + \epsilon \geq 0$. In particular, let $g \in G$ be the polynomial such that $s(x^*) = g(x^*)$, then $0 \leq g(x^*) - \frac{v}{2} = g(x^*) - \frac{\max_{x \in \mathbb{R}}(s(x))}{2} \leq g(x^*) - \frac{s(x^*)}{2} \leq \frac{g(x^*)}{2}$, which implies that $g(x^*) \geq 0$. As g evaluates to a minimum value at x^* then for every $g_i \in G$ we have $g_i(x^*) \geq 0$. This means that $x^* \in \mathcal{S}(G)$, which contradicts the assumption that $\mathcal{S}(G)$ is empty. $\square$

Example 4.11 Let $G := \{g_1, g_2\}$ where $g_1 := (X+2)(X+1)(X-3)$ and $g_2 := -(X+3)(X-1)(X-2)$. We can check $\mathcal{S}(g_1, g_2)$ is empty, so -1 belongs to $\mathrm{QM}(g_1, g_2)$. We use the Certificate algorithm to compute its certificate as follows:

1. We construct the polynomial g in line 11 as $g = X^2 \cdot g_1 + (X + \frac{1}{2})^2 \cdot g_2 = -\frac{3}{2} - \frac{17}{4}X - 5X^2 - \frac{1}{4}X^3 - X^4$.
2. The output in line 12 of the Certificate algorithm is $(0, 0)$ as $\mathcal{S}(g)$ is empty.
3. We skip lines 14, 15 as $\tilde{f} = -1$ is a global lower bound.
4. We use the construction in Proposition 4.10 to determine the value of ϵ. We find that the maximum value of g is ≈ -0.304514. We can set a rational upper bound for ϵ with $\frac{1}{7}$.
5. The output in line 16 is $\left(\frac{g-\gamma}{\gamma+\epsilon}\right)^{2N}$ where $\gamma = 0$, $\epsilon = \frac{1}{7}$, and $N = 1$.
6. The final certificate of -1 is $s_0 + s_1 \cdot g_1 + s_2 \cdot g_2$ where
   - $s_0 = \frac{9354955}{416}(-\frac{1156467}{74839640}X^6 - \frac{11666252}{84194595}X^5 + \frac{32122727}{112259460}X^4 + X^3 + \frac{21152573}{37419820}X^2 + \frac{180544}{9354955}X - \frac{1612871}{74839640})^2$

$$+\frac{59714904742547}{4789736960}(\frac{22810018055003}{1074868285365846}X^6-\frac{46756167384452}{537434142682923}X^5-\frac{1488158656563691}{2328881284959333}X^4+X^2+\frac{32795486806654}{59714904742547}X+\frac{7970520925719}{119429809485094})^2$$

$$+\frac{4966424403127192392077}{1453221921814623792}(-\frac{7777818559142431122349}{1489927320938157717623 10}X^6+\frac{24240888766648224613917}{7946279045003507827 3232}X^5+X^4+\frac{21538709760036419911935}{7946279045003507827 3232}X+\frac{2960623920346085779071}{4966424403127192392 0770})^2$$

$$+\frac{2180029993675121552519735293771}{2746234037953212305122897920}(\frac{39270690249753776441050901 9384}{6540089981025364657559205881313}X^6+X^5-\frac{96623622422576792936910195079 95}{2834038991777658018275655881902 3}X-\frac{24725022570906394311746983 1784}{2180029993675121552519735293771})^2$$

$$+\frac{9625551749055578259535197731003422 1117}{2357920441159011471205345693742713 60}(-\frac{1420817574435773827888891132456928 8535}{1155066209886669391144223727720410653404}X^6+X+\frac{1596989178347036356874166270603121 76103}{3850220699622231303814079092401368844 68})^2$$

$$+\frac{3087183576236762022811917310528196400094274 4323}{1197572646410498824738331160900521765449267200}(X^6+\frac{3850696612177232367751391010025291932565526 0187}{4013338649107790629655492503686655320122556761 99})^2$$

$$+\frac{1454956884222567830200409003068231298025163941 63473}{2608670121920063909276070127396325958079661895 2935}$$

- $s_1 = \frac{49}{16}(X(6+17X+20X^2+X^3+4X^4))^2$
- $s_2 = \frac{49}{16}((X+\frac{1}{2})(6+17X+20X^2+X^3+4X^4))^2$

Example 4.12 Consider $G = \{g_1, g_2\}$ where $g_1 = -(X+1)^3$ and $g_2 = (X-1)^3$. We use the Certificate algorithm to compute the certificate of -1 in $\mathrm{QM}(G)$ as follows:

1. We construct the polynomial g in line 11 as $g = X^2 \cdot g_1 + (X-1)^2 \cdot g_2 = -1 + 5X - 11X^2 + 7X^3 - 8X^4$
2. The output in line 12 of the algorithm is $(0, 0)$ as $\mathcal{S}(g)$ is empty.
3. We skip lines 14, 15 as $\tilde{f} = -1$ is a global lower bound.
4. We use the construction in Proposition 4.10 to determine the value of ϵ. We find that the maximum value of g is ≈ -0.178296. We can set a rational upper bound for ϵ with $\frac{1}{11}$.
5. The output in line 16 is $\left(\frac{g-\gamma}{\gamma+\epsilon}\right)^{2N}$ where $\gamma = 0, \epsilon = \frac{1}{11}$ and $N = 1$.
6. The final certificate of -1 is $s_0 + s_1 \cdot g_1 + s_2 \cdot g_2$ where

- $s_0 = \frac{4605827}{4}(-\frac{3148937}{46058270}X^6 - \frac{3275261}{9211654}X^5 + X^4 - \frac{6500446}{13817481}X^3 - \frac{187488}{4605827}X^2 + \frac{208644}{4605827}X - \frac{41431}{9211654})^2$
  $+ \frac{80153887766729}{165809772}(\frac{115955119397469}{801538877667290}X^6 - \frac{58902831240816}{80153887766729}X^5 + X^3 - \frac{85567160304819}{160307775533458}X^2 + \frac{5636029678173}{80153887766729}X + \frac{80738990799}{160307775533458})^2$
  $+ \frac{953312152029806020477}{6412311021338320}(-\frac{371343945609790627023}{953312152029806020477}X^6 + X^5 - \frac{804162213785858755960}{2859936456089418061431}X^2 + \frac{116332308447982838350}{953312152029806020477}X - \frac{121021981473331111 5}{953312152029806020477})^2$
  $+ \frac{364467608257711287422573909}{152529944324768963276 32}(X^6 - \frac{646341734779287595171422841}{182233804128855643711286954 5}X^2 + \frac{107382294982931221244015419 6}{546701412386566931133860863 5}X - \frac{440978559380237477174540 23}{182233804128855643711286954 5})^2$
  $+ \frac{149593553866392087192725881446553 1}{164010423715970079340158259050}(X^2 - \frac{729100608484334409365267065206087}{149593553866392087192725881446553 1}X + \frac{400382022843033234799156015880157}{598374215465568348770903525786212 4})^2$
  $+ \frac{427566314103156455375715267016989334 87}{269268396959505756946906586603795580}(X - \frac{262158882419817866336148096047484119 79}{855132628206312910751430534033978669 74})^2$
  $+ \frac{196538059267229981573305375471795497449 01}{684106102565050328601144427227182935792 0}$
- $s_1 = 121(X((-1+X)^5 - X^2(1+X)^3))^2$
- $s_2 = 121((X-1)((-1+X)^5 - X^2(1+X)^3))^2$

4.4 Certificates computation with SaturatedCert

This section describes SaturatedCert in Algorithm 12 by combining results from previous sections in this chapter. In addition, we discuss examples to illustrate the method.

Algorithm 12: Algorithm to compute a certificate of a member in Archimedean saturated quadratic modules

SaturatedCert(f, G)

Input: $f \in \mathbb{A}[X], G = \{g_1, \ldots, g_s\} \subseteq \mathbb{A}[X]$

Output: $\tau_0^{'}, \tau_1^{'}, \ldots, \tau_s^{'}$

Requires: $\mathrm{QM}(G)$ is Archimedean saturated and $f \in \mathrm{QM}(G)$

Ensures: $f = \tau_0^{'} + \sum_{i=1}^{s} \tau_i^{'} \cdot g_i$ and each $\tau_i^{'}$ is a sums of squares

1 Let $S := \mathcal{S}(G) = \bigcup_{i=0}^{k} [a_i, b_i]$

2 Let $\mathrm{Nat}(S) = \{n_1, \ldots, n_t\}$

3 Obtain a certificate of $f = \sum_{i \subseteq \{1,\ldots,t\}} \sigma_i \cdot n_1^{i_1} \cdots n_t^{i_t}$ in $\mathrm{PO}(\mathrm{Nat}(S))$ using the construction in Section 4.1

4 **for** *each* $\sigma_i \neq 0$ **do**

5 Find the certificate in $\mathrm{QM}(\mathrm{Nat}(S))$ of $n_1^{i_1} \cdots n_t^{i_t} = \sigma_{i,0} + \sum_{j=1}^{t} \sigma_{i,j} \cdot n_j$ using the certificate of products of natural generators from Section 4.2.

6 Lift the certificate of f from $\mathrm{PO}(\mathrm{Nat}(S))$ to $\mathrm{QM}(\mathrm{Nat}(S))$ using the certificates for each $g_1^{i_1} \cdots g_s^{i_s}$, i.e., $f = \tau_0 + \sum_{i=1}^{t} \tau_i \cdot n_i$

7 Compute a certificates in $\mathrm{QM}(G)$ of the natural generators using the construction in Section 4.3, i.e, $n_i = \tau_{i,0}^{'} + \sum_{j=1}^{s} \tau_{i,j}^{'} \cdot g_j$.

8 Lift the certificate of f from $\mathrm{QM}(\mathrm{Nat}(S))$ to $\mathrm{QM}(G)$ using the certificates of each natural generator from the previous step, i.e., $f = \tau_0^{'} + \sum_{i=1}^{s} \tau_i^{'} \cdot g_i$.

9 **return** $\tau_0^{'}, \ldots, \tau_s^{'}$

4.4.1 Examples

In this section, we discuss a simple example to highlight the key steps of our approach and a comparison of the results with Shang et. al's general method.

A simple example

Consider the generators $G := \{g_1, g_2\}$, where $g_1 := (X+3)(X+2)(X-1)$ and $g_2 := -(X+1)(X-2)(X-3)$. The semialgebraic set of G is $S = [-3,-2] \cup [2,3]$. We will compute a certificate for the polynomial $(X+3)$ which is the left natural generator of $\mathrm{Nat}(S)$.

We compute a certificate for $X+3$ by splitting the generator g_1. The semialgebraic set of g_1 is unbounded, thus we apply Algorithm 7 to compute the sums of squares σ_1 and τ_1 and an auxiliary polynomial h_1 such that $1 = \sigma_1 g_1 + \tau_1 h_1$. We obtain the following.

- $\sigma_1 := \frac{1}{2880}$
- $\tau_1 := \frac{1}{2880}((X+\frac{17}{2})^2 + \frac{599}{4})$
- $h_1 := -(X-13)$

The semialgebraic set of $g_1 h_1$ is bounded. We can check $g_1 h_1 = \sigma_1 g_1^2 \cdot h_1 + \tau_1 h_1^2 \cdot g_1$. Since h_1 is strictly positive over $\mathcal{S}(G)$, we find certificates in $\mathrm{QM}(G)$ using the algorithm of Section 2.1.9; $h_1 := s_0 + s_1 \cdot g_1 + s_2 \cdot g_2$ where $s_1 := \frac{589}{526513}X^2$, $s_2 := \frac{589}{526513}(X+5)^2$ and $s_0 := h_1 - s_1 \cdot g_1 - s_2 \cdot g_2$.

Therefore, a certificate of $g_1 h_1$ in $\mathrm{QM}(G)$ is

$$g_1 h_1 = \sigma_1 g_1^2 s_0 + (\sigma_1 g_1^2 s_1 + \tau_1 h_1^2) \cdot g_1 + \sigma_1 g_1^2 s_2 \cdot g_2$$

Next, using BasicLemma with inputs $X+3$, $\frac{g_1}{X+3}h_1$ and $\{g_1h_1\}$, we find σ_2 and τ_2 satisfying $\sigma_2(X+3)+\tau_2\frac{g_1}{X+3}h_1$:

- $\sigma_2 := \frac{2543783}{78097856}X^2 - \frac{16649805}{78097856}X + \frac{16870389}{39048928} - \frac{27573}{19524464}X^3 - \frac{232}{1102409}$ $(\frac{2543783}{78097856}X^2 - \frac{16649805}{78097856}X + \frac{16870389}{39048928} - \frac{27573}{19524464}X^3)(X+3)(2 - \frac{232}{1102409}(X+3)(-X^3+12X^2+15X-26)+(1-\frac{232}{1102409}(X+3)(-X^3+12X^2+15X-26))^2+(1-\frac{232}{1102409}(X+3)(-X^3+12X^2+15X-26))^3+(1-\frac{232}{1102409}(X+3)(-X^3+12X^2+15X-26))^4+(1-\frac{232}{1102409}(X+3)(-X^3+12X^2+15X-26))^5)(-X^3+12X^2+15X-26)$
- $\tau_2 := \frac{889403}{78097856} - \frac{27573}{19524464}X + \frac{232}{1102409}(\frac{2543783}{78097856}X^2 - \frac{16649805}{78097856}X + \frac{16870389}{39048928} - \frac{27573}{19524464}X^3)(X+3)^2(2-\frac{232}{1102409}(X+3)(-X^3+12X^2+15X-26)+(1-\frac{232}{1102409}(X+3)(-X^3+12X^2+15X-26))^2+(1-\frac{232}{1102409}(X+3)(-X^3+12X^2+15X-26))^3+(1-\frac{232}{1102409}(X+3)(-X^3+12X^2+15X-26))^4+(1-\frac{232}{1102409}(X+3)(-X^3+12X^2+15X-26))^5)$

We can check $X+3 = (X+3)^2\sigma_2 + \tau_2 \cdot g_1h_1$. Since σ_2 and τ_2 are strictly positive polynomials over $\mathcal{S}(g_1h_1)$, we compute a certificate of these in $\mathrm{QM}(g_1h_1)$. Using a square free decomposition, we notice that $\sigma_2 := \sigma_{2_{sos}}\sigma_{2_{rest}}$ where $\sigma_{2_{sos}} := (232X^4 - 2088X^3 - 11832X^2 - 4408X + 1120505)^6 \in \sum \mathbb{Q}[X]^2$ and

$$\begin{aligned}\sigma_{2_{rest}} := &-\frac{3939}{5006538738653003018117160074353414225724432}X^3 \\ &+\frac{21023}{1158537889936232103365954397371038002646976}X^2 \\ &-\frac{16649805}{1401830846822840845072804820818955983202840 96}X \\ &+\frac{16870389}{7009154234114204225364024104094779916014 2048}\end{aligned}$$

is strictly positive over $\mathcal{S}(g_1h_1)$. We find certificates for the strictly positive polynomial σ_2 using the Certificate algorithm from [82]; $\sigma_2 := \sigma_{2_{sos}}s_3 + \sigma_{2_{sos}}s_4 \cdot g_1h_1$ where:

- $s_3 := \sigma_{2_{rest}} - s_4 \cdot g_1 h_1$,
- $s_4 := \frac{1}{11077069702969986520794623280923002600254 1}$
  $+ \frac{1}{75725680710000000000}(\frac{519162}{1690512571} g_1 h_1 - \frac{1111230756}{1690512571})^{116}$

Using the Certificate algorithm, we also find $\tau_2 = s_5 + s_6 \cdot g_1 h_1$ where

- $s_5 := \tau_2 - s_6 \cdot g_1 h_1$,
- $s_6 := \frac{1}{3057473673129919885032630345074 9}(\frac{2999}{1000} + X)^{22}$
  $+ 33637053286776059147144239790568790458376494 5 \ldots$
  $\ldots 92119310208217769662829702462177034 91(\frac{57575}{240688632} g_1 h_1 - \frac{20539225}{40114772})^{304}$

Finally, the certificate of $X + 3$ in $\mathrm{QM}(G)$ is:

$$\begin{aligned}
X + 3 &= (X+3)^2 \sigma_2 + \tau_2 \cdot g_1 h_1 \\
&= (X+3)^2 (\sigma_{2_{sos}} s_3 + \sigma_{2_{sos}} s_4 \cdot g_1 h_1) + (s_5 + s_6 \cdot g_1 h_1) g_1 h_1 \\
&= ((X+3)^2 \sigma_{2_{sos}} s_3 + s_6 (g_1 h_1)^2) + ((X+3)^2 \sigma_{2_{sos}} s_4 + s_5) \cdot g_1 h_1 \\
&= ((X+3)^2 \sigma_{2_{sos}} s_3 + s_6 (g_1 h_1)^2) + ((X+3)^2 \sigma_{2_{sos}} s_4 + s_5) \sigma_1 g_1^2 s_0 \\
&\quad + ((X+3)^2 \sigma_{2_{sos}} s_4 + s_5)(\sigma_1 g_1^2 s_1 + \tau_1 h_1^2) \cdot g_1 \\
&\quad + ((X+3)^2 \sigma_{2_{sos}} s_4 + s_5) \sigma_1 g_1^2 s_2 \cdot g_2
\end{aligned}$$

Comparison with an alternative approaches

In this section, we compare the certificates produced by our SaturatedCert algorithm, the algorithm by Shang et al, and the Extended Gram matrix approach. For the latter, we consider $G = \{g_1\}$ where $g_1 := -(X+2)(X+1)(X-1)(X-3)$ and compute a certificate for different input polynomials.

Example 4.13 Let the input polynomial be the left natural generator $X + 2$.

- SaturatedCert: We use SOSBasicLemma with inputs $X + 2$ and $-(X + 1)(X - 1)(X - 3)$ to obtain sums of squares σ, τ such that $1 = \sigma(X + 2) + \tau(-(X + 1)(X - 1)(X - 3))$ where

  - $\sigma = \frac{3}{5}(-\frac{5}{18}X + 1)^2 + \frac{11}{540}(X)^2$
  - $\tau = \frac{1}{15}$

  Hence, we obtain the certificate $X+2 = s_0+\frac{1}{15}\cdot g_1$ where $s_0 = \frac{3}{5}((X+2)(-\frac{5}{18}X+1))^2 + \frac{11}{540}((X + 2)X)^2$. In expanded form, the degrees of the certificates are $\deg(s_0) = 4$ and $\deg(s_1) = 0$.

- Shang et. al's general method: First, we fix the input polynomial f at $x = -2$ with $f_1 := f - \frac{1}{15}\frac{(X-(-1))^2}{(-2-(-1))^2}\frac{(X-1)^2}{(-2-1)^2}\frac{(X-3)^2}{(-2-3)^2} \cdot g_1$ which is a sums of squares, i.e.,

$$
\begin{aligned}
f_1 = {} & \frac{63}{125}((X+2)(\frac{521}{559872}X^4 + \frac{13513}{435456}X^3 - \frac{61819}{483840}X^2 - \frac{16}{63}X + 1))^2 \\
& + \frac{2153239}{10080000}((X+2)(\frac{39646495}{2092948308}X^4 - \frac{3092335}{38758302}X^3 - \frac{558449}{2153239}X^2 + X))^2 \\
& + \frac{21390631747541989}{428635813478400000}((X+2) \\
& (\frac{21685151003276690}{577547057183633703}X^4 - \frac{25691071038847010}{64171895242625967}X^3 + X^2))^2 \\
& + \frac{202747529142388112939383}{898201183331987134905600000}((X+2) \\
& (-\frac{923471009955226677313145}{364945552456298603290889 4}X^4 + X^3))^2 \\
& + \frac{595737239839081188872948694 91}{137917887532748647844631860355840 00}((X+2)(X^4))^2
\end{aligned}
$$

  Hence, we obtain the certificates $X + 2 = s_2 + s_3 \cdot g_1$ where

  - $s_2 = f_1$
  - $s_3 = \frac{1}{15}\frac{(X-(-1))^2}{(-2-(-1))^2}\frac{(X-1)^2}{(-2-1)^2}\frac{(X-3)^2}{(-2-3)^2}$

In expanded form, the degrees of the certificates are $\deg(s_2) = 10$ and $\deg(s_3) = 6$.

- Extended Gram matrix method: Let $m^d := \begin{pmatrix} 1 \\ X \\ \vdots \\ X^{d-1} \end{pmatrix}$. There are no positive semidefinite matrices A_0 and A_1 of dimension $d \times d$ with $d < 3$ such that $f = ((m^d)^T A_0 m^d) + ((m^d)^T A_1 m^d) \cdot g_1$. However, there exist positive semidefinite matrices that satisfy these conditions with $d = 3$. We obtain the following.

$$A_0 = \begin{pmatrix} 2.40002 & 0.533338 & -0.333341 \\ 0.533338 & 0.199996 & -0.0333376 \\ -0.333341 & -0.0333376 & 0.0666673 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 0.066669 & 4.526684 \times 10^{-7} & -0.000024 \\ 4.526684 \times 10^{-7} & 0.000050 & 4.720584 \times 10^{-8} \\ -0.000024 & 4.720584 \times 10^{-8} & 2.817258 \times 10^{-8} \end{pmatrix}$$

The certificate obtained with this approach is

$$\begin{aligned} f &= ((m^3)^T A_0 m^3) + ((m^3)^T A_1 m^3) \cdot g_1 \\ &\approx (1.5492 + 0.344267X - 0.21517X^2)^2 + (0. + 0.000581784X^2)^2 \\ &\quad + (0. + 0.285441X + 0.14272X^2)^2 \\ &\quad + ((0.258205 + 1.75314 * 10^{-6}X - 0.0000967477X^2)^2 \\ &\qquad + (0. + 0.00708567X + 6.68609 * 10^{-6}X^2)^2 + (0. + 0.000136995X^2)^2) \cdot g_1 \end{aligned}$$

In expanded form, the degrees of the certificates are $\deg((m^3)^T A_0 m^3) = 4$ and $\deg((m^3)^T A_1 m^3) = 4$.

Example 4.14 Let the input polynomial be a natural generator $(X+1)(X-1)$.

- SaturatedCert: We use SOSBasicLemma with inputs $(X+1)(X-1)$ and $-(X+2)(X-3)$ to obtain sums of squares σ, τ such that $1 = \sigma(X+1)(X-1) + \tau(-(X+2)(X-3))$ where

  - $\sigma = \frac{1224263}{3896156}(-\frac{1518593}{9794104}X^2 - \frac{723815}{14691156}X + 1)^2 + \frac{14940634371083}{686868427156032}(-\frac{10649700814773}{29881268742166}X^2 + X)^2 + \frac{360146266268039063}{931376675979220111168}(X^2)^2 + \frac{1323}{170573}((X+2)(X-3))^2$
  - $\tau = \frac{1043377}{4093752}(-\frac{280155}{2086754}X^2 - \frac{138821}{2086754}X + 1)^2 + \frac{222337281587}{17085306722016}(-\frac{105150010263}{222337281587}X^2 + X)^2 + \frac{76390416521127}{303397897057114808}(X^2)^2 + \frac{10426}{974039}((X+1)(X-1))^2$

  Hence, we obtain the certificate $(X+1)(X-1) = s_0 + s_1 \cdot g_1$ where

  - $s_0 = (X+1)^2(X-1)^2\sigma$
  - $s_1 = \tau$

  In expanded form, the degrees of the certificates are $\deg(s_0) = 8$ and $\deg(s_1) = 4$.

- Shang et. al's general method: Let $t_n(a,b) := \sum_{i=0}^{n} \frac{(-1)^i}{i!} \frac{(X-a)^{2i}}{(2b)^i}$ First, we fix the input polynomial f at $x = -1$ with

  $$f_1 := f - \frac{1}{4} t_1(-1,4)^2 \frac{(X-(-2))^2}{(-1-(-2))^2} \frac{(X-1)^2}{(-1-1)^2} \frac{(X-5)^2}{(-1-5)^2} \cdot g_1$$

  which is a local sums of squares in $\mathbb{R}[[X-1]]$.

  We fix f_1 at $x = 1$ with

  $f_2 := f_1 - \frac{(X-2)^2}{6} \frac{(X-(-2))^2}{(1-(-2))^2} \frac{(X-(-1))^2}{(1-(-1))^2} \frac{(X-5)^2}{(1-5)^2} \cdot g_1$, which is a sums of squares. Therefore, the certificate obtained using this method is $(X+1)(X-1) = s_2 + s_3 \cdot g_1$ where

– $s_2 = \frac{2267}{4608}((X+1)(X-1)(-\frac{3445}{1305792}X^5+\frac{47369}{3917376}X^4+\frac{12215}{217632}X^3-\frac{359423}{1450880}X^2-\frac{5875}{27204}X+1))^2+\frac{13078917289}{30085447680}((X+1)(X-1)(\frac{277035625}{52315669156}X^5+\frac{828545605}{470841022404}X^4-\frac{7343687830}{39236751867}X^3+\frac{3249138825}{104631338312}X^2+X))^2+\frac{9964899093437107939}{666512004476200550400}((X+1)(X-1)(\frac{256724554600819050}{9964899093437107939}X^5-\frac{176869388065834730}{1423557013348158277}X^4-\frac{2239275551240532900}{9964899093437107939}X^3+X^2))^2+\frac{175161724132613869398983}{952160936147766697988063232}((X+1)(X-1)(\frac{8243659133685020005983}{875808620663069346994915}X^5-\frac{1244400643078044203791 13}{525485172397841608196949}X^4+X^3))^2$
$+\frac{75017030716875205016260073 53}{188290838042063601193243027537920}((X+1)(X-1)(-\frac{14888115057049474705073850 48}{75017030716875205016260073 53}X^5+X^4))^2$
$+\frac{233020616053820196557387 3681}{74666551149365964058424106466467 8400}((X+1)(X-1)(X^5))^2$

– $s_3 = \frac{(X-2)^2}{6}\frac{(X-(-2))^2}{(1-(-2))^2}\frac{(X-(-1))^2}{(1-(-1))^2}\frac{(X-5)^2}{(1-5)^2}$
$+\frac{1}{4}(t_1(-1,4))^2\frac{(X-(-2))^2}{(-1-(-2))^2}\frac{(X-1)^2}{(-1-1)^2}\frac{(X-5)^2}{(-1-5)^2}$

In expanded form, the degrees of the certificates are $\deg(s_2) = 14$ and $\deg(s_3) = 10$.

- Extended Gram matrix method: We use the vector of monomials m^d as in Example 4.13. There are no positive semidefinite matrices A_0 and A_1 of dimension $d \times d$ with $d < 4$ such that $f = ((m^d)^T A_0 m^d) + ((m^d)^T A_1 m^d) \cdot g_1$. However, there exist positive semidefinite matrices that satisfy these conditions with $d = 4$. We obtain the following.

$$A_0 = \begin{pmatrix} 0.10149 & -0.0332116 & -0.10149 & 0.0332115 \\ -0.0332116 & 0.0247483 & 0.0332116 & -0.0247482 \\ -0.10149 & 0.0332116 & 0.10149 & -0.0332115 \\ 0.0332115 & -0.0247482 & -0.0332115 & 0.0247481 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 0.183582 & -0.020833 & -0.000437 & 0.000017 \\ -0.020833 & 0.025625 & -0.000016 & -0.000021 \\ -0.000437 & -0.000016 & 0.000044 & 2.766806 \times 10^{-8} \\ 0.000017 & -0.000021 & 2.766806 \times 10^{-8} & 5.527610 \times 10^{-8} \end{pmatrix}$$

The certificate obtained with this approach is

$$
\begin{aligned}
f &= ((m^4)^T A_0 m^4) + ((m^4)^T A_1 m^4) \cdot g_1 \\
&\approx (0. + 0.117814X + 2.53858 * 10^{-7} X^2 - 0.117813X^3)^2 \\
&\quad + (0. + 0.000294231X^2 - 4.34777 * 10^{-8} X^3)^2 \\
&\quad + (0. + 0.00029423X^3)^2 + (0.318576 - 0.10425X - 0.318575X^2 + 0.10425X^3)^2 \\
&\quad + ((0. + 0.152517X - 0.000436259X^2 - 0.000130442X^3)^2 \\
&\qquad + (0. + 0.0065479X^2 + 1.81908 * 10^{-6} X^3)^2 \\
&\qquad + (0.428464 - 0.0486242X - 0.00102061X^2 + 0.0000403189X^3)^2 \\
&\qquad + (0. + 0.000191395X^3)^2) \cdot g_1
\end{aligned}
$$

In expanded form, the degrees of the certificates are $\deg((m^4)^T A_0 m^4) = 6$ and $\deg((m^4)^T A_1 m^4) = 6$.

4.5 Experiments

The SaturatedCert algorithm discussed in Section 4.4 has been implemented in `Maple` as a package named `CertSatQM`[2]; `CertSatQM` provides a procedure, `liftPO2QM`, which uses natural generators and an arbitrary polynomial as generators. The method first checks the membership of the input polynomial in the quadratic module by checking nonnegativity and returns a `Maple`'s table encoding the generators of the quadratic module as indices and the sum of squares multipliers as entries. The implementation has been experimented with on several examples.

The implementation has been compared with the implementation with `RealCertify` [51] and `MomentPolynomialOpt` [9] , a `Maple` and a `Julia` package

[2] `CertSatQM` is freely available in the following GitHub repository: https://github.com/typesAreSpaces/CertSatQM

that compute certificates of strictly positive elements in quadratic modules using a semidefinite programming approach. All experiments were performed on an M1 MacBook Air with 8GB of RAM running macOS.

To the best of our knowledge, this is the first implementation that computes certificates for saturated quadratic modules. Previous implementations have focused on the case whenever the input polynomial is strictly positive polynomials using Positivstellensatz results. In particular, the latter excludes polynomials sharing common zeros with the original generators without, and most of the time, it is needed to include a polynomial in the generators that witnesses the Archimedean property. A consequence is that these implementations generate certificates in terms of the original generators, and it is not necessary to include an Archimedean polynomial in the original generators.

4.5.1 Computing certificates of natural generators

This benchmark uses a monogenic generator of the form $p = -\prod_{n=1}^{k}(X^2 - n^2)$. We compute a certificate for the left natural generator $X + k$ and a certificate for the right natural generator $-(X - k)$. Table 4.1 reports the time in seconds required to complete this benchmark.

k	Left natural generator	Right natural generator
1	0.101	0.035
2	0.189	0.157
3	0.295	0.297
4	0.371	0.352
5	0.524	0.534
6	0.705	0.666
7	0.903	0.847
8	1.220	1.141
9	1.523	1.345
10	1.848	1.683

Table 4.1: Computing certificates of products of natural generators

The following benchmark compares `CertSatQM` with `RealCertify` and `MomentPolynomialOpt` using strictly positive polynomials by adding 1 to the the left natural generator and right natural generator from the benchmark discussed in Table 4.1. We use the `multivsos_internal` command from `RealCertify` with settings [3] $\texttt{algo} = 1, \texttt{gmp} = \textit{true}$ allowing `RealCertify` to find exact certificates; we use the `sos_decompose` command from `MomentPolynomialOpt` with settings $\texttt{exact} = \textit{true}, \texttt{round} = 10000$ to obtain exact certificates and allow the software to handle high-precision terms. The command `sos_decompose` also includes an argument denoting the dimension of the search space of polynomials of fixed degree; we use a loop to increase this parameter in the case where `sos_decompose` is unable to find a solution.

Table 4.2 reports the timings in seconds of the two packages and the degree of the sums of squares term.

[3] These are interval variables of the package

	`RealCertify`		`MomentPolynomialOpt`		`CertSatQM`	
k	Time	Degree	Time	Degree	Time	Degree
1	0.03	4	0.071	2	0.094	2
2	0.06	4	0.289	4	0.021	4
3	0.012	6	0.085	6	0.029	6
4	0.027	8	0.104	8	0.035	8
5	0.037	10	0.124	10	0.044	10
6	N/A	N/A	0.154	12	0.055	12
7	N/A	N/A	0.173	14	0.068	14
8	N/A	N/A	0.191	16	0.083	16
9	N/A	N/A	0.221	18	0.094	18
10	N/A	N/A	N/A	N/A	0.161	20

Table 4.2: Computing certificates of strictly positive polynomials of the form $(X + k) + 1$

	`RealCertify`		`MomentPolynomialOpt`		`CertSatQM`	
k	Time	Degree	Time	Degree	Time	Degree
1	0.03	4	0.115	2	0.093	2
2	0.005	4	0.261	4	0.019	4
3	0.012	6	0.086	6	0.027	6
4	0.027	8	0.102	8	0.04	8
5	0.036	10	0.122	10	0.046	10
6	N/A	N/A	0.146	12	0.056	12
7	N/A	N/A	0.176	14	0.067	14
8	N/A	N/A	0.2009	16	0.083	16
9	N/A	N/A	0.214	18	0.095	18
10	N/A	N/A	N/A	N/A	0.157	20

Table 4.3: Computing certificates of strictly positive polynomials of the form $-(X - k) + 1$

The N/A entries in Tables 4.2, 4.3 indicate that the implementation was unable to find a certificate. We compare the certificates obtained by the previous implementations for case $-X + 2 \in \mathrm{QM}(-(X^2 - 1))$; we show the sums of squares σ_0, σ_1 in $-X + 2 = \sigma_0 + \sigma_1 \cdot (-(X^2 - 1))$.

- `RealCertify`

$$\begin{aligned}
\sigma_0 = & \frac{2442079919018640903387597573476994211206586093767920977422187363401295226126201X^4}{31250000000000000000000000000000000000000000000000000000000000000000000000000000000} \\
& + \frac{244140624999999999999999999999999966589435183851492141828286349634030273953 29X^2}{3125000000000000000000000000000000000000000000000000000000000000000000000000000000 0} \\
& + 2\Big(-\frac{5073300512324625981875447467629096921611X^2}{10000000000000000000000000000000000000000} \\
& \quad + \frac{920212869799648942085416819931834464543X}{125000000000000000000000000000000000000000} + \frac{200027342320752718586482929628464751087}{400000000000000000000000000000000000000}\Big)^2 \\
& \quad + 2\Big(\frac{2499878069966048103761958160758822467269X}{5000000000000000000000000000000000000000} - \frac{2499587325459640748846805052717437199869X^2}{5000000000000000000000000000000000000000}\Big)^2 \\
& + \frac{1740871302325074754119609205384786859389(X^2+X)^2}{19531250000000000000000000000000000000000000000000000000000000000000000000000000} \\
& + \frac{832249365719387096406349883240494507593(1-X)^2}{12500000000000000000000000000000000000000000000000000000000000000000000000000000} \\
& + \frac{49218640874676983436633944346100928555035040190451169585238618162809132963740103}{50000000000000000000000000000000000000000000000000000000000000000000000000000000} \\
\sigma_1 = & \frac{6384937177808586970315090635845040435291340627581019258892883350846107654249X^2}{12207031250000000000000000000000000000000000000000000000000000000000000000000} \\
& + 2\Big(\frac{507686153818313936360606454553717608317}{1000000000000000000000000000000000000000} - \frac{4996814523822776714108503870298404939983X}{10000000000000000000000000000000000000000}\Big)^2
\end{aligned}$$

- `MomentPolynomialOpt`

$$\sigma_0 = (-\frac{1}{2}X+1)^2 + \frac{3}{4}(X)^2$$

$$\sigma_1 = 1$$

- `CertSatQM`

$$\sigma_0 = \frac{35152}{20931}(X - \frac{20931}{70304})^2 + \frac{505372919}{2943066048}$$

$$\sigma_1 = \frac{35152}{20931}$$

4.6 Summary

This chapter provides the SaturatedCert algorithm to compute a certificate for a member f in an Archimedean saturated quadratic module $\mathrm{QM}(G)$. The method relies on a sequence of lifting operations of the certificate obtained from $\mathrm{PO}(\mathrm{Nat}(\mathcal{S}(G)))$, $\mathrm{QM}(\mathrm{Nat}(\mathcal{S}(G)))$, to $\mathrm{QM}(G)$.

The application of the "Basic Lemma" Theorem 2.19 is crucial to obtain several sums of squares needed for the certificates, which are non-trivial to obtain. The Basic Lemma requires two conditions to operate, the first one asks the semialgebraic set involved to be bounded and that the polynomials in consideration are coprime. Although the second condition is not hard to fulfill in our methods, the boundedness condition does not follow easily from the input of the algorithm in SaturatedCert. To alleviate these shortcomings of the Basic Lemma, Chapter 3 introduces key results that provide more flexibility of the inputs of the algorithm and use the Basic Lemma in our construction.

The implementation `CertSatQM` discussed in Section 4.5 shows that the proposed algorithms are suitable to find a certificate in a reasonable amount of time. Comparison with Shang et. al's general method shows that our method provides a lower degree in the certificates obtained. Comparisons with `RealCertify` and `MomentPolynomialOpt` indicate that our implementation finds solutions of comparable quality (in terms of polynomial degree) in a similar amount of time, while also handling cases that these packages cannot.

Chapter 5

Certificates Computation in Monogenic Archimedean Quadratic Modules in $\mathbb{A}[X]$

In this chapter, we introduce the MonogenicCert algorithm that computes certificates for members in an Archimedean monogenic quadratic module. The monogenic case has been studied in the context of a formal power series ring $\mathbb{R}[[X-a]]$ in [5] which provides a decision procedure based on a lattice structure using signs and order conditions at the localization of $\mathbb{R}[[X-a]]$. This line of research ended with a general membership decision procedure in [4]. However, in this case, no algorithms have been discussed to compute certificates for members. We use methods from Chapter 5 like compactification and the certificate of non-negative factors. The MonogenicCert algorithm is extensively used in Chapter 7 as an intermediate step.

Our approach follows a divide-and-conquer approach; the divide part identifies and computes certificates for the strictly positive factor, "redundant factors" and the nonnegative factor of the input polynomial; the conquer part combines the certificates

of the previous factors to obtain the original input polynomial. One of the key properties of this case is that the monogenic case is a preordering, thus the conquer part can simply multiply each certificate of the factors directly. Section 5.1 discusses techniques to simplify the degree of both the input polynomial and the generator of the quadratic module, which preserve the sign and order conditions of the original quadratic module; these reductions help us reduce the degree of the input to handle a simpler problem. In Section 5.2, we consider a reduction of the original certificate problem that does not involve roots of even multiplicities; temporarily we "ignore" these and add them back as factors or sums of squares multipliers in a certificate of the original problem. In Section 5.3 we describe the MonogenicCert algorithm, which applies the reductions discussed in the previous section and uses the Basic Lemma to compute a certificate for the nonnegative factor of the input polynomial. Section 5.4 provides examples and a comparison of the results obtained between the proposed method and Shang et. al's general method. Lastly, Section 5.5 concludes this chapter by summarizing the contributions.

5.1 Redundant sums of squares

In this section, we discuss simplifications for both the generator and the input polynomial by removing factors that do not contribute information to its signature and obtain conditions to detect them. We will formally define redundant sums of square factors.

Definition 5.1 Suppose $f \in \mathrm{QM}(g)$ such that $f = sf_1$ and $s \in \sum \mathbb{A}[X]^2$. We call s a **redundant sum of squares factor** of f if $f_1 \in \mathrm{QM}(g)$.

Clearly, from the definition of redundant sums of squares factors in Definition 5.1, it is possible to reduce the certificate problem of f to f_1 as a certificate of f is obtained

from a certificate of f_1, i.e., by obtaining a certificate of $f_1 = \sigma_0 + \sigma_1 \cdot g$ in $\mathrm{QM}(g)$ we obtain $f = sf_1 = s\sigma_0 + s\sigma_1 \cdot g$ which is a certificate of f in $\mathrm{QM}(g)$.

5.1.1 Redundancies in generator

Let us consider redundant sums of squares factors in the monogenic case. First, we will discuss a few examples in order to provide some insight into our statement.

Example 5.2 Let us consider $G_1 = \{-(X+1)X^2(X-1)\}$. We see that the polynomials $g_1 = X^2$ and $g_2 = -(X+1)(X-1)$ are coprime and both are nonnegative over $\mathcal{S}(G_1)$. Hence, g_1 is a redundant factor in G_1 so $G_2 = \{g_2\}$ is an equivalent sets of generators.

Example 5.3 Let us consider $G_1 = \{-(X+2)(X+1)X^2(X-1)(X-2)\}$. We see that the polynomials $g_1 = X^2$ and $g_2 = -(X+2)(X+1)(X-1)(X-2)$ are coprime but only g_1 is nonnegative over $\mathcal{S}(G_1)$ since at $X = 0$ the polynomial g_2 evaluates to -16. Hence, X^2 is not a redundant factor in G_1, in fact $\mathrm{QM}(G_1) \neq \mathrm{QM}(g_2)$.

The following theorem states our main result:

Theorem 5.4 *Let* $g \in \mathbb{A}[X]$ *be such that* $\mathcal{S}(g)$ *is bounded and* $s \in \sum \mathbb{A}[X]^2$ *such that* $g = s \cdot g_1$. *Then* $\mathrm{QM}(g) = \mathrm{QM}(g_1)$ *if and only if* $(s, g_1) = 1$ *and* $\mathcal{S}(g) = \mathcal{S}(g_1)$.

Proof. First, we will prove the converse of the double implication. The two conditions satisfy the requirements of Theorem 2.20. Hence, there exist sums of squares σ, τ such that $1 = \sigma s + \tau g_1$. Multiplying g_1 to both sides of this equation, we obtain $g_1 = \tau g_1^2 + \sigma \cdot g \in \mathrm{QM}(g)$, and hence $\mathrm{QM}(g_1) \subseteq \mathrm{QM}(g)$. The reverse containment is trivial; therefore $\mathrm{QM}(g) = \mathrm{QM}(g_1)$.

Now, we will prove the forward direction of the implication. Since $Q = \mathrm{QM}(g) = \mathrm{QM}(g_1)$ we have $S = \mathcal{S}(g) = \mathcal{S}(g_1)$. Suppose, by contradiction, that s and g_1 are not coprime. The latter implies that $(s, g_1) = p$ with $\deg(p) \geq 1$. Since both quadratic modules are monogenic, from Theorem 2.16 we find that Q is uniquely specified by the order information, hence $\kappa^+_{a_i}(g) = \kappa^+_{a_i}(g_1)$ and $\kappa^-_{b_i}(g) = \kappa^-_{b_i}(g_1)$ for every endpoint $a_i < b_i$ of S and $\kappa_{a_i}(g) = \kappa_{a_i}(g_1)$ for every isolated point a_i in S. Since $\deg(p) \geq 1$, at least one of these equations is false. Hence $(s, g_1) = 1$ as required. □

By iterating Theorem 5.4 to given a generator g such that $\mathrm{QM}(g)$ is Archimedean, we obtain a polynomial g_1 such that is free of sums of squares factors. We refer to g_1 as the **reduced generator** of g.

5.1.2 Redundancies in input polynomial

Any polynomial f in $\mathbb{A}[X]$ is a product of powers of linear and quadratic irreducible factors. The following observation allows us to ignore the quadratic factors whenever they satisfy similar conditions as in Theorem 5.4.

Theorem 5.5 *Let* $G \subseteq \mathbb{A}[X]$, $\mathcal{S}(G) = \bigcup_{i=0}^{k} [a_i, b_i]$ *and* $f \in \mathbb{A}[X]$ *be a polynomial such that* $f = sf_1$ *where* $s = (X - b)^2 + c^2$ *with* $b, c \in \mathbb{A}$. *If* $(s, f_1) = 1$ *and* $f_1 \geq 0$ *over* $\mathcal{S}(G)$ *then* $f \in \mathrm{QM}(G)$ *if and only if* $f_1 \in \mathrm{QM}(G)$.

Proof. The converse of the implication is trivial from the definition of quadratic modules. Then, let us prove that if $f \in \mathrm{QM}(G)$ then $f_1 \in \mathrm{QM}(G)$.

First, consider the case where $c \neq 0$. Since $f \in \mathrm{QM}(G)$, by Theorem 2.5, we have $f|_{\mathcal{S}(G)} \geq 0$ and f satisfies the order conditions for each a_i and b_i of $\mathcal{S}(G)$. Applying the same theorem, in order to prove $f_1 \in \mathrm{QM}(G)$, it is enough to prove $f_1|_{\mathcal{S}(G)} \geq 0$ and f_1 also satisfies the same order conditions.

By contradiction, suppose that there exists $a \in \mathcal{S}(G)$ such that $f_1(a) < 0$. Since $f(a) \geq 0$, we have $(a-b)^2 + c^2 \leq 0$. This is impossible since $a, b \in \mathbb{R}$, and $c \neq 0$. Thus, $f_1(a) \geq 0$ for every $a \in \mathcal{S}(G)$.

Since the first non-zero coefficient of the Taylor series of $((X-b)^2+c^2)$ at any point $a \in \mathbb{R}$ is $(a-b)^2 + c^2 > 0$, then for every a_i and b_i we have $\mathrm{ord}_{a_i}(f_1) = \mathrm{ord}_{a_i}(f)$ and $\mathrm{ord}_{b_i}(f_1) = \mathrm{ord}_{b_i}(f)$, thus f_1 satisfies the same order conditions as the polynomial f.

If $c = 0$ then we see that b in $s = (X-b)^2$ cannot be any endpoint of $\mathcal{S}(G)$. Hence, the nonnegativity and order conditions of f_1 are the same as the other conditions of f since $(s, f_1) = 1$, so $f_1 \in \mathrm{QM}(G)$. □

Theorem 5.5 allows us to remove sums of squares factors s from the input polynomial $f = sf_1$ under the condition that s and f_1 are coprime. However, there are redundant sums squares factors of f that do not satisfy the coprime condition. For the latter, we use information about the signature of the quadratic module in order to safely reduce the input polynomial. The following proposition indicates how to perform this observation.

Proposition 5.6 *Let f be the input polynomial, g a reduced monogenic generator, and $s = (X-c)^2$ such that s divides both f and g.*

1. *If $\mathrm{ord}_c(g)$ is even, $\mathrm{ord}_c(f)$ is even, and $\epsilon_c(f) > 0$ then $(X-c)^{\mathrm{ord}_c(f)}$ is a redundant factor of f.*

2. *If $\mathrm{ord}_c(g)$ is even, $\mathrm{ord}_c(f)$ is even, and $\epsilon_c(f) < 0$ then $(X-c)^{\mathrm{ord}_c(f)-\mathrm{ord}_c(g)}$ is a redundant factor of f.*

3. *If $\mathrm{ord}_c(g)$ is even and $\mathrm{ord}_c(f)$ is odd, then $(X-c)^{\mathrm{ord}_c(f)-\mathrm{ord}_c(g)-1}$ is a redundant factor of f.*

4. *If* $\operatorname{ord}_c(g)$ *is odd and* $\operatorname{ord}_c(f)$ *is even, then* $(X-c)^{\operatorname{ord}_c(f)}$ *is a redundant factor of* f*.*

5. *If* $\operatorname{ord}_c(g)$ *is odd and* $\operatorname{ord}_c(f)$ *is odd, then* $(X-c)^{\operatorname{ord}_c(f)-\operatorname{ord}_c(g)}$ *is a redundant factor of* f*.*

Proof. Since g is a reduced monogenic generator, by Theorem 2.16, it is of the form $-\prod_{\substack{i=0\\ a_i<b_i}}^{m}(X-a_i)^{\kappa_{a_i}^+}(X-b_i)^{\kappa_{b_i}^-}\prod_{\substack{i=0\\ a_i=b_i}}^{m}(X-a_i)^{\kappa_{a_i}}$ and so s divides only one factor. First, suppose that s divides the factor of the form $(X-a_i)^{\kappa_{a_i}}$. Since κ_{a_i} is even, this case covers items 1, 2, and 3 above.

If $\operatorname{ord}_c(f)$ is even and $\epsilon_c(f) > 0$ then f is positive over any neighbor around c so f_1, where $f = f_1(X-c)^{\operatorname{ord}_c(f)}$, is nonnegative over $\mathcal{S}(g)$ and satisfies the order conditions to be a member of $\mathrm{QM}(g)$.

If $\operatorname{ord}_c(f)$ is even and $\epsilon_c(f) < 0$ then f_1, where $f = f_1(X-c)^{\operatorname{ord}_c(f)-\operatorname{ord}_c(g)}$, is nonnegative over $\mathcal{S}(g)$ and satisfies the order conditions to be a member in $\mathrm{QM}(g)$.

If $\operatorname{ord}_c(f)$ is odd, then f_1, where $f = f_1(X-c)^{\operatorname{ord}_c(f)-\operatorname{ord}_c(g)-1}$ is nonnegative over $\mathcal{S}(g)$. Additionally, $\operatorname{ord}_c(f_1) = \operatorname{ord}_c(g)+1$, so f_1 satisfies the order conditions to be a member of $\mathrm{QM}(g)$.

Now, suppose that s divides the factor of the form $(X-a_i)^{\kappa_{a_i}^+}$. Since $\kappa_{a_i}^+$ is odd, this case covers items 4 and 5 above since the reasoning for the case where s divides a factor of the form $(X-b_i)^{\kappa_{b_i}^-}$ is similar.

If $\operatorname{ord}_c(f)$ is even, then f is positive over any neighbor around c as otherwise f is negative over any neighbor around c and would contradict the nonnegativity condition of f being a member of $\mathrm{QM}(g)$. Hence $(X-c)^{\operatorname{ord}_c(f)}$ is a redundant sums of squares factor using a similar argument for item 1.

If $\operatorname{ord}_c(f)$ is odd, then $\epsilon_c(f) > 0$ as otherwise f would evaluate to a negative value

to the left of c which contradicts the nonnegativity conditions of being a member of $\mathrm{QM}(g)$. Hence f_1, where $f = f_1(X-c)^{\mathrm{ord}_c(f)-\mathrm{ord}_c(g)}$ is nonnegative over $\mathcal{S}(g)$ and satisfies the order conditions of $\mathrm{QM}(g)$, so f_1 is a member of $\mathrm{QM}(g)$. □

Corollary 5.7 *Let g be a single generator without redundant sums of squares factors, f be an input polynomial of $\mathrm{QM}(g)$ without redundant sums of squares factors and $a \in \partial\,\mathcal{S}(g) \cap \mathcal{Z}(f)$.*

- *If $\mathrm{ord}_a(g)$ is even and $\mathrm{ord}_a(f)$ is even, then $\mathrm{ord}_a(f) = \mathrm{ord}_a(g)$*
- *If $\mathrm{ord}_a(g)$ is even and $\mathrm{ord}_a(f)$ is odd, then $\mathrm{ord}_a(f) = \mathrm{ord}_a(g) + 1$*
- *If $\mathrm{ord}_a(g)$ is odd, then $\mathrm{ord}_a(f)$ is odd and $\mathrm{ord}_a(f) = \mathrm{ord}_a(g)$*

We call **reduced** an input polynomial f whenever there are no sums of squares factors s such that $f = s \cdot f_1$ for some f_1. We assume for the rest of this chapter that both the input polynomial and the generator in the quadratic module of discussion are both reduced.

5.2 Removing isolated points of even multiplicities

The multiplicities of isolated points of type A in the monogenic generator are even according to Theorem 2.16. From this observation, it is possible to ignore these isolated points as shown in the following proposition. First, we introduce a useful notation.

Notation 5.8 *Let $\mathcal{P}$ be a finite subset of $\mathbb{A}$, f any polynomial in $\mathbb{A}[X]$ and G a set of generators. We denote by $f_{\overline{\mathcal{P}}}$ the polynomial $\frac{f}{\prod_{a\in\mathcal{P}}(X-a)^{\min\{\mathrm{ord}_a(f),\kappa_a(G)\}}}$.*

Proposition 5.9 *Consider a monogenic generator g and a member f of $\mathrm{QM}(g)$. Let $\mathcal{A}$ be the intersection of real zeros of f and isolated points of even multiplicities in $\mathcal{S}(g)$. We have $f_{\overline{\mathcal{A}}} \in \mathrm{QM}(g_{\overline{\mathcal{A}}})$. Furthermore, if σ_0, σ_1 are the certificates of $f_{\overline{\mathcal{A}}} \in \mathrm{QM}(g_{\overline{\mathcal{A}}})$, i.e. $f_{\overline{\mathcal{A}}} = \sigma_0 + \sigma_1 \cdot g_{\overline{\mathcal{A}}}$, then $f = \sigma_0 \prod_{a \in \mathcal{A}} (X-a)^{\kappa_a(g)} + \sigma_1 \cdot g$.*

Proof. Since $f \in \mathrm{QM}(g)$, from Theorem 2.5, we have $\mathrm{ord}_a(f) \geq \kappa_a(g)$, therefore, both $f_{\overline{\mathcal{A}}}$ and $g_{\overline{\mathcal{A}}}$ are polynomials and the order conditions of Theorem 2.5 are satisfied. To prove that $f_{\overline{\mathcal{A}}} \in \mathrm{QM}(g_{\overline{\mathcal{A}}})$, it is enough to show that $f_{\overline{\mathcal{A}}}$ is nonnegative over $\mathcal{S}(g_{\overline{\mathcal{A}}})$.

Suppose, by contradiction, that there exists $x \in \mathcal{S}(g_{\overline{\mathcal{A}}})$ such that $f_{\overline{\mathcal{A}}}(x) = \frac{f(x)}{\prod_{a \in \mathcal{A}} (x-a)^{\kappa_a(g)}} < 0$, so $f(x) < 0$. We see that $\mathcal{S}(g_{\overline{\mathcal{A}}}) = \mathcal{S}(g) \setminus \mathcal{A}$, since $\kappa_a(g)$ is even; thus, we have that $f(x) < 0$ for some point in $\mathcal{S}(g) \setminus \mathcal{A}$, which is a contradiction to the fact that f is nonnegative over $\mathcal{S}(g)$.

Clearly, multiplying the sums of squares $\prod_{a \in \mathcal{A}} (X-a)^{\kappa_a(g)}$ by the certificate of $f_{\overline{\mathcal{A}}}$ in $\mathrm{QM}(g_{\overline{\mathcal{A}}})$ provides a certificate of f in $\mathrm{QM}(g)$. □

Working with $f_{\overline{\mathcal{A}}}$ and $\mathrm{QM}(g_{\overline{\mathcal{A}}})$ instead of f and $\mathrm{QM}(g)$ allows us to use Theorem 2.19 by splitting factors of g into f.

An interesting property of $f_{\overline{\mathcal{A}}}$ and $g_{\overline{\mathcal{A}}}$ is that their common zeros are the end points of regular intervals of $\mathcal{S}(g_{\overline{\mathcal{A}}})$. This allows us to use Theorem 2.19 to find a certificate of $f_{\overline{\mathcal{A}}}$ in $\mathrm{QM}(g_{\overline{\mathcal{A}}})$ as the orders of the common zeros are the same as f and g by Corollary 5.7, since both are reduced polynomials.

In the previous discussion, we addressed factors with even multiplicities shared by both the generator g and the input polynomial f. In case the factor with even multiplicity belongs only to f, i.e., $f = (X-a)^2 f_1$, the procedure to compute a certificate for f reduces to compute a certificate for f_1 from Theorem 5.5. On the other hand, if the factor with even multiplicity belongs only to g and it is not

redundant[1], we use the following construction to compute a certificate of f in $\mathrm{QM}(g)$.

Theorem 5.10 *Consider a monogenic generator g and a member $f \in \mathrm{QM}(g)$. Let $\{a_1, \ldots, a_m\}$ be roots of g such that $\mathrm{ord}_{a_i}(f) = 0$, $\mathrm{ord}_{a_i}(g) = 2n_i$, and $\epsilon_{a_i}(g) = -1$, i.e., $g = \prod_{i=1}^{m} (X - a_i)^{2n_i} g_1$, but $\prod_{i=1}^{m} (X - a_i)^{2n_i}$ does not divide f. We have $f \in \mathrm{QM}(g_1)$. Furthermore, given a certificate of $f = \sigma_0 + \sigma_1 \cdot g_1$ in $\mathrm{QM}(g_1)$, a certificate $f = \sigma_2 + \sigma_3 \cdot g$ is computable.*

Proof. From Theorem 2.15, we conclude that if $f \in \mathrm{QM}(g)$ then $f \in \mathrm{QM}(g_1)$ as g_1 has m isolated point less than g. Assume $f = \sigma_0 + \sigma_1 \cdot g_1$. Multiplying $\prod_{i=1}^{m} (X - a_i)^{2n_i}$ by the previous equation, we obtain

$$\prod_{i=1}^{m} (X - a_i)^{2n_i} f = \prod_{i=1}^{m} (X - a_i)^{2n_i} \sigma_0 + \sigma_1 \cdot g \tag{5.1}$$

We use the BasicLemma algorithm with inputs $\prod_{i=1}^{m} (X - a_i)^{2n_i}$, f, and g to find σ, τ strictly positive over $\mathcal{S}(g)$ such that $1 = \sigma \prod_{i=1}^{m} (X - a_i)^{2n_i} + \tau f$. Multiplying f to both sides of the previous equation, we obtain

$$f = \sigma \prod_{i=1}^{m} (X - a_i)^{2n_i} f + f^2 \tau \tag{5.2}$$

As σ, τ are strictly positive over $\mathcal{S}(g)$, we use the Certificate algorithm to compute a certificate of σ, τ in $\mathrm{QM}(g)$, i.e., $\sigma = \sigma_2 + \sigma_3 \cdot g$, $\tau = \sigma_4 + \sigma_5 \cdot g$. Substituting the certificates of σ, τ and Equation (5.1) into Equation (5.2), we obtain

[1]See Section 5.1.1

$$
\begin{aligned}
f &= \sigma(\prod_{i=1}^{m}(X-a_i)^{2n_i}\sigma_0+\sigma_1\cdot g)+f^2\tau\\
&= (\sigma_2+\sigma_3\cdot g)(\prod_{i=1}^{m}(X-a_i)^{2n_i}\sigma_0+\sigma_1\cdot g)+f^2(\sigma_4+\sigma_5\cdot g)\\
&= \prod_{i=1}^{m}(X-a_i)^{2n_i}\sigma_0\sigma_2+\sigma_1\sigma_3 g^2+f^2\sigma_4+(\prod_{i=1}^{m}(X-a_i)^{2n_i}\sigma_0\sigma_3+\sigma_1\sigma_2+f^2\sigma_5)\cdot g
\end{aligned}
$$

Hence, the certificate of f in $\mathrm{QM}(g)$ is $f=\sigma_2+\sigma_3\cdot g$ where

- $\sigma_2=\prod_{i=1}^{m}(X-a_i)^{2n_i}\sigma_0\sigma_2+\sigma_1\sigma_3 g^2+f^2\sigma_4$
- $\sigma_3=\prod_{i=1}^{m}(X-a_i)^{2n_i}\sigma_0\sigma_3+\sigma_1\sigma_2+f^2\sigma_5$

□

Example 5.11 We will illustrate the construction in the proof of Theorem 5.10 considering the generator $g=-(X+2)^2X(X-2)$ and the input polynomial $f=(X+1)X$.

First, we "ignore" the square factor $(X+2)^2$ ($g_1:=-X(X-2)$) from the generator and compute a certificate of f in $\mathrm{QM}(g_1)$. This problem is a saturated quadratic module certificate problem, so we obtain a certificate of f using the SaturatedCert algorithm; we obtain $f=(\frac{1}{2}X^2(\frac{1}{2}X^2+1)+\frac{1}{4}g_1^2)+(\frac{1}{2}(\frac{1}{2}X^2+1)+\frac{1}{4}X^2)\cdot g_1$.

Now, we multiply $(X+2)^2$ on both sides of the previous equation, and we obtain $(X+2)^2f=(X+2)^2(\frac{1}{2}X^2(\frac{1}{2}X^2+1)+\frac{1}{4}g_1^2)+(\frac{1}{2}(\frac{1}{2}X^2+1)+\frac{1}{4}X^2)\cdot g$. We apply the BasicLemma algorithm with inputs $(X+2)^2$, f and $\{g\}$ to compute σ,τ strictly positive over $\mathcal{S}(g)$ so that $1=\sigma(X+2)^2+\tau f$; we obtain

- $\sigma=\frac{1}{4}-\frac{3X}{4}+\frac{101}{400}X(1+X)-\frac{3}{320}X(1+X)(2+X)^2(\frac{1}{4}-\frac{3X}{4}+\frac{101}{400}X(1+X))\sum_{i=0}^{5}(1-\frac{3}{320}X(1+X)(2+X)^2)^i$

- $\tau = 2 + \frac{3X}{4} - \frac{101}{400}(2+X)^2 + \frac{3}{320}(2+X)^4(\frac{1}{4} - \frac{3X}{4} + \frac{101}{400}X(1+X))\sum_{i=0}^{5}(1-\frac{3}{320}X(1+X)(2+X)^2)^i$.

We check σ is a nonnegative polynomial over $\mathbb{R}$, so we can find a sums of squares decomposition; in fact, $\sigma = \frac{101}{4294967296*(10)^8}((X-\frac{199}{202})(-320+12X+24X^2+15X^3+3X^4)^3)^2 + \frac{799}{1735166787584*(10)^8}((-320+12X+24X^2+15X^3+3X^4)^3)^2$. On the other hand, τ is not nonnegative over $\mathbb{R}$, so we compute a certificate of it in $\mathrm{QM}(g)$ using the Certificate algorithm. We obtain $\tau = s_0 + s_1 \cdot g$ where

- $s_0 = 2 + \frac{3}{4}X - \frac{101}{400}(X+2)^2 + \frac{3}{320}(X+2)^4(\frac{1}{4} - \frac{3}{4}X + \frac{101}{400}X(X+1))\sum_{i=0}^{5}(1-\frac{3}{320}X(X+1)(X+2)^2)^i + (X+2)^2X(X-2)(\frac{1}{2771241597396986}(X+2)^{24} + (-\frac{30213590}{209948599}(X+2)^2X(X-2) - \frac{149932894}{209948599})^{12})$
- $s_1 = \frac{1}{2771241597396986}(X+2)^{24} + (-\frac{30213590}{209948599}(X+2)^2X(X-2) - \frac{149932894}{209948599})^{12}$

From equation $1 = \sigma(X+2)^2 + \tau f$, we multiply f on both sides to obtain a certificate of f in $\mathrm{QM}(g)$,

$$
\begin{aligned}
f &= \sigma(X+2)^2 f + f^2\tau \\
&= \sigma((X+2)^2(\frac{1}{2}X^2(\frac{1}{2}X^2+1)+\frac{1}{4}g_1^2) + (\frac{1}{2}(\frac{1}{2}X^2+1)+\frac{1}{4}X^2)\cdot g) + f^2(s_0+s_1\cdot g) \\
&= (X+2)^2(\frac{1}{2}X^2(\frac{1}{2}X^2+1)+\frac{1}{4}g_1^2)\sigma + f^2 s_0 + \left((\frac{1}{2}(\frac{1}{2}X^2+1)+\frac{1}{4}X^2)\sigma + f^2 s_1\right)\cdot g
\end{aligned}
$$

5.3 Certificates in Archimedean monogenic quadratic modules

In this section, we discuss how to compute certificates in Archimedean monogenic quadratic modules. The approach uses the fact that $\mathrm{QM}(g)$ is a preordering, so

we compute a certificate of the factors of the reduced input polynomial that are members of $\mathrm{QM}(g)$ and combine their certificates to obtain a certificate for the input polynomial.

First, we identify a strictly positive factor from the reduced input polynomial with respect to the semialgebraic set $\mathcal{S}(g)$. By Putinar's Positivstellensatz [72], this polynomial is a member of $\mathrm{QM}(g)$. We use the Certificate algorithm discussed in [82] to obtain this certificate.

Then, we apply a transformation to both the generator and the remaining factor of the input polynomial so that no isolated points of even multiplicities are involved. Finally, the remaining input polynomial is a product of "in between" factors and a divisor of the transformed generator.

5.3.1 Certificates of strictly positive factor of input polynomial

In this section, we identify a strictly positive factor of the input polynomial with respect to the semialgebraic set of the generator g considered. The goal is to split the input polynomial into a strictly positive polynomial and a polynomial that shares common zeros with g so that both are nonnegative over $\mathcal{S}(g)$. In [82], the authors discuss algorithms for computing certificates for strictly positive polynomials in Archimedean quadratic modules, especially with an emphasis on the univariate case. Thus, we apply their algorithm to compute a certificate for the strictly positive factor of the input polynomial.

We identify a strictly positive factor f_{pos} using Algorithm 13.

Algorithm 13: Compute strictly positive factor f_{pos} from input polynomial f with respect to a bounded semialgebraic set S

StrictPos(f, S)

Input: $f \in \mathbb{A}[X], S \subseteq \mathbb{R}$
Output: $f_{pos} \in \mathbb{A}[X]$
Requires: S is a union of intervals $\bigcup_{i=0}^{k} [a_i, b_i]$
Ensures: $f_{pos} > 0$ over S and f_{pos} divides f

1 **if** *S is empty* **then**
2 **return** f
3 Let $Z = \mathcal{Z}(f)$
4 Let $f_{pos} := 1$
// Process left part
5 **for** $r \in Z$ *and* $r < a_0$ **do**
6 $f_{pos} := (X - r) f_{pos}$
// Process in between parts
7 **for** $i = 0$ *to* $k - 1$ **do**
8 Let $\mathcal{B}_i := \{x \in Z \mid b_i < x < a_{i+1}\}$
9 $h := \prod_{r \in \mathcal{B}_i} (X - r)$
10 **if** $|\mathcal{B}_i|$ *is odd* **then**
11 **if** $\mathrm{ord}_{b_i}(f) > 0$ **then**
12 $f_{pos} := \frac{h}{X - \max(\mathcal{B}_i)} f_{pos}$
13 **else**
14 $f_{pos} := \frac{h}{X - \min(\mathcal{B}_i)} f_{pos}$
15 **else**
16 $f_{pos} := h f_{pos}$
// Process right part
17 **for** $r \in Z$ *and* $r > b_k$ **do**
18 $f_{pos} := (-(X - r)) f_{pos}$
19 **return** f_{pos}

Theorem 5.12 *Given an input polynomial $f \in \mathrm{QM}(g)$, Algorithm 13 returns a polynomial f_{pos} such that f_{pos} divides f and f_{pos} is strictly positive over $\mathcal{S}(g)$.*

Proof. The polynomial f_{pos} clearly divides f as it is the product of factors of f. Let $S = \mathcal{S}(g)$, to show that f_{pos} is strictly positive over S, we use the invariant that f_{pos} is strictly positive over S after every update done to f_{pos} in the Algorithm 13:

- Base case (line 2): f_{pos} is 1, so it is trivially strictly positive over S.
- Left part (line 4): Assuming f_{pos} is strictly positive over S, line 4 updates f_{pos} to be a product between strictly positive polynomials over S.

- In between parts (lines 11, 13, or 16): The degree of the polynomial h on line 8 is the cardinality of the set $\mathcal{B}_i$ from line 7. If $|\mathcal{B}_i|$ is even, then h is strictly positive over S, so f_{pos} on line 16 is strictly positive. If $|\mathcal{B}_i|$ is odd, the degree of the polynomial $\frac{h}{X-\max(\mathcal{B}_i)}$ (resp. $\frac{h}{X-\min(\mathcal{B}_i)}$) on line 11 (resp. line 13) is even and strictly positive over S, so f_{pos} is strictly positive.

- Right part (line 20): Assuming f_{pos} is strictly positive over S, line 20 updates f_{pos} to be a product between strictly positive polynomials over S.

Hence, f_{pos} is strictly positive over $\mathcal{S}(g)$. □

We call the polynomial $f_{non-neg} := \frac{f}{f_{pos}}$ the "nonnegative" part of f where f_{pos} is the output of Algorithm 13.

5.3.2 Certificates of in-between factors

In this section, we assume that the input polynomial f and generator g are reduced. Let us consider the open interval (b_i, a_{i+1}) which is between the intervals of $\mathcal{S}(g)$ and $\mathcal{B}_i$ as the number of real roots of f in (b_i, a_{i+1}) as in line 7 of Algorithm 13; we assume further that the orders of g at the endpoints b_i, a_{i+1} are odd. If the number of these roots is odd, then the update operation to f_{pos} in Algorithm 13 leaves one factor $X - r$ as part of the nonnegative factor of f. In such a case, we have two possibilities:

1. f is divisible by $X - b_i$: Observing line 11 in Algorithm 13, we see that the nonnegative part of f is divisible by $(X - b_i)^{\mathrm{ord}_{b_i}(f)}(X - \max(\mathcal{B}_i))$. Since the orders of g at the endpoints are odd, $\mathrm{ord}_{b_i}(f)$ is also odd .

2. f is not divisible by $X - b_i$: Since the number of real roots between b_i and a_{i+1} is odd and $\mathrm{ord}_{b_i}(f) = 1$, then $\mathrm{ord}_{a_{i+1}}(f) > 0$. Consequently, observing line 13,

f is divisible by $(X - \min(\mathcal{B}_i))(X - a_{i+1})^{\mathrm{ord}_{a_{i+1}}(f)}$. Since the orders of g at the endpoints are odd, $\mathrm{ord}_{b_i}(f)$ is also odd.

From the previous observation, we define the i^{th} in-between factor of f, denoted $f_{(b_i,a_{i+1})}$,

$$f_{(b_i,a_{i+1})} := \begin{cases} 1 & \text{if } \mathcal{B}_i \text{ is even} \\ (X - b_i)^{\mathrm{ord}_{b_i}(f)}(X - \max(\mathcal{B}_i)) & \text{if } \mathrm{ord}_{b_i}(f) > 0 \\ (X - \min(\mathcal{B}_i))(X - a_{i+1})^{\mathrm{ord}_{a_{i+1}}(f)} & \text{otherwise} \end{cases} \tag{5.3}$$

For the rest of this section, we prove that the factor i^{th} in-between of f belongs to $\mathrm{QM}(g)$ by discussing a method to compute a certificate of it. For the latter, we assume the i^{th} in-between factor corresponds to the second case in Equation (5.3), i.e., it is of the form $(X - b_i)^{\mathrm{ord}_{b_i}(f)}(X - r)$ since the first case is trivial and the third case is obtained from the second one by involution. Our main result uses and generalizes Lemma 2.21.

Theorem 5.13 *Let $r \in \mathbb{A}$ be such that $b_i < r < a_{i+1}$ where b_i (resp. a_{i+1}) is a right (resp. left) end point of an interval in $\mathcal{S}(g)$. We have $(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r) \in \mathrm{QM}(g)$.*

Proof. Using Lemma 2.21, we obtain a certificate of $(X - b_i)(X - r) = \sigma_0 + \sigma_1 \cdot (X - b_i)(X - a_{i+1})$. Since g does not have isolated points of type A, we multiply the sums of squares $(X - b_i)^{\mathrm{ord}_{b_i}(g)-1}(X - a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)-1}$ on both sides of the previous equation and obtain a certificate of $(X - b_i)^{\mathrm{ord}_{b_i}(g)}(X - r)(X - a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)-1}$ in $\mathrm{QM}((X - b_i)^{\mathrm{ord}_{b_i}(g)}(X - a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)})$. Since g has no isolated points of type A, $(X - b_i)^{\mathrm{ord}_{b_i}(g)}(X - a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)}$ divides g and is non-negative over $\mathcal{S}(g)$. Hence, using the SOSBasicLemma algorithm with inputs

$(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)}$ and $\frac{g}{(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)}}$, we compute a certificate of $(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)}$ in $\mathrm{QM}(g)$. We obtain a certificate of $(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r)(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)-1}$ in $\mathrm{QM}(g)$ by lifting the certificate of $(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)}$ in $\mathrm{QM}(g)$. Since $(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)-1}$ is a squared term, we use the BasicLemma algorithm with input $(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r)$, $(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)-1}$, and g to obtain σ, τ that are strictly positive over $\mathcal{S}(g)$ so that $1=\sigma(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r)+\tau(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)-1}$. We use the Certificate algorithm to compute certificates of σ, τ in $\mathrm{QM}(g)$. Multiplying $(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r)$ to both sides of the previous equation we obtain $(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r) = ((X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r))^2\sigma + (X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r)(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)-1}\tau$. We distribute and rearrange the sums of squares multipliers in the certificates of $(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r)(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)-1}$ and τ to obtain a certificate of its product in $\mathrm{QM}(g)$. We multiply $((X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r))^2$ by the certificate of σ in $\mathrm{QM}(g)$ to obtain a certificate of $(X-b_i)^{\mathrm{ord}_{b_i}(g)}(X-r)$ in $\mathrm{QM}(g)$. □

A similar construction works to compute a certificate of the form $(X-r)(X-a_{i+1})^{\mathrm{ord}_{a_{i+1}}(g)}$ in $\mathrm{QM}(g)$.

Example 5.14 Let us consider the generator $g=-(X+2)(X+1)^3(X-1)(X-2)$ and the input polynomial to be $f=X(X-1)$. We will follow the procedure in the proof of Theorem 5.13 to compute a certificate of f in $\mathrm{QM}(g)$.

First, we compute a certificate of $X(X-1)$ in $\mathrm{QM}((X+1)(X-1))$. We use Lemma 2.21 to obtain $X(X-1) = \frac{1}{2}(X-1)^2 + \frac{1}{2}\cdot(X+1)(X-1)$. Now, we multiply the square term $(X+1)^2$ on both sides of the previous equation to obtain $(X+1)^2X(X-1) = \frac{1}{2}((X+1)(X-1))^2 + \frac{1}{2}\cdot(X+1)^2(X+1)(X-1)$.

We compute a certificate of $(X+1)^3(X-1)$ in $\mathrm{QM}(g)$ using SOSBasicLemma using inputs $(X+1)^3(X-1)$ and $\frac{g}{(X+1)^3(X-1)}$. We obtain $(X+1)^3(X-1) = s_0 + s_1\cdot g$ where

- $s_0 = ((X+1)^3(X-1))^2(\frac{1222\left(X^2-4\right)^2\left(X+\frac{1999}{1000}\right)^4}{5562249}$
  $-\ \frac{2X}{27}\ -\ \frac{7097(X-2)(X-1)(X+1)^3(X+2)(2X-5)}{3291219}\ +\ (X\ -\ 2)(X\ -\ 1)(X\ +\ 1)^3(X\ +$
  $2)\left(\left(\frac{10259040(X-2)(X-1)(X+2)(X+1)^3}{1220688757}+\frac{79372768}{1220688757}\right)^4+\frac{59\left(X+\frac{1999}{1000}\right)^2}{2201088}\right)+\frac{5}{27})$

- $s_1 = -\frac{2X^3}{27}+\frac{X^2}{27}-\frac{7097(X-1)^2(2X-5)(X+1)^6}{3291219}$
  $+(X-1)^2\left(\left(\frac{10259040(X-2)(X-1)(X+2)(X+1)^3}{1220688757}+\frac{79372768}{1220688757}\right)^4+\frac{59\left(X+\frac{1999}{1000}\right)^2}{2201088}\right)(X+1)^6$
  $+\ \frac{1222(X-2)(X-1)\left(X+\frac{1999}{1000}\right)^4(X+2)(X+1)^3}{5562249}+\frac{2X}{27}+\frac{8}{27}$

We lift the certificate of $(X+1)^2X(X-1)$ to $\mathrm{QM}(g)$ with the above certificate as $(X+1)^2X(X-1) = \frac{1}{2}((X+1)(X-1))^2+\frac{1}{2}\cdot(s_0+s_1\cdot g) = \frac{1}{2}((X+1)(X-1))^2+\frac{1}{2}s_0+\frac{1}{2}s_1\cdot g$.

Finally, to obtain a certificate of f in $\mathrm{QM}(g)$, we use the BasicLemma algorithm to compute σ, τ strictly positive over $\mathcal{S}(g)$ such that $1 = \sigma(X+1)^2+\tau f$. We obtain:

- $\sigma = \frac{101((X-\frac{401}{202})^2+\frac{799}{202^2})((-20-X-X^2+X^3+X^4)^{16})}{52428800000000000000000000}(-(-20-X-X^2+X^3+X^4))$

- $\tau\ =\ \frac{3}{4}X\ +\ \frac{5}{4}\ -\ \frac{101}{400}(X\ +\ 1)^2\ +\ \frac{1}{20}(-\frac{3}{4}X\ +\ 1\ +\ \frac{101}{400}X(X\ -\ 1))(X\ +$
  $1)^4\sum_{i=0}^{16}(1-\frac{1}{20}(X+1)^2X(X-1))^i$

We check τ is a nonnegative polynomial over $\mathbb{R}$, so it can be decomposed into sums of squares. On the other hand, σ is not nonnegative over $\mathbb{R}$ as $\mathcal{S}(\sigma) \approx [-2.50782, 2.04944]$ but only strictly positive over $\mathcal{S}(g)$; hence, we compute a certificate of σ in $\mathrm{QM}(g)$ using the Certificate algorithm. We obtain $\sigma = s_2 + s_3 \cdot g$ where

- $s_2 = \sigma - (\frac{g-16}{18})^{340}g$

- $s_3 = (\frac{g-16}{18})^{340}$

Multiplying f by both sides to $1 = \sigma(X+1)^2 + \tau f$ we obtain

$$\begin{aligned} f &= f^2\tau + \sigma(X+1)^2X(X-1) \\ &= f^2\tau + (s_2 + s_3 \cdot g)(X+1)^2X(X-1) \quad \text{[By lifting } \sigma \in \mathrm{QM}(g)\text{]} \\ &= f^2\tau + (s_2 + s_3 \cdot g)(\frac{1}{2}((X+1)(X-1))^2 + \frac{1}{2}s_0 + \frac{1}{2}s_1 \cdot g) \\ &\quad \text{[By lifting } (X+1)^2X(X-1) \in \mathrm{QM}(g)\text{]} \\ &= (f^2\tau + \frac{1}{2}((X+1)(X-1))^2 s_2 + \frac{1}{2}s_0s_2 + \frac{1}{2}s_1s_3g^2) \\ &+ (\frac{1}{2}((X+1)(X-1))^2 s_3 + \frac{1}{2}s_0s_3 + \frac{1}{2}s_1s_2) \cdot g \end{aligned} \tag{5.4}$$

Example 5.15 Let us consider the generator $g = -(X+2)(X+1)^3X^2(X-1)(X-2)$ and the input polynomial to be $f = (X - \frac{1}{2})(X-1)$. We will follow the procedure in the proof of Theorem 5.13 to compute a certificate of f in $\mathrm{QM}(g)$. First, we compute a certificate of f in $\mathrm{QM}(g_1)$ where $g_1 = -(X+2)(X+1)^3(X-1)(X-2)$, then we compute the certificate of f in $\mathrm{QM}(g)$ using Theorem 5.10.

We compute a certificate of $(X - \frac{1}{2})(X-1)$ in $\mathrm{QM}((X+1)(X-1))$. We use Lemma 2.21 to obtain $(X - \frac{1}{2})(X-1) = \frac{3}{4}(X-1)^2 + \frac{1}{4} \cdot (X+1)(X-1)$. Now, we multiply the square term $(X+1)^2$ on both sides of the previous equation to obtain $(X+1)^2(X - \frac{1}{2})(X-1) = \frac{3}{4}((X-1)(X+1))^2 + \frac{1}{4} \cdot (X+1)^3(X-1)$.

We use the certificate $(X+1)^3(X-1) = s_0 + s_1 \cdot g_1$ obtained in Example 5.14. We lift the certificate of $(X+1)^2(X-\frac{1}{2})(X-1)$ to $\mathrm{QM}(g_1)$ with the above certificate as $(X+1)^2(X-\frac{1}{2})(X-1) = \frac{3}{4}((X-1)(X+1))^2 + \frac{1}{4} \cdot (s_0 + s_1 \cdot g_1) = \frac{3}{4}((X-1)(X+1))^2 + \frac{1}{4}s_0 + \frac{1}{4}s_1 \cdot g_1$.

To obtain a certificate of f in $\mathrm{QM}(g_1)$, we use the BasicLemma algorithm to compute σ, τ strictly positive over $\mathcal{S}(g_1)$ such that $1 = \sigma(X+1)^2 + \tau f$. We obtain:

- $\sigma = \frac{101(29+X+3X^2-X^3-2X^4)^{12}}{573956280000000000000}(X - \frac{1}{404}(1143 - (109801)^{\frac{1}{2}}))(X - \frac{1}{404}(1143 +$

$(109801)^{\frac{1}{2}}))$

- $\tau = \frac{7}{18}X + \frac{13}{18} - \frac{101}{1080}(X+1)^2 + \frac{1}{15}(-\frac{7}{18}X + \frac{23}{36} + \frac{101}{1080}(X-\frac{1}{2})(X-1))(X+1)^4(\sum_{i=0}^{11}(1-\frac{1}{15}(X+1)^2(X-\frac{1}{2})(X-1))^i)$

We use the Certificate algorithm to compute certificates of σ, τ in $\mathrm{QM}(g_1)$. We obtain $\sigma = \sigma_2 + \sigma_3 \cdot g_1, \tau = \tau_2 + \tau_3 \cdot g_1$ where

- $\sigma_2 = \frac{101(29+X+3X^2-X^3-2X^4)^{10}}{573956280000000000000}((\frac{1}{404}(-1143 - (109801)^{\frac{1}{2}}) + X)(\frac{1}{404}(-1143 + (109801)^{\frac{1}{2}}) + X)(29 + X + 3X^2 - X^3 - 2X^4)^2 - (\frac{4782969}{459986536544739960976801}(-2-X)(-2+X)(-1+X)(1+X)^3(16+(-2+X)(-1+X)(1+X)^3(2+X))^{14}))$

- $\sigma_3 = \frac{101(29+X+3X^2-X^3-2X^4)^{10}}{573956280000000000000}$
$(\frac{4782969}{459986536544739960976801}(16+(-2+X)(-1+X)(1+X)^3(2+X))^{14})$

- $\tau_2 = \frac{13}{18} + \frac{7X}{18} - (\frac{101}{1080}(1+X)^2)$
$- \frac{((-2-X)(-2+X)(-1+X)(1+X)^3(-16+(-2-X)(-2+X)(-1+X)(1+X)^3)^{14})}{4139545122369384765625}$
$+ \frac{1}{15}(\frac{23}{36} + (\frac{101}{1080}(-1+X)(-(\frac{1}{2})+X)) - \frac{7X}{18})(1+X)^4\sum_{i=0}^{11}(1-\frac{1}{15}(-1+X)(-(\frac{1}{2})+X)(1+X)^2)^i$

- $\tau_3 = \frac{(-16+(-2-X)(-2+X)(-1+X)(1+X)^3)^{14}}{4139545122369384765625}$

Multiplying f by both sides of $1 = \sigma(X+1)^2 + \tau f$ we obtain

$$
\begin{aligned}
f &= f^2\tau + \sigma(X+1)^2(X-\frac{1}{2})(X-1) \\
&= f^2(\tau_2+\tau_3\cdot g_1) + \sigma(X+1)^2(X-\frac{1}{2})(X-1) \quad [\text{By lifting } \tau \in \mathrm{QM}(g_1)] \\
&= f^2(\tau_2+\tau_3\cdot g_1) + (\sigma_2+\sigma_3\cdot g_1)(X+1)^2(X-\frac{1}{2})(X-1) \\
&[\text{By lifting } \sigma \in \mathrm{QM}(g_1)] \\
&= f^2(\tau_2+\tau_3\cdot g_1) + (\sigma_2+\sigma_3\cdot g_1)(\frac{3}{4}((X-1)(X+1))^2 + \frac{1}{4}s_0 + \frac{1}{4}s_1\cdot g_1) \\
&[\text{By lifting } (X+1)^2(X-\frac{1}{2})(X-1) \in \mathrm{QM}(g_1)] \\
&= (f^2\tau_2 + \frac{3}{4}\sigma_2((X-1)(X+1))^2 + \frac{1}{4}\sigma_2 s_0 + \frac{1}{4}\sigma_3 s_1 g_1^2) \\
&+ (f^2\tau_3 + \frac{1}{4}\sigma_2 s_1 + \frac{3}{4}\sigma_3((X-1)(X+1))^2 + \frac{1}{4}\sigma_3 s_0)\cdot g_1
\end{aligned}
\tag{5.5}
$$

Hence, $f = s_2 + s_3\cdot g_1$ where $s_2 = f^2\tau_2 + \frac{3}{4}\sigma_2((X-1)(X+1))^2 + \frac{1}{4}\sigma_2 s_0 + \frac{1}{4}\sigma_3 s_1 g_1^2$ and $s_3 = f^2\tau_3 + \frac{1}{4}\sigma_2 s_1 + \frac{3}{4}\sigma_3((X-1)(X+1))^2 + \frac{1}{4}\sigma_3 s_0$. Finally, we compute a certificate of f in $\mathrm{QM}(g)$ from the above certificate of f in $\mathrm{QM}(g_1)$ using Theorem 5.10. For the latter, we multiply X^2 by the certificate of f in $\mathrm{QM}(g_1)$, and we obtain $X^2 f = X^2 s_2 + s_3\cdot g$. We use the BasicLemma algorithm with inputs X^2, f and $\{g\}$ to compute ϕ, ψ strictly positive over $\mathcal{S}(g)$ so that $1 = \phi X^2 + \psi f$. We obtain

- $\phi = -6X + 7 + \frac{101}{25}(X-\frac{1}{2})(X-1) - \frac{3}{100}(-6X+7+\frac{101}{25}(X-\frac{1}{2})(X-1))X^2\sum_{i=0}^{138}(1-\frac{3}{100}X^2(X-\frac{1}{2})(X-1))^i(X-\frac{1}{2})(X-1)$
- $\psi = 6X + 2 - \frac{101}{25}X^2 + \frac{3}{100}(-6X+7+\frac{101}{25}(X-\frac{1}{2})(X-1))X^4\sum_{i=0}^{138}(1-\frac{3}{100}X^2(X-\frac{1}{2})(X-1))^i$

We check ψ is a nonnegative polynomial over $\mathbb{R}$, hence it is a sums of squares. On the other hand, $\mathcal{S}(\phi) \approx [-2.06161, 2.81567]$. We use the Certificate algorithm to compute a certificate of ϕ in $\mathrm{QM}(g)$. We obtain $\phi = \phi_0 + \phi_1\cdot g$ where

- $\phi_0 = \frac{101}{(17422457186352049329324779900506532426547 2*(10^{280}))}(\frac{799}{163216} + (-\frac{603}{404} + X)^2)(-200 + 3X^2 - 9X^3 + 6X^4)^{138}(200 - 3X^2 + 9X^3 - 6X^4 + (-2 + X)(-1 + X) * X^2 * (1 + X)^3(2 + X)(\frac{9}{10} + \frac{1}{(838950473)^2}(754571078 + 33640755(-2 + X)(-1 + X)X^2(1 + X)^3(2 + X))^2))$
- $\phi_1 = \frac{101}{(17422457186352049329324779900506532426547 2*(10^{280}))}(\frac{799}{163216} + (-\frac{603}{404} + X)^2)(-200 + 3X^2 - 9X^3 + 6X^4)^{138}(\frac{9}{10} + \frac{1}{(838950473)^2}(754571078 + 33640755(-2 + X)(-1 + X) * X^2 * (1 + X)^3(2 + X))^2)$

Multiplying f to both sides of the equation $1 = \phi X^2 + \psi f$ we obtain $f = \phi X^2 f + f^2\psi$. Lifting the certificates of ϕ and $X^2 f$ in $\mathrm{QM}(g)$ we obtain

$$\begin{aligned} f &= \phi X^2 f + f^2\psi \\ &= (\phi_0 + \phi_1 \cdot g)(X^2 s_2 + s_3 \cdot g) + f^2\psi \\ &= (s_2 X^2\phi_0 + s_3 g^2\phi_1 + f^2\psi) + (s_3\phi_0 + s_2 X^2\phi_1) \cdot g \end{aligned}$$

Example 5.16 Let us consider the generator $g = -(X + \frac{33}{16})(X + 1)(X - 1)^3(X - \alpha)$, where α is a radical number $\alpha = \frac{1}{48}(65 - \frac{65*34^{\frac{2}{3}}}{(14477+147\sqrt{10131})^{\frac{1}{3}}} + (34(14477 + 147\sqrt{10131}))^{\frac{1}{3}})$, and the input polynomial to be $f = (X + 1)X$. We will follow the procedure in the proof of Theorem 5.13 to compute a certificate of f in $\mathrm{QM}(g)$.

First, we compute a certificate of $(X + 1)X$ in $\mathrm{QM}((X + 1)(X - 1))$. We use Lemma 2.21 to obtain $(X + 1)X = \frac{1}{2}(X + 1)^2 + \frac{1}{2} \cdot (X + 1)(X - 1)$. Now, we multiply the square term $(X - 1)^2$ on both sides of the previous equation to obtain $(X + 1)X(X - 1)^2 = \frac{1}{2}((X + 1)(X - 1))^2 + \frac{1}{2} \cdot (X + 1)(X - 1)^3$.

Now, we obtain a certificate of $(X + 1)(X - 1)^3$ in $\mathrm{QM}(g)$. For the latter, we use the SOSBasicLemma algorithm with inputs $(X + 1)(X - 1)^3$ and $\frac{g}{(X+1)(X-1)^3}$ to obtain σ, τ sums of squares such that $1 = \sigma(X + 1)(X - 1)^3 + \tau\frac{g}{(X+1)(X-1)^3}$. We obtain

- $\sigma = \frac{65536}{2000033}$
- $\tau = \frac{442630144X^2}{49(300648099+10469680(16\alpha)-223665(16\alpha)^2)} + \frac{256(2145+8X(-65+(16\alpha))-65(16\alpha)+(16\alpha)^2)^2}{2000033(2145-65(16\alpha)+(16\alpha)^2)}$

We get a certificate of $(X+1)(X-1)^3$ in $\mathrm{QM}(g)$ multiplying $(X+1)(X-1)^3$ on both sides of equation $1 = \sigma(X+1)(X-1)^3 + \tau\frac{g}{(X+1)(X-1)^3}$. We obtain $(X+1)(X-1)^3 = \sigma((X+1)(X-1)^3)^2 + \tau \cdot g$. This certificate is used to lift the previous certificate obtained for $(X+1)X(X-1)^2$ to a certificate of $\mathrm{QM}(g)$; we have $(X+1)X(X-1)^2 = \frac{1}{2}((X+1)(X-1))^2 + \frac{1}{2}(\sigma((X+1)(X-1)^3)^2 + \tau \cdot g) = \frac{1}{2}((X+1)(X-1))^2 + \frac{1}{2}\sigma((X+1)(X-1)^3)^2 + \frac{1}{2}\tau \cdot g$.

Finally, to obtain a certificate of f in $\mathrm{QM}(g)$, we use the BasicLemma algorithm to compute σ_2, τ_2 strictly positive over $\mathcal{S}(g)$ such that $1 = \sigma_2 f + \tau_2(X-1)^2$. We obtain

- $\sigma_2 = \frac{(19-17X+4X^2)(-46+X-X^2-X^3+X^4)^{18}}{46^{19}*2^2*3} \cdot (46 - X + X^2 + X^3 - X^4)$
- $\tau_2 = -\frac{1}{3}X(X+1) + \frac{1}{4}(4+3X) + \frac{1}{46}(\frac{1}{4}(5-3X) + \frac{1}{3}(X-1)^2)((X+1)X)^2 \sum_{i=0}^{18}(1 - \frac{1}{46}(X-1)^2(X+1)X)^i$

We check τ_2 is a nonnegative polynomial over $\mathbb{R}$, so it can be decomposed into sums of squares. On the other hand, σ_2 is not nonnegative over $\mathbb{R}$ as $\mathcal{S}(\sigma_2) \approx [-2.50083, 2.97327]$ but only strictly positive over $\mathcal{S}(g)$; hence, we compute a certificate of σ_2 in $\mathrm{QM}(g)$ using the Certificate algorithm. We obtain $\sigma_2 = s_0 + s_1 \cdot g$ where

- $s_0 = \frac{(19-17X+4X^2)(-46+X-X^2-X^3+X^4)^{18}}{46^{19}*2^2*3}(46 - X + X^2 + X^3 - X^4 - \frac{432}{887}g)$
- $s_1 = \frac{432}{887}\frac{(19-17X+4X^2)(-46+X-X^2-X^3+X^4)^{18}}{46^{19}*2^2*3}$

Multiplying f by both sides to $1 = \sigma_2 f + \tau_2(X-1)^2$ we obtain

$$\begin{aligned}
f &= f^2\sigma_2 + \tau_2(X-1)^2(X+1)X \\
&= f^2(s_0 + s_1 \cdot g) + \tau_2(X-1)^2(X+1)X \quad \text{[By lifting } \sigma_2 \in \mathrm{QM}(g)\text{]} \\
&= f^2(s_0 + s_1 \cdot g) + \tau_2(\frac{1}{2}((X+1)(X-1))^2 + \frac{1}{2}\sigma((X+1)(X-1)^3)^2 + \frac{1}{2}\tau \cdot g) \\
&\quad \text{[By lifting } (X+1)X(X-1)^2 \in \mathrm{QM}(g)\text{]} \\
&= (f^2 s_0 + \frac{1}{2}\tau_2((X+1)(X-1))^2 + \frac{1}{2}\tau_2\sigma((X+1)(X-1)^3)^2) \\
&\quad + (f^2 s_1 + \frac{1}{2}\tau_2\tau) \cdot g
\end{aligned} \tag{5.6}$$

5.3.3 Certificates computation with MonogenicCert

In this section, we combine the results from the previous subsections into an algorithm to compute certificates of the members in Archimedean monogenic quadratic modules. We use Algorithm 14 to combine the certificates of products in $\mathrm{QM}(g)$.

Algorithm 14: Combine certificate of products

CombineCert$(\sigma_0, \sigma_1, \sigma_2, \sigma_3, g)$

Input: $\sigma_0, \sigma_1, \sigma_2, \sigma_3, g \in \mathbb{A}[X]$

Output: σ, τ

Requires: $\sigma_0, \sigma_1, \sigma_2, \sigma_3 \in \sum \mathbb{A}[X]^2$

Ensures: $\sigma, \tau \in \sum \mathbb{A}[X]^2, f_1 f_2 = \sigma + \tau \cdot g$ where $f_1 = \sigma_0 + \sigma_1 \cdot g$ and $f_2 = \sigma_2 + \sigma_3 \cdot g$

1 $\sigma := \sigma_0\sigma_2 + \sigma_1\sigma_3 g^2$

2 $\tau := \sigma_0\sigma_3 + \sigma_1\sigma_2$

3 **return** σ, τ

The following algorithm finds the factors of a generator g with even multiplicities that are coprime with input polynomial f. In particular, this construction is necessary to guaranty the conditions in Section 5.3.2 so that the in-between factors of the nonnegative polynomial $f_{non-neg}$ have an in-between interval $[b_i, a_{i+1}]$ and $\mathrm{ord}_{b_i}(g), \mathrm{ord}_{a_{i+1}}(g)$ that are both odd.

Algorithm 15: Compute a squared divisor of g disjoint to $f_{non-neg}$

MonogenicDivisor($f_{non-neg}, g$)

Input: $f_{non-neg}, g \in \mathbb{A}[X]$

Output: $p \in \mathbb{A}[X]$

Requires: $\mathcal{S}(g) = \bigcup_{i=0}^{k} [a_i, b_i]$ is bounded non-empty, and $f_{non-neg}$ is nonnegative over $\mathcal{S}(g)$ without a strictly positive factor

Ensures: $p|g$, $p \in \sum \mathbb{A}[X]^2$, and $(p, f) = 1$

```
p := 1
for i = 0 to k − 1 do
    Let f_(b_i,a_{i+1}) be the i^th in-between factor of f_non−neg
    if f_(b_i,a_{i+1}) is of the form (X − b_i)^ord_{b_i}(f)(X − r) then
        Let c_1 < c_2 < ··· < c_m such that g(c_j) = 0 and r < c_j
    if f_(b_i,a_{i+1}) is of the form (X − r)(X − a_{i+1})^ord_{a_{i+1}}(f) then
        Let c_1 > c_2 > ··· > c_m such that g(c_j) = 0 and r > c_j
    j := 1
    while ord_{c_j}(g) ∈ 2ℕ and ε_{c_j}(g) = −1 do
        p := p ∗ (X − c_j)^ord_{c_j}(g)
        j := j + 1
        if j > m then
            break
return p
```

In Algorithm 16, we present a solution to compute certificates in Archimedean

monogenic quadratic modules:

Algorithm 16: Compute a certificate of membership of an input polynomial f in $\mathrm{QM}(g)$

MonogenicCert(f, g)

Input: $f, g \in \mathbb{A}[X]$

Output: $\sigma_2, \sigma_3 \in \mathbb{A}[X]$

Requires: $\mathcal{S}(g) = \bigcup_{i=0}^{k} [a_i, b_i]$ is bounded non-empty, and $f \in \mathrm{QM}(g)$

Ensures: $\sigma_2, \sigma_3 \in \sum \mathbb{A}[X]^2$ and $f = \sigma_2 + \sigma_3 \cdot g$

Let g_1 be the reduced generator g

Let f_1 be the reduced input polynomial of f w.r.t. g_1, $\sigma_f := \frac{f}{f_1}$

Let $\mathcal{A}$ be the roots of g_1 with even multiplicities that are zeros of f_1

Let $f_{\overline{\mathcal{A}}} := \frac{f_1}{\prod_{a \in \mathcal{A}} (X-a)^{\mathrm{ord}_a(g_1)}}, g_{\overline{\mathcal{A}}} := \frac{g_1}{\prod_{a \in \mathcal{A}} (X-a)^{\mathrm{ord}_a(g_1)}}$

$f_{pos} := \mathsf{StrictPos}(f_{\overline{\mathcal{A}}}, \mathcal{S}(g_{\overline{\mathcal{A}}})), f_{non-neg} := \frac{f_{\overline{\mathcal{A}}}}{f_{pos}}$

$p := \mathsf{MonogenicDivisor}(f_{non-neg}, g_{\overline{\mathcal{A}}})$

$g_{\overline{\mathcal{A}}} := \frac{g_{\overline{\mathcal{A}}}}{p}, f_{pos} := f_{pos}\, \mathsf{StrictPos}(f_{non-neg}, \mathcal{S}(g_{\overline{\mathcal{A}}})), f_{non-neg} := \frac{f_{\overline{\mathcal{A}}}}{f_{pos}}$

$\sigma_0 := 1, \sigma_1 := 0$

for $i = 0$ *to* $k-1$ **do**

  Let $f_{(b_i, a_{i+1})}$ be the i^{th} in-between factor of $f_{non-neg}$

  if $f_{(b_i, a_{i+1})} \neq 1$ **then**

    Use Theorem 5.13 to compute a certificate of $f_{(b_i, a_{i+1})} (= \sigma_2 + \sigma_3 \cdot g_{\overline{\mathcal{A}}})$ in $\mathrm{QM}(g_{\overline{\mathcal{A}}})$

    $\sigma_0, \sigma_1 := \mathsf{CombineCert}(\sigma_0, \sigma_1, \sigma_2, \sigma_3, g_{\overline{\mathcal{A}}})$

    Let $f_{non-neg} := \frac{1}{f_{(b_i, a_{i+1})}} f_{non-neg}$

if $f_{non-neg} \neq 1$ **then**

  Use the SOSBasicLemma algorithm with input $f_{non-neg}$ and $\frac{g_{\overline{\mathcal{A}}}}{f_{non-neg}}$ to compute sums of squares σ_2, σ_3 such that $1 = \sigma_2 f_{non-neg} + \sigma_3 \frac{g_{\overline{\mathcal{A}}}}{f_{non-neg}}$, hence $f_{non-neg} = \sigma_2 f_{non-neg}^2 + \sigma_3 \cdot g_{\overline{\mathcal{A}}}$

  $\sigma_0, \sigma_1 := \mathsf{CombineCert}(\sigma_0, \sigma_1, \sigma_2 f_{non-neg}^2, \sigma_3, g_{\overline{\mathcal{A}}})$

if $f_{pos} \neq 1$ **then**

  Use the Certificate algorithm to compute certificates of $f_{pos} (= \sigma_2 + \sigma_3 \cdot g_{\overline{\mathcal{A}}})$ in $\mathrm{QM}(g_{\overline{\mathcal{A}}})$

  $\sigma_0, \sigma_1 := \mathsf{CombineCert}(\sigma_0, \sigma_1, \sigma_2, \sigma_3, g_{\overline{\mathcal{A}}})$

$\sigma_0 := \prod_{a \in \mathcal{A}} (X-a)^{\mathrm{ord}_a(g_1)} \sigma_0$

$\sigma_0 := \sigma_f \sigma_0$

$\sigma_1 := \sigma_f \sigma_1$

Use Theorem 5.10 with inputs σ_0, σ_1 to compute σ_2, σ_3 that removes the squared term p

Use the SOSBasicLemma algorithm with inputs g_1 and $\frac{g}{g_1}$ to compute sums of squares σ_4, σ_5 such that $1 = \sigma_4 g_1 + \sigma_5 \frac{g}{g_1}$, hence $g_1 = \sigma_4 g_1^2 + \sigma_5 \cdot g$

$\sigma_2 := \sigma_2 + \sigma_3 \sigma_4 g_1^2$

$\sigma_3 := \sigma_3 \sigma_5$

return σ_2, σ_3

5.3.4 Examples

In this section, we summarize the results in the previous section using algorithmic descriptions of our methods.

Example 5.17 Let us consider $g = -(X-a_1)^2(X-a_2)^2$ where $a_1 < a_2 \in \mathbb{A}$. We will compute certificates for the generalized natural generators $g_1 = (X-a_1)^3, g_2 = (X-a_1)^3(X-a_2)^3, g_3 = -(X-a_2)^3$ of $\mathrm{QM}(g)$.

For g_1, we notice that the only common isolated point is a_1, and hence the transform problem becomes finding certificates of $X-a_1 \in \mathrm{QM}(-(X-a_2)^2)$. Symbolically, we have the following certificate $X - a_1 = \frac{a_2-a_1}{2} + \frac{(X-a_1)^2}{2(a_2-a_1)} + \frac{1}{2(a_2-a_1)} \cdot (-(X-a_2)^2)$. Thus, multiplying $(X-a_1)^2$ to both sides of the previous equation, we obtain $g_1 = \frac{(a_2-a_1)(X-a_1)^2}{2} + \frac{(X-a_1)^4}{2(a_2-a_1)} + \frac{1}{2(a_2-a_1)} \cdot g$. Similarly, for g_3 we obtain $g_3 = \frac{(a_2-a_1)(X-a_2)^2}{2} + \frac{(X-a_2)^4}{2(a_2-a_1)} + \frac{1}{2(a_2-a_1)} \cdot g$.

For g_2, we notice that the common isolated points are a_1 and a_2. After the reduction, we need to compute a certificate of $(X-a_1)(X-a_2) \in \mathrm{QM}(-1)$. This is easy using the identity $f = (\frac{f+1}{2})^2 + (\frac{f-1}{2})^2 \cdot (-1)$. Hence, we have $(X-a_1)(X-a_2) = (\frac{(X-a_1)(X-a_2)+1}{2})^2 + (\frac{(X-a_1)(X-a_2)-1}{2})^2 \cdot (-1)$. Multiplying $(X-a_1)^2(X-a_2)^2$ to both sides of the previous equation, we obtain $g_2 = (X-a_1)^2(X-a_2)^2(\frac{(X-a_1)(X-a_2)+1}{2})^2 + (\frac{(X-a_1)(X-a_2)-1}{2})^2 \cdot g$.

Example 5.18 Let us consider $g = -(X+1)X^2(X-1)$ and $f = -(X+2)X^2(X-2)$. The common isolated point of type A is at $x = 0$, so the problem reduces to computing a certificate of $-(X+2)(X-2) \in \mathrm{QM}(-(X+1)(X-1))$. We notice that the transformed input polynomial is strictly positive over $[-1, 1]$. Hence, we use the Certificate algorithm to compute this certificate; we obtain $-(X+2)(X-2) = s_0 + s_1 \cdot (-(X+1)(X-1))$ where

- $s_0 = \frac{30509247835754489220357674256936214671 1X^4}{9967541089866907346920569907474456250 00}$
  $+ \frac{3X^2(21610648533130655389439+69482441530166146830 00X)^2}{21645393633411848416500086307945700996145000 00}$
  $+ \frac{(-42185392614385+41573999385000X+37765651445754X^2)^2}{88291725528740690298250000 0}$

- $s_1 = \frac{47210}{23743}(X + \frac{999}{1000})^2$

Multiplying X^2 to both sides of the previous equation, we obtain $f = X^2 s_0 + s_1 \cdot g$.

Example 5.19 Consider a generator $g = -(X+3)(X+2)X^2(X-2)(X-3)$ and the input polynomial $f = (X+1)X^2(X-1)$. The common isolated point of even multiplicity is at $x = 0$, therefore, we work on the problem of finding certificates of $f_1 := (X+1)(X-1) \in \mathrm{QM}(g_1)$ where $g_1 := -(X+3)(X+2)(X-2)(X-3)$. The latter is a strictly positive polynomial over $S_1 = \mathcal{S}(g_1) = [-3,-2] \cup [2,3]$. Using the Certificate algorithm, we obtain $(X+1)(X-1) = \frac{1}{875301}(1459587 + 32147X^2 + 64858X^4) + \frac{64858}{875301} \cdot g_1$. Multiplying X^2 by both sides of the previous equation, we obtain $f = \frac{X^2}{875301}(1459587 + 32147X^2 + 64858X^4) + \frac{64858}{875301} \cdot g$.

Example 5.20 In this example, we compare the results of our approach with Shang et. al's general method. Consider $G = g$ where $g = -(X+1)^2(X-1)^2$ and the input polynomial $f = (X+1)^3$.

First, we compute a certificate for this problem using our proposed method. The common isolated point of even multiplicity is -1. Hence, we work on the problem of finding a certificate of $f_1 := X + 1$ in $\mathrm{QM}(-(X-1)^2)$. Since f_1 is strictly positive over $\mathcal{S}(-(X-1)^2) = \{1\}$, we use Certificate to compute a certificate, we obtain $f_1 = s_0 + s_1 \cdot (-(X-1)^2)$ where

- $s_0 = \frac{19}{10}(-\frac{4}{19}X + 1)^2 + \frac{31}{38}(X)^2$

- $s_1 = \frac{9}{10}$

Multiplying $(X+1)^2$ by the certificate of f_1, we obtain $f = (X+1)^2 s_0 + \frac{9}{10}(X+1)^2 \cdot g$. Notice that in expanded form, $\deg((X+1)^2 s_0) = 4$ and $\deg(\frac{9}{10}(X+1)^2) = 2$.

Now, we use Shang et. al's general method. We fix the input polynomial f at $x = -1$ with $f_1 := f - \frac{1}{4}\frac{(X-1)^2}{(-1-1)^2} \cdot g$. Since f_1 is a sums of squares, we are done. Hence, the certificate obtained is $f = f_1 + \frac{1}{4}\frac{(X-1)^2}{(-1-1)^2} \cdot g$ where

$$\begin{aligned} f_1 = &\frac{41}{16}(-\frac{5}{41}X^3 + \frac{13}{41}X^2 + X + \frac{23}{41})^2 \\ &+ \frac{25}{82}(\frac{3}{25}X^3 + X^2 - \frac{22}{25})^2 + \frac{1}{50}(X^3+1)^2 \end{aligned}$$

Notice that in expanded form, $\deg(f_1) = 6$ and $\deg(\frac{1}{4}\frac{(X-1)^2}{(-1-1)^2}) = 2$.

5.4 Experiments

This section compares the degree of certificates obtained by our method and Shang et. al's general method. This benchmark uses a monogenic generator $g_n := -\prod_{m=1}^{n}(X^2 - m^2)^2$ whose semialgebraic set consists of only isolated points; the semialgebraic set of g_n is $\bigcup_{m=-n}^{n}\{m\}$. We compute certificates for the left generalized natural generator $(X+n)^3$ and the intermediate generalized natural generator $(X+1)^3(X-1)^3$. All experiments were performed on an M1 MacBook Air with 8GB of RAM running macOS.

We report in Table 5.1 the time (in seconds) and degree of the sums of squares term s_0 in the certificate $s_0 + s_1 \cdot g_n$ for the left natural generator. Table 5.2 reports the same information for the intermediate generalized natural generator.

	Shang et. al's general method		MonogenicCert	
n	Degree	Time (seconds)	Degree	Time (Seconds)
1	6	0.009	4	0.008
2	10	0.016	8	0.015
3	14	0.024	12	0.020
4	18	0.045	16	0.032
5	22	0.037	20	0.072
6	26	0.049	24	0.052
7	30	0.062	28	0.074
8	34	0.093	32	0.087
9	38	0.087	36	0.171

Table 5.1: Benchmark for left generalized natural generator $(X+n)^3$

To illustrate the differences between the results, we show the certificates obtained for $n = 3$. Using Shang et. al's general method we obtain the following:

- $s_0 = \frac{7413}{2470}((X+3)(-\frac{4582199}{18946916352}X^6 + \frac{9271979}{8744730624}X^5 + \frac{76315279}{4372365312}X^4 - \frac{192321023}{5829820416}X^3 - \frac{3909944081}{14574551040}X^2 + \frac{1}{6}X + 1))^2$
  $+ \frac{1233081947}{809369600}((X+3)(\frac{22725230}{144270587799}X^6 + \frac{9821461295}{1731247053588}X^5 - \frac{1020935735}{66586425138}X^4 - \frac{13849811725}{88781900184}X^3 + \frac{3389515393}{22195475046}X^2 + X))^2$
  $+ \frac{4071471689322571989137540807 9}{261822230452628412551921664000}((X+3)(\frac{11989004304696284059327204 10}{52929131961193435858788030 5027}X^6 - \frac{12075663412512594339280166965}{15878739588358030757636409150 81}X^5 - \frac{16995776754885852786357971 2930}{15878739588358030757636409150 81}X^4 + \frac{11353451622434581163446949285}{81429433786451439782750816158}X^3 + X^2))^2$
  $+ \frac{9861781910900455545602419496346197755 37}{14805238013721796620782450787641198641 1520}((X+3)$
  $(-\frac{24935858937511421372076406905957750 16}{98617819109004555456024194963461977553 7}X^6 - \frac{71062377988229463621870677026771460514}{98617819109004555456024194963461977553 7}X^5 + \frac{11934090578157953141938586762760193 1172}{98617819109004555456024194963461977553 7}X^4 + X^3))^2$
  $+ \frac{55717115656112579449185987116925024560604744670 63}{21852644538347772302489436526945452664484792048025 60}((X+3)(-\frac{34513064803590339823426330843267847393994334271}{55717115656112579449185987116925024560604744670630}X^6 + \frac{20491196950800695689525390757242949070482493324 1}{22286846262445031779674394846770009802418978682 52}X^5 + X^4))^2$
  $+ \frac{14700151320777857736364673079578001994501182717184959251}{10974499191913429931406831618357309830124508078800753446092800}((X+3)(\frac{71349039428429792585670250290224855998587 18090231056 02}{44100453962333573209094019238734005983503548151554877753}X^6 + X^5))^2$

$+ \frac{1407967417877161712136510784167501469989925078623983804044649}{339311969233862924660836450421601536121333708685582281899854049280000}((X+3)(X^6))^2$

- $s_1 = \frac{(X+2)^2}{474240}$

Using our approach, we obtain the following.

- $s_0 = \frac{295993}{1440}((X+3)(\frac{19440}{295993}X^5 - \frac{58961}{295993}X^4 - \frac{776379}{2367944}X^3 + X^2 + \frac{78930}{295993}X - \frac{4724307}{5919860}))^2$
  $+\frac{1828280059}{473588800}((X+3)(\frac{740267620}{16454520531}X^5 - \frac{274501585}{10969680354}X^4 - \frac{29327648005}{65818082124}X^3 + X + \frac{493844435}{1828280059}))^2$
  $+ \frac{164807382800357}{117009923776000}((X+3)(\frac{37691841745750}{1483266445203213}X^5 - \frac{88126422948670}{1483266445203213}X^4 - \frac{276375357806195}{2966532890406426}X^3 + 1))^2$
  $+ \frac{3823759689133438519}{22782972598321351680}((X+3)(-\frac{3103476070015783114}{19118798445667192595}X^5 + \frac{1671476728056867100}{11471279067400315557}X^4 + X^3))^2$
  $+ \frac{7877019099865908996101}{5946711068540323584748800}((X+3)(-\frac{2345831642781271795560}{7877019099865908996101}X^5 + X^4))^2$
  $+ \frac{214359050447152166037749}{22685815007613817908770880000}((X+3)(X^5))^2$

- $s_1 = \frac{9}{10}$

	Shang et. al's general method		MonogenicCert	
n	Degree	Time (seconds)	Degree	Time (Seconds)
1	8	0.014	8	0.003
2	12	0.020	8	0.090
3	16	0.027	12	0.102
4	20	0.036	16	0.234
5	24	0.055	20	0.369
6	28	0.057	24	1.549
7	32	0.078	28	0.950
8	36	0.107	32	1.524
9	40	0.106	36	2.328

Table 5.2: Benchmark for intermediate generalized natural generator $(X+1)^3(X-1)^3$

Similarly to the previous case, we illustrate the differences between the results for $n = 3$. Using Shang et. al's general method we obtain the following:

- $s_0 = \frac{2752117}{1271808}((X+1)(X-1)(\frac{34444}{2752117}X^6 - \frac{8004375}{35777521}X^4 + X^2 - \frac{9727911}{13760585}))^2 + \frac{71013}{117760}((X+1)(X-1)(\frac{349835}{11078028}X^5 - \frac{1606835}{3834702}X^3 + X))^2 + \frac{863261864477}{3240892979200}((X+1)(X-1)(\frac{1101329539835}{134668850858412}X^6 - \frac{54355980031625}{606009828862854}X^4 + 1))^2 + \frac{473671068125}{7925132606976}((X+1)(X-1)(-\frac{816810183927}{7578737090000}X^5 + X^3))^2 + \frac{2105589010768172011 43}{20038931859086624120832}((X+1)(X-1)(-\frac{19571224243466025 6291}{2105589010768172011430}X^6 + X^4))^2 + \frac{7877267215516599}{241324450355855360000}((X+1)(X-1)(X^5))^2 + \frac{24221143817487318053959759}{45256593631459099530765735936 00}((X+1)(X-1)(X^6))^2$

- $s_1 = \frac{1}{4416}((X-1)(X+2))^2 + \frac{1}{4416}((X+1)(X-2))^2$

Using our approach, we obtain the following.

- $s_0 = 1295((X+1)(X-1)(\frac{5167}{186480}X^4 - \frac{74937}{207200}X^2 + 1))^2 + \frac{137}{80}((X+1)(X-1)(-\frac{175}{1233}X^3 + X))^2 + \frac{99694279}{298368000}((X+1)(X-1)(-\frac{12383610}{99694279}X^4 + X^2))^2 + \frac{1333}{443880}((X+1)(X-1)(X^3))^2 + \frac{40390949}{64601892792}((X+1)(X-1)(X^4))^2$

- $s_1 = 1$

5.5 Summary

In this chapter, we provide the MonogenicCert algorithm that allows us to compute a certificate for a member in a monogenic Archimedean quadratic module. First, we simplify the structure of both the generator and input polynomial by identifying redundant sums of squares; these are sums of squares that can be part of the certificate itself as sums of squares multipliers; hence, the overall simplification reduces the degree of the generator and input polynomial. The key idea of "ignoring" factors with even multiplicities in Section 5.2 provides a bridge between the saturated and general case; the latter stands for some factors with multiplicity 1 while in the general case this might not be the case. In fact, Appendix B elaborates this observation in more detail.

Thanks for the simplifications presented in Section 5.1 and Section 5.2, we were able to use the methods in Chapter 3, particularly Section 3.2.1 to derive a certificate of the nonnegative factor of the input polynomial and the Certificate algorithm to obtain a certificate of the strictly positive factor. Since monogenic quadratic modules are preorderings, these two certificates, corresponding to the nonnegative and strictly positive components, can be combined to produce a certificate for the original input polynomial.

The benchmark discussed in Section 5.4 shows that our method tends to obtain certificates with smaller degrees compared to Shang et. al's general method; however, the timings seem to be a bit longer for our method.

Chapter 6

Certificates Computation in Archimedean Quadratic Modules in $\mathbb{A}[X]$

This chapter introduces the QMCert algorithm to compute a certificate of a member in an Archimedean quadratic module. The literature extensively discusses several algorithmic aspects for these cases, such as membership procedures [4, 81], and equivalent sets of generators [3] known as generalized natural generators, which extend the concept of natural generators introduced in [41, 42]. Not much attention is given to the certificate problem in the general case, or some cases are addressed under specific conditions. For example, in [8] the authors require that the ideals $\langle f \rangle$ and $\langle I : f \rangle$ are coprime where f is the input polynomial and I is a zero-dimensional ideal associated to a set of equality constraints; notice that such conditions force the addition of non-trivial equations in the system as the ideal needs to be zero-dimensional, which implies the class of quadratic modules have associated finite semialgebraic sets, i.e., a finite collection of points. Our proposed solution does not need additional conditions and no constraints are imposed on the input polynomial nor generators.

Section 6.1 generalizes the results in Section 5.1 by extending the simplification of the input polynomial using not only a single generator but multiple generators of an Archimedean quadratic module. In Section 6.2 we introduce notation and concepts that allow us to identify which generators are relevant in the certificate of the input polynomial. Section 6.3 uses the concepts in the previous section to identify monogenic cases within the general case. Section 6.4 describes the intermediate set of generators $H_{f,G}$ that have the property that a certificate of the product of its members is computable. Section 6.5 presents the QMCert algorithm and discusses examples and a comparison with Shang et. al's general method. In Section 6.7 we apply the specialized version of the QMCert algorithm to obtain certificates using the equivalent set of two generators from [81]; these two generators have the smallest degree possible that fully encodes the information from any other Archimedean quadratic module. Section 6.8 concludes this chapter summarizing the main findings.

6.1 Redundant sums of squares with respect to a set of generators

Any polynomial f in $\mathbb{A}[X]$ is a product of powers of linear and quadratic irreducible factors. We define a reduction operation on the input polynomial with respect to the generators G using its signature to simplify the computation of the membership certificate in Archimedean quadratic modules $\mathrm{QM}(G)$. For the latter, Algorithm 17 computes a sums of squares divisor s_0 of f in order to simplify the input polynomial and focus on $\frac{f}{s_0}$; observe that a certificate for f can be constructed from a certificate of $\frac{f}{s_0}$ by multiplying the latter by s_0, i.e., if $\frac{f}{s_0} = \sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i$ then $f = s_0\sigma_0 + \sum_{i=1}^{s} s_0\sigma_i \cdot g_i$. In the rest of the section we prove important properties of this divisor.

Algorithm 17: Reduction of input polynomial with respect to G

$\mathsf{Reduced}_{\text{input}}(f, G)$

Input: $f \in \mathbb{A}[X]$, $G \subseteq \mathbb{A}[X]$

Output: $r \in \mathbb{A}[X]$

Requires: $\mathcal{S}(G)$ is bounded

Ensures: $r \mid f$ and $r \in \sum \mathbb{A}[X]^2$

Assume $\mathcal{S}(G) = \bigcup_{i=0}^{k} [a_i, b_i]$

Assume $f = d \prod_{i=1}^{u} (X - c_i)^{m_i} \prod_{j=1}^{v} ((X - d_j)^2 + e_j^2)^{n_j}$ with $e'_j s \neq 0$

$r := \prod_{j=1}^{v} ((X - d_j)^2 + e_j^2)^{n_j}$

Let $\omega_S(G) := (\kappa_0, \kappa_0^+, \kappa_0^-, \ldots, \kappa_k, \kappa_k^+, \kappa_k^-)$

Let $\omega_S(f) := (l_0, l_0^+, l_0^-, \ldots, l_k, l_k^+, l_k^-)$

for i *from* 1 *to* u **do**

  if $\exists \eta$ *such that* $c_i = a_\eta$ *or* $c_i = b_\eta$ **then**

    if $l_\eta < \infty$ **then**

      $r := r * (X - c_i)^{l_\eta - \kappa_\eta}$

    if $l_\eta^+ < \infty$ **then**

      $r := r * (X - c_i)^{l_\eta^+ - \kappa_\eta^+}$

    if $l_\eta^- < \infty$ **then**

      $r := r * (X - c_i)^{l_\eta^- - \kappa_\eta^-}$

  else

    if m_i *is even* **then**

      $r := r * (X - c_i)^{m_i}$

    else

      $r := r * (X - c_i)^{m_i - 1}$

return r

By inspection of Algorithm 17, the output r is a sums of squares as it is a

product of at most two squared polynomials[1]. We will prove that $\frac{f}{r}$ belongs to $\mathrm{QM}(G)$ whenever $f \in \mathrm{QM}(G)$.

Theorem 6.1 *Let* $G = \{g_1, \ldots, g_s\}$ *be such that* $\mathcal{S}(G)$ *is compact,* $f \in \mathbb{A}[X]$*, and* r *the output of* $\mathsf{Reduced}_{\mathrm{input}}(f, G)$*. If* $f \in \mathrm{QM}(G)$ *then* $\frac{f}{r} \in \mathrm{QM}(G)$*.*

Proof. We prove the main statement using an invariant that $\frac{f}{r}$ belongs to $\mathrm{QM}(G)$ in Algorithm 17.

For the base case, we have $r = \prod_{j=1}^{v} ((X - d_j)^2 + e_j^2)^{n_j}$. The polynomial $\frac{f}{r}$ satisfies the order and sign conditions of Theorem 2.5 as each e_j is not zero, therefore $\frac{f}{r}$ belongs to $\mathrm{QM}(G)$.

For the inductive case, assume that the polynomial $\frac{f}{r}$ is a member of $\mathrm{QM}(G)$ for the $(i-1)^{th}$ iteration of the algorithm. We will prove that $\frac{f}{r}$ is a member of $\mathrm{QM}(G)$ after the i^{th} iteration of the `for` loop.

We have two cases from the `if` statement inside the `for` loop of Algorithm 17. The first case occurs whenever there exists η that $c_i = a_\eta$ or $c_i = b_\eta$, which means that c_i corresponds to an endpoint of $\mathcal{S}(G)$. In such a case, we notice that only one of the elements in the triplets l_i, l_i^+, l_i^- of the signature of a single polynomial $\omega_S(\{f\})$ is non-infinity, while the other two are infinity; this follows from Definition Definition 2.13. Hence, the update operations from line 9 to line 13 in the algorithm decrease the multiplicity of $X - c_i$ in f to the signature of $\mathrm{QM}(G)$. This allows $\frac{f}{r}$ to preserve the sign and order conditions of Theorem 2.5 so $\frac{f}{r}$ belongs to $\mathrm{QM}(G)$.

For the second case, we have that c_i is not an endpoint of $\mathcal{S}(G)$. Hence, the algorithm removes an even power of $X - c_i$ to preserve the order and sign conditions of $\frac{f}{r}$, so that by Theorem 2.5, it is a member of $\mathrm{QM}(G)$. □

[1]This follows from the Brahmagupta-Fibonacci identity $(a^2+b^2)(c^2+d^2) = (ac+bd)^2 + (ad-bc)^2$

We refer to the polynomial $\frac{f}{\mathsf{Reduced}_{\text{input}}(f,G)}$ as the normal form of f with respect to G. From the proof of Theorem 6.1, we observe the following properties:

- the normal form of f consists of products of monomials with multiplicities from the signature of $\mathrm{QM}(G)$, while additional factors (roots that are not end points of $\mathcal{S}(G)$) have multiplicity 1.
- the normal form of f does not have complex conjugate roots.

6.2 Matching, pseudo-matching and pair-matching

In this section, we introduce the concepts of matching, pseudo-matching and pair-matching. The three encode the sign and order conditions at any given point $a \in \mathbb{A}$ for membership in an Archimedean quadratic module of Theorem 2.5; nevertheless, we often use these ideas on the roots of f. Additionally, it helps us to keep track of the generators (or pair of generators) involved in the certificate of the input polynomial.

Definition 6.2 Let $f, g, h \in \mathbb{A}[X]$, and $a \in \mathbb{A}$.

I. f **matches** g at a if

- $\mathrm{ord}_a(f)$ is even and $\epsilon_a(f) = 1$, or
- $0 \leq \mathrm{ord}_a(f) - \mathrm{ord}_a(g)$ is even and $\epsilon_a(f) = \epsilon_a(g)$

II. f **pseudo-matches** g at a if

- $f(a) > g(a)$, or
- $\mathrm{ord}_a(g) + 1 \leq \mathrm{ord}_a(f)$ is odd, $\mathrm{ord}_a(g)$ is even, and $\epsilon_a(g) = -1$

III. f **pair-matches** (g, h) at a if the following hold

- $\max(\text{ord}_a(g), \text{ord}_a(h)) + 1 \leq \text{ord}_a(f)$ is even,
- $\text{ord}_a(g), \text{ord}_a(h)$ are odd, and
- $\epsilon_a(gh) = \epsilon_a(f) = -1$

The concepts of matching, pseudo-matching and pair-matching encode the sign and order conditions from Theorem 2.5. In particular, these are useful in the following context:

I. **matching**: this case detects when f is a local sums of squares at a, or identifies the generators with same order parity and signs with f at a. The latter is applicable for left/right endpoints of intervals of $\mathcal{S}(G)$ or isolated points of type A, B, C or D.

II. **pseudo-matching**: this case detects when f is a strictly positive polynomial over $\mathcal{S}(G)$, particularly at a, or identifies the generators with even order, $\text{ord}_a(f)$ is odd and the sign of the associated generator g at a is negative. The latter is applicable for isolated points of type A, C or D.

III. **pair-matching**: this case identifies the pairs of generators with both odd orders and $\text{ord}_a(f)$ is even, and the sign with f at a is negative. This case is applicable for isolated points of type B.

Example 6.3 Consider $G = \{g_1, g_2\}$ where $g_1 = -(X+2)^2(X+1)^3(X-1)(X-2)(X-3)$ and $g_2 = -(X+2)^5(X-2)^3$.

- Consider $f_1 = (X+2)^3(X+1)^5$.
  - f_1 does not match g_1 or g_2 at -2 because $\text{ord}_a(f_1)$ is odd, $\text{ord}_{-2}(f_1) - \text{ord}_{-2}(g_1) = 1$ and $\epsilon_{-2}(f_1) = -1$ while $\epsilon_{-2}(g_2) = 1$.

  - f_1 pseudo-matches g_1 at -2 because $\text{ord}_{-2}(g_1) = 2$ is even, $\text{ord}_{-2}(f_1) = 3$ is odd, $\text{ord}_{-2}(f_1) > \text{ord}_{-2}(g_1)$ and $\epsilon_{-2}(g_1) = -1$.
  - f_1 matches g_1 at -1 because $\text{ord}_{-1}(f_1) - \text{ord}_{-1}(g_1) = 2$ is even and $\epsilon_{-1}(f_1) = \epsilon_{-1}(g_1) = 1$.

- Consider $f_2 = -(X+2)^4(X+1)^2(X-2)^4$.
  - f_2 matches g_1 at -2 because $\text{ord}_{-2}(f_2) - \text{ord}_{-2}(g_1) = 2$ is even and $\epsilon_{-2}(f_2) = \epsilon_{-2}(g_1) = -1$.
  - f_2 does not match g_1 or g_2 at -1 because $\text{ord}_{-1}(f_2) - \text{ord}_{-1}(g_1) = -1$ is odd and $\epsilon_{-1}(f_2) = -1$ while $\epsilon_{-1}(g_2) = 1$.
  - f_2 does not pseudo-match g_1 at -1 because $f_2(-1) = 0 = g_1(-1)$ and $\text{ord}_{-1}(f_2) = 2$ is even.
  - f_2 does not pseudo-match g_2 at -1 because $f_2(-1) = 0 < g_2(-1) = 27$ and $\text{ord}_{-1}(f_2) = 2$ is even.
  - f_2 does not match g_1 nor g_2 at 2 because $\text{ord}_2(f_2) - \text{ord}_2(g_1) = 3$ is odd and $\text{ord}_2(f_2) - \text{ord}_2(g_2) = 1$ is odd.
  - f_2 pair-matches (g_1, g_2) at 2 because both $\text{ord}_2(g_1) = 1$ and $\text{ord}_2(g_2) = 3$ are odd, $\text{ord}_2(f_2) = 4$ is even, $\text{ord}_2(f_2) = \text{ord}_2(g_2) + 1$ and $\epsilon_2(g_1 g_2) = \epsilon_2(f_2) = -1$.

Previously, we discussed how isolated points of type B are related with the concept of pair-matching, as we need two generators with odd order and opposite sign conditions at a given point. The following theorem relates the real zeros of a member in a quadratic module with the generators with matching or pseudo-matching generators excluding the zeros of type B with respect to G.

Theorem 6.4 *Assume $f \in \mathrm{QM}(G)$. For every real root a of f that is not of type B with respect to G, there exists a generator $g \in G$ such that f matches g at a or f pseudo-matches g at a.*

Proof. Let a be a real zero of f. If $\mathrm{ord}_a(f)$ is even and $\epsilon_a(f) = 1$, then f matches any $g \in G$ at a. So, for the rest of the analysis, we consider that $\mathrm{ord}_a(f)$ is odd, or $\mathrm{ord}_a(f)$ is even and $\epsilon_a(f) = -1$.

We have two cases whether a is an end point of $S := \mathcal{S}(G)$ or not. Consider the case where a is an end point of S. Since $f \in \mathrm{QM}(G)$ we know from [4, Theorem 2.18 p. 47] that f satisfies the following conditions:

1. for every left boundary point a of $S \setminus S_{isol}$, we have $\mathrm{ord}_a(f)$ is even or $\mathrm{ord}_a(f) - \kappa_a^+(G) \in 2\mathbb{N}_0$: if $\mathrm{ord}_a(f)$ is even, from the fact that $f \geq 0$ over S we have $\epsilon_a(f) = 1$, so f matches any $g \in G$. If $\mathrm{ord}_a(f) - \kappa_a^+(G)$, then f matches the generator $g_i \in G$ such that $\mathrm{ord}_a(g_i) = \kappa_a^+$.

2. for every right boundary point a of $S \setminus S_{isol}$, we have $\mathrm{ord}_a(f)$ is even or $\mathrm{ord}_a(f) - \kappa_a^-(G) \in 2\mathbb{N}_0$: this case is handled similarly to the previous case.

3. for every isolated point a of S, we have $\mathrm{ord}_a(f)$ is even and $\epsilon_a(f) = 1$ or

   Case 1: $\mathrm{ord}_a(f) \geq \kappa_a(G)$ if $\kappa_a(G) < \kappa_a^+(G)$ and $\kappa_a(G) < \kappa_a^-(G)$

   Case 2: $(\mathrm{ord}_a(f) - \kappa_a^+(G) \in 2\mathbb{N}_0$ and $\epsilon_a(f) = 1)$ or $\mathrm{ord}_a(f) \geq \min\{\kappa_a(G), \kappa_a^-(G)\}$ if $\kappa_a^+(G) \leq \min\{\kappa_a(G), \kappa_a^-(G)\}$

   Case 3: $(\mathrm{ord}_a(f) - \kappa_a^-(G) \in 2\mathbb{N}_0$ and $\epsilon_a(f) = -1)$ or $\mathrm{ord}_a(f) \geq \min\{\kappa_a(G), \kappa_a^+(G)\}$ if $\kappa_a^-(G) \leq \min\{\kappa_a(G), \kappa_a^+(G)\}$

   If the first disjunct is satisfied, then f matches any $g \in G$ as mentioned before. For Case 1, if $\mathrm{ord}_a(f)$ is even, then f matches some $g_i \in G$ that $\mathrm{ord}_a(g_i) = \kappa_a(G)$; if $\mathrm{ord}_a(f)$ is odd and $\epsilon_a(f) = 1$, then f pseudo-matches some $g_i \in G$ that

$\text{ord}_a(g_i) = \kappa_a^+(G)$; if $\text{ord}_a(f)$ is odd and $\epsilon_a(f) = -1$, then f pseudo-matches some $g_i \in G$ that $\text{ord}_a(g_i) = \kappa_a^-(G)$.

For Case 2, if $\text{ord}_a(f) - \kappa_a^+(G) \in 2\mathbb{N}_0$ and $\epsilon_a(f) = 1$, then f matches some generator $g_i \in G$ such that $\text{ord}_a(g_i) = \kappa_a^+(G)$.

If $\kappa_a(G) = \min\{\kappa_a(G), \kappa_a^-(G)\} \leq \text{ord}_a(f)$, then f pseudo-matches some generator $g_i \in G$ such that $\text{ord}_a(g_i) = \kappa_a(G)$.

If $\kappa_a^-(G) = \min\{\kappa_a(G), \kappa_a^-(G)\} \leq \text{ord}_a(f)$, $\text{ord}_a(f)$ is odd, and $\epsilon_a(f) = -1$, then f matches some generator $g_i \in G$ such that $\text{ord}_a(g_i) = \kappa_a^-(G)$.

If $\kappa_a^-(G) = \min\{\kappa_a(G), \kappa_a^-(G)\} \leq \text{ord}_a(f)$, $\text{ord}_a(f)$ is odd, and $\epsilon_a(f) = 1$, then f matches some generator $g_i \in G$ such that $\text{ord}_a(g_i) = \kappa_a^+(G)$.

We dismiss the case where $\min\{\kappa_a(G), \kappa_a^-(G)\} = \kappa_a^-(G) \leq \text{ord}_a(f)$, $\text{ord}_a(f)$ is even and $\epsilon_a(f) = -1$, since for this case the isolated point a corresponds to a type B.

Case 3 is handled similarly to Case 2 above.

If a is not an end point of $\mathcal{S}(G)$, we have two cases to determine whether a is in the interior of $\mathcal{S}(G)$ or not. If a is in the interior of $\mathcal{S}(G)$, then it must be the case that $\epsilon_a(f) = 1$ as otherwise it contradicts the nonnegativity condition of f over $\mathcal{S}(G)$. For such a case, f matches any $g \in G$ as mentioned before. If a is not in the interior of $\mathcal{S}(G)$ nor an end point, then $a \notin \mathcal{S}(G)$, so there exists a generator $g \in G$ such that $g(a) < 0$. Then, f pseudo-matches g at a. □

The case of an isolated point a of type B when $\text{ord}_a(f)$ is even and $\epsilon_a(f) = -1$ is not considered as for this case, precisely f pair-matches (g_i, g_j) at a where g_i, g_j are the two generators that define a. We denote $(X - \infty) := 1$ and $(X - (-\infty)) := -1$. The following theorem allows us to compute the certificates of a polynomial of the form $(X-c)(X-a)^{\text{ord}_a(f)}(X-d)$ when f pair-matches (g_i, g_j) at a such that c (resp. d) is the half point between a and the closest real root of the generators G to the

left (resp. right) of a; in case there is not a real root of the generators G closest to the left (resp. right) of a, we assign c (resp. d) with ∞ (resp. $-\infty$).

Theorem 6.5 *Let g_i, g_j be generators in G such that f pair-matches (g_i, g_j) at a. The polynomial $(X-c)(X-a)^{\mathrm{ord}_a(f)}(X-d)$ described above belongs to $\mathrm{QM}(G)$ and its certificates are computable.*

Proof. Let us assume that $\mathrm{ord}_a(g_i) \geq \mathrm{ord}_a(g_j)$. We will provide a procedure for computing a certificate of the polynomial $(X-c)(X-a)^{\mathrm{ord}_a(g_i)+1}(X-d)$.

We use the MonogenicCert algorithm to compute a certificate of $(X-c)(X-a)^{\mathrm{ord}_a(g_i)}$ in $\mathrm{QM}(\mathrm{compact}_G(g_i))$ as it is nonnegative over $\mathcal{S}(\mathrm{compact}_G(g_i))$. Similarly, we obtain a certificate of $(X-a)^{\mathrm{ord}_a(g_j)}(X-d)$ in $\mathrm{QM}(\mathrm{compact}_G(g_j))$.

Let us consider the case where $\mathrm{ord}_a(g_i) = 1$. It is enough to find a certificate of $(X-c)(X-a)^2(X-d)$ by finding the certificate of the product $(X-c)(X-a)$ and $(X-a)(X-d)$ in the set of generators $\{X-\alpha, \mathrm{compact}_G(g_i), \mathrm{compact}_G(g_j), -(X-\beta)\}$ where $X-\alpha$ and $-(X-\beta)$ are strictly positive polynomials over $\mathcal{S}(G)$, i.e.,

$$\begin{aligned}(X-c)(X-a)^2(X-d) = s_0 + s_1\cdot(X-\alpha) + s_2\cdot(X-c)(X-a) \\ + s_3\cdot(X-a)(X-d) + s_4\cdot(-(X-\beta))\end{aligned} \tag{6.1}$$

where s_i are sums of squares.

Now, let us consider the case where $\mathrm{ord}_a(g_i) = 2n+1$ for some $n \in \mathbb{N}$. Our goal is to compute a certificate of the polynomial $(X-c)(X-a)^{2n+2}(X-d)$. Multiplying the sums of squares $(X-a)^{2n}$ on both sides of Equation (6.1) we obtain the following.

$$\begin{aligned}
&(X-c)(X-a)^{2n+2}(X-d)\\
&= (X-a)^{2n}s_0 + (X-a)^{2n}s_1\cdot(X-\alpha) + s_2\cdot(X-c)(X-a)^{\mathrm{ord}_a(g_i)}\\
&+ (X-a)^{\mathrm{ord}_a(g_i)-\mathrm{ord}_a(g_j)}s_3\cdot(X-a)^{\mathrm{ord}_a(g_j)}(X-d)\\
&+ (X-a)^{2n}s_4\cdot(-(X-\beta))
\end{aligned} \tag{6.2}$$

Since $\mathrm{ord}_a(g_i) - \mathrm{ord}_a(g_j)$ is even, Equation (6.2) is a certificate of $(X-c)(X-a)^{\mathrm{ord}_a(g_i)+1}(X-d)$ in $\mathrm{QM}(X-\alpha, (X-c)(X-a)^{\mathrm{ord}_a(g_i)}, (X-a)^{\mathrm{ord}_a(g_j)}(X-d), -(X-\beta))$.

We use the Certificate algorithm to compute certificates of $X-\alpha$ and $-(X-\beta)$ in $\mathrm{QM}(G)$ as these are strictly positive over $\mathcal{S}(G)$. As each generator is a certificate itself, the Algorithm 9 for compactification provides certificates of $\mathrm{compact}_G(g_i)$ and $\mathrm{compact}_G(g_i)$ in $\mathrm{QM}(G)$. Finally, we use Algorithm 20 to lift the certificates of $(X-c)(X-a)^{\mathrm{ord}_a(g_i)+1}(X-d)$ from $\mathrm{QM}(X-\alpha, \mathrm{compact}_G(g_i), \mathrm{compact}_G(g_j), -(X-\beta))$ to $\mathrm{QM}(G)$. □

Example 6.6 We discuss a simple example to highlight the construction in the proof of Theorem 6.5. Consider $G = \{g_1, g_2\}$ where $g_1 = X^3$ and $g_2 = -X^5$. The input polynomial $f = -X^6$ is a member of $\mathrm{QM}(g_1, g_2)$.

First, we compute the certificate of $-X^2$ in $\mathrm{QM}(X, -X)$. Using the formula of Section 4.2 we obtain $-X^2 = \frac{1}{4}(X-1)^2\cdot X + \frac{1}{4}(X+1)^2\cdot(-X)$. Multiplying X^4 by the previous equation, we obtain $-X^6 = \frac{1}{4}((X-1)X)^2\cdot g_1 + \frac{1}{4}(X+1)^2\cdot g_2 \in \mathrm{QM}(g_1, g_2)$.

In this section, we have introduced the concepts of matching, pseudo-matching, and pair-matching. The conditions for each of these cases identify a generator or a pair of generators that are required by the certificate of an input polynomial. We prove in Theorem 6.4 and Theorem 6.5 that for a member $f \in \mathrm{QM}(G)$, we can always associate a generator in G with the real root of f.

6.3 Pure and mixed polynomials

From Theorem 6.4, we observe the relation between the concepts of matching and pseudo-matching with the real roots of a member in a quadratic module. The concept of pure polynomials allows us to identify a kind of member in quadratic modules that only requires a single generator to be used in its certificates.

Definition 6.7 Let $G = \{g_1, \ldots, g_s\} \subseteq \mathbb{A}[X]$ and $f \in \mathbb{A}[X]$. Let us define LocalRelativeGens(f, G, a) and RelativeGens(f, G) as subsets of G as

$$\text{LocalRelativeGens}(f, G, a) := \{g \in G \mid f \text{ matches or pseudo-matches } g \text{ at } a\}$$

$$\text{RelativeGens}(f, G) := \bigcap_{a \in \mathcal{Z}(f)} \text{LocalRelativeGens}(f, G, a)$$

We say that f is **pure** with respect to G if RelativeGens(f, G) is not empty. We say that f is **mixed** with respect to G if f is not pure with respect to G or there are $g_i, g_j \in G$ and a point $a \in \mathcal{Z}(f)$ such that f pair-matches (g_i, g_j) at a.

The motivation for the above definitions allows us to distinguish two cases of members in a quadratic module. If $f \in \text{QM}(G)$ is pure and $f \geq 0$ over $\mathcal{S}(\text{compact}_G(g))$ for some $g \in \text{RelativeGens}(f, G)$, then $f \in \text{QM}(\text{compact}_G(g))$.

Theorem 6.8 *Assume that $f \in \text{QM}(G)$ is pure with respect to G and satisfies $f \geq 0$ over $\mathcal{S}(\text{compact}_G(g))$ for some $g \in \text{RelativeGens}(f, G)$. Then $f \in \text{QM}(\text{compact}_G(g))$.*

Proof. We will assume that the semialgebraic set of $g \in \text{RelativeGens}(f, G)$ is bounded, as otherwise we use the compactification algorithm on g. Since $f \geq 0$ over $\mathcal{S}(g)$, we find that f is nonnegative over the end points of $\mathcal{S}(g)$.

Using Theorem 2.15, it is enough to check the conditions at the end points x of $\mathcal{S}(g)$. If $f(x) > 0$, we have that $\text{ord}_x(f) = 0$ is even and $\epsilon_x(f) = 1$, so we satisfy item (1) of Theorem 2.15; hence, we just consider when x is a root of f. We have two cases:

- x is a boundary point of $\mathcal{S}(g)$ that is not isolated: since x is a boundary point of $\mathcal{S}(g)$, we find that $\text{ord}_x(g)$ is odd, so f cannot pseudo-match g at x; hence, f matches g at x, so $\text{ord}_x(f)$ is even and $\epsilon_x(f) = 1$,or $\text{ord}_x(f) - \text{ord}_x(g)$ is even, so we satisfy item (1) from Theorem 2.15.
- x is an isolated point of $\mathcal{S}(g)$: f matches or pseudo-matches g at x, so in any case, we have $\text{ord}_x(f) \geq \text{ord}_x(g)$, which satisfies item (2) from Theorem 2.15.

□

Example 6.9 From Example 6.3, we have that f_1 matches g_1 at -2 and pseudo-matches g_1 at -1; thus, f_1 is pure with respect to G as $\text{RelativeGens}(f_1, G) = \{g_1\}$; in fact, $f_1 \in \text{QM}(g_1)$. As $\mathcal{S}(g_1)$ is bounded, it is enough to use the MonogenicCert algorithm to compute a certificate of f_1. Let $\hat{g_1} = -(X+1)(X-1)(X-2)(X-3)$ and $\hat{f_1} = (X+2)(X+1)$. We have $X + 1 = p_0 + p_1 \cdot \hat{g_1}$ and $X + 2 = q_0 + q_1 \cdot \hat{g_1}$ where:

- $p_0 = \frac{3}{4}((X+1)(-\frac{7}{36}X+1))^2 + \frac{23}{1728}((X+1)X)^2$
- $p_1 = \frac{1}{24}$
- $q_0 = \frac{34591}{20892}(-\frac{10744}{34591}X^2 + X + \frac{21190}{34591})^2 + \frac{130480693}{903343965}(\frac{10424235}{1043845544}X^2 + 1)^2 + \frac{3341007408243}{72693403684160}(X^2)^2$
- $q_1 = \frac{5372}{26115}$

Hence, $\hat{f}_1 = (p_0 + p_1 \cdot \hat{g_1})(q_0 + q_1 \cdot \hat{g_1}) = (p_0q_0 + p_1q_1\hat{g_1}^2) + (p_0q_1 + p_1q_0) \cdot \hat{g_1}$. Multiplying the sums of squares $(X+2)^2(X+1)^4$ by the previous equation, we obtain $f_1 = (X+2)^2(X+1)^4(p_0q_0 + p_1q_1\hat{g_1}^2) + (X+1)^2(p_0q_1 + p_1q_0) \cdot g_1$, which is a certificate of $f_1 \in \mathrm{QM}(g_1)$.

Notice that the conditions of Theorem 6.8 are true, we can reduce the certificate computation to a monogenic certificate problem and obtain certificates using the MonogenicCert algorithm with inputs f and the compactification of the polynomial $g \in \mathrm{RelativeGens}(f, G)$.

6.4 Intermediate generators $H_{f,G}$

In this section, we describe a procedure for computing the elements of the intermediate generators $H_{f,G}$. This set of generators allows us to reconstruct a certificate of f by multiplying the elements of $H_{f,G}$ so that we can obtain a certificate of these products. Then, we use the Basic Lemma (Theorem 2.20) to remove additional factors not in f. We prove the containment of $H_{f,G}$ in $\mathrm{QM}(G)$. The goal of the algorithm is to ensure that $H_{f,G}$ satisfies the following conditions:

1. The strictly positive factor f_{pos} of f is strictly positive over $\mathcal{S}(H_{f,G})$.
2. Let $f_{non-neg} := \frac{f}{f_{pos}}$. $\mathcal{S}(H_{f,G}) \subseteq \mathcal{S}(f_{non-neg}) \cap [\alpha, \beta]$ for some $\alpha < \mathcal{S}(G) < \beta$.

The first condition allows us to simplify the input polynomial, as the method processes the real roots of the input polynomial. The second condition ensures that the semialgebraic set of $H_{f,G}$ is bounded and $f_{non-neg}$ remains nonnegative over $\mathcal{S}(H_{f,G})$. Additionally, we can use the Certificate algorithm to compute a certificate of f_{pos} in $\mathrm{QM}(H_{f,G})$.

We construct $H_{f,G}$ in four steps. First, we construct the set $H^0_{f,G}$ that involves nonnegative factors of $f_{non-neg}$. Then, we set $H^1_{f,G}$ from $H^0_{f,G}$ by adding two strictly positive polynomials that guaranty that $\mathcal{S}(H^1_{f,G})$ is bounded. Next, we define $H^2_{f,G}$ from $H^1_{f,G}$ by adding polynomials of the form $(X-\alpha_i)(X-\beta_i)$ which remove the intervals from $H^1_{f,G}$ that make $f_{non-neg}$ evaluate to a negative number. Finally, $H_{f,G}$ is constructed from $H^2_{f,G}$ by adding polynomials of the form $(X-\gamma_i)(X-\delta_i)$ that remove the intervals of $H^2_{f,G}$ that make f_{pos} a negative number.

First, we compute the strictly positive factor f_{pos} of f using the StrictPos algorithm with inputs f and $\mathcal{S}(G)$ and set $f_{non-neg} := \frac{f}{f_{pos}}$. Let a be a real root of $f_{non-neg}$. From Theorem 6.4, we know that LocalRelativeGens$(f_{non-neg}, G, a)$ is not empty or there exists a pair of generators $g_i, g_j \in G$ such that $f_{non-neg}$ pair-matches (g_i, g_j). For every root a of $f_{non-neg}$ such that there exist g_i, g_j that $f_{non-neg}$ pair-matches (g_i, g_j), we add the polynomial $(X-c_1)(X-a)^{\mathrm{ord}_a(f)}(X-c_2)$ where c_1 (resp. c_2) is the mid point between a and $\max\{z \in \mathcal{Z}(f_{non-neg}) \mid z < a\}$ (resp. $\min\{z \in \mathcal{Z}(f_{non-neg}) \mid z > a\}$); Theorem 6.5 will allow us to compute a certificate in QM(G) of each polynomial $(X-c_1)(X-a)^{\mathrm{ord}_a(f)}(X-c_2)$ added to $H^0_{f,G}$ in this step.

For every real root a of $f_{non-neg}$ such that LocalRelativeGens$(f_{non-neg}, G, a)$ is not empty, we add to $H^0_{f,G}$ the following polynomials:

- $a = a_0$:
  - Case $\epsilon_a(f_{non-neg}) = 1$ and $\mathrm{ord}_a(f_{non-neg})$ is odd: We add to $H^0_{f,G}$ the polynomial $(X-a)^{\mathrm{ord}_a(f)}$.
  - Case $\epsilon_a(f_{non-neg}) = -1$ and $\mathrm{ord}_a(f_{non-neg})$ is even: We add to $H^0_{f,G}$ the polynomial $(X-a)^{\mathrm{ord}_a(f)}(X-b)$ where b is the midpoint between a and $\min\{z \in \mathcal{Z}(f_{non-neg}) \mid z > a\}$.
- $a = b_k$:

  - Case $\epsilon_a(f_{non-neg}) = -1$ and $\text{ord}_a(f_{non-neg})$ is odd: We add to $H^0_{f,G}$ the polynomial $-(X-a)^{\text{ord}_a(f)}$.
  - Case $\epsilon_a(f_{non-neg}) = -1$ and $\text{ord}_a(f_{non-neg})$ is even: We add to $H^0_{f,G}$ the polynomial $-(X-b)(X-a)^{\text{ord}_a(f)}(X-b)$ where b is the midpoint between $\max\{z \in \mathcal{Z}(f_{non-neg}) \mid z < a\}$ and a.

- $a_0 < a < b_k$:
  - Case $\epsilon_a(f_{non-neg}) = -1$ and $\text{ord}_a(f_{non-neg})$ is odd: We add to $H^0_{f,G}$ the polynomial $(X-a)^{\text{ord}_a(f)}(X-b)$ where b is the midpoint between a and $\min\{z \in \mathcal{Z}(f_{non-neg}) \mid z > a\}$.
  - Case $\epsilon_a(f_{non-neg}) = 1$ and $\text{ord}_a(f_{non-neg})$ is odd: We add to $H^0_{f,G}$ the polynomial $(X-b)(X-a)^{\text{ord}_a(f)}$ where b is the midpoint between $\max\{z \in \mathcal{Z}(f_{non-neg}) \mid z < a\}$ and a.
  - Case $\epsilon_a(f_{non-neg}) = -1$ and $\text{ord}_a(f_{non-neg})$ is even: We add to $H^0_{f,G}$ the polynomial $(X-c_1)(X-a)^{\text{ord}_a(f)}(X-c_2)$ where c_1 (resp. c_2) is the midpoint between a and $\max\{z \in \mathcal{Z}(f_{non-neg}) \mid z < a\}$ (resp. $\min\{z \in \mathcal{Z}(f_{non-neg}) \mid z > a\}$).

Now, we set $H^1_{f,G} = H^0_{f,G} \cup \{X-\alpha, -(X-\beta)\}$ so that $X-\alpha, -(X-\beta)$ are strictly positive over $\mathcal{S}(G)$.

Now, we compute $H^2_{f,G}$. First, we compute the semialgebraic of $H^1_{f,G}$. For each interval $[c_i, d_i] \subseteq \mathcal{S}(H^1_{f,G})$ such that $\{x \in [c_i, d_i] \mid f_{non-neg}(x) < 0\} \neq \emptyset$ we add the polynomial $(X-\alpha_i)(X-\beta_i)$ such that α_i (resp. β_i) is the midpoint in-between c_i (resp. d_i) and the closest root of a generator $h \in H^1_{f,G}$ to the left (resp.) of c_i (resp. d_i).

Finally, we compute $H_{f,G}$. First, we compute the semialgebraic of $H^2_{f,G}$. For each interval $[c_i, d_i] \subseteq \mathcal{S}(H^2_{f,G})$ such that $\{x \in [c_i, d_i] \mid f_{pos}(x) < 0\} \neq \emptyset$ we add the

polynomial $(X-\gamma_i)(X-\delta_i)$ such that γ_i (resp. δ_i) is the midpoint in-between c_i (resp. d_i) and the closest root of a generator $h \in H^2_{f,G}$ to the left (resp.) of c_i (resp. d_i).

Proposition 6.10 *Let $H^0_{f,G}$ and $H_{f,G}$ be defined as above. We have $\mathcal{S}(f_{non-neg}) \subseteq \mathcal{S}(H^0_{f,G})$. Furthermore, let α, β be such that $\alpha < \mathcal{S}(G) < \beta$. We have $\mathcal{S}(H_{f,G}) \subseteq \mathcal{S}(f_{non-neg}) \cap [\alpha, \beta]$.*

Proof. Assume by contradiction that there exists $x \in \mathcal{S}(f_{non-neg})$ such that $x \notin \mathcal{S}(H^0_{f,G})$. The latter implies that there is an $h \in H^0_{f,G}$ such that $h(x) < 0$. We consider all possible cases of a polynomial h in $H^0_{f,G}$:

- Pair-matching (g_i, g_j) at a: For this case, h is of the form $(X-c_1)(X-a)^{\mathrm{ord}_a(f)}(X-c_2)$ with $\mathrm{ord}_a(f)$ even. If $h(x) < 0$ then $(x-c_1)(x-a)^{\mathrm{ord}_a(f)}(x-c_2) < 0$, so h is negative over $(c_1, a) \cup (a, c_2)$. On the other hand, by continuity, $f_{non-neg}$ is negative over $(r_1, a) \cup (a, r_2) \supseteq (c_1, a) \cup (a, c_2)$, where r_1 (resp. r_2) is the closest root of $f_{non-neg}$ to the left (resp. right) of a, so $x \notin \mathcal{S}(f_{non-neg})$.
- Matching g_i at a:
  - Case $a = a_0$:
    - Case $\epsilon_a(f_{non-neg}) = 1$ and $\mathrm{ord}_a(f_{non-neg})$ is odd. In this case, the polynomial h is $(X-a)^{\mathrm{ord}_a(f)}$. Hence, h is negative over $(-\infty, a)$. As $f_{non-neg}$ does not have a strictly positive factor, it is also negative over $(-\infty, a)$, so $x \notin \mathcal{S}(f_{non-neg})$.
    - Case $\epsilon_a(f_{non-neg}) = -1$ and $\mathrm{ord}_a(f_{non-neg})$ is even. In this case, the polynomial h is $(X-a)^{\mathrm{ord}_a(f)}(X-b)$. If $h(x) < 0$ then $(x-a)^{\mathrm{ord}_a(f)}(x-b) < 0$, so h is negative over (a, b). On the other hand, by continuity, $f_{non-neg}$ is negative over $(a, r) \supseteq (a, b)$, where r is the closest root of $f_{non-neg}$ to the right of a, so $x \notin \mathcal{S}(f_{non-neg})$.

- Case $a = b_k$: This case is handled similarly to the previous case.
- Case $a_0 < a < b_k$:
  - Case $\epsilon_a(f_{non-neg}) = -1$ and $\text{ord}_a(f_{non-neg})$ is odd. This case is handled similarly to the previous case.
  - Case $\epsilon_a(f_{non-neg}) = 1$ and $\text{ord}_a(f_{non-neg})$ is odd. This case is handled similarly to the previous case.
  - Case $\epsilon_a(f_{non-neg}) = -1$ and $\text{ord}_a(f_{non-neg})$ is even. This case is handled similarly to the pair-matching case.

Now, we prove $\mathcal{S}(H_{f,G}) \subseteq \mathcal{S}(f_{non-neg}) \cap [\alpha, \beta]$. Suppose, by contradiction, $x \in \mathcal{S}(H_{f,G})$ but $x \notin \mathcal{S}(f_{non-neg}) \cap [\alpha, \beta]$. Since $H_{f,G}$ contains the polynomials $X - \alpha, -(X - \beta)$, we have $x \in [\alpha, \beta]$, thus the latter implies $x \notin \mathcal{S}(f_{non-neg})$, so $f_{non-neg}(x) < 0$. Since $x \in \mathcal{S}(H_{f,G})$, there must exist an interval $[c, d] \subseteq \mathcal{S}(H_{f,G})$ such that $x \in [c, d]$. However, this is not possible as the third step in the construction of $H_{f,G}$ adds polynomials of the form $(X - \alpha_i)(X - \beta_i)$ to avoid intervals for which $f_{non-neg}$ evaluates to a negative number. □

Proposition 6.11 *The strictly positive factor* f_{pos} *of* f *is strictly positive over* $\mathcal{S}(H_{f,G})$.

Proof. The proof is similar to the proof of Proposition 6.10. Suppose, by contradiction, $x \in \mathcal{S}(H_{f,G})$ but $x \notin \mathcal{S}(f_{pos})$. The latter implies $f_{pos}(x) < 0$. Since $x \in \mathcal{S}(H_{f,G})$, there must exist an interval $[c, d] \subseteq \mathcal{S}(H_{f,G})$ such that $x \in [c, d]$. However, this is not possible as the fourth step in the construction of $H_{f,G}$ adds polynomials of the form $(X - \gamma_i)(X - \delta_i)$ to avoid intervals for which f_{pos} evaluates to a negative number.

□

Proposition 6.12 *$H_{f,G} \subseteq \mathrm{QM}(G)$. Additionally, a certificates for each $h \in H_{f,G}$ in $\mathrm{QM}(G)$ are computable.*

Proof. The polynomials $X-\alpha$, $-(X-\beta)$, $(X-\alpha_i)(X-\beta_i)$, and $(X-\gamma_i)(X-\delta_i)$ are strictly positive over $\mathrm{QM}(G)$; hence, we can use the Certificate algorithm to compute their certificates in $\mathrm{QM}(G)$.

Consider a nonnegative factor of $f_{non-neg}$ added to $H_{f,G}$. These contain a root a such that there exists a generator $g \in G$ such that $f_{non-neg}$ matches or pseudo-matches g at a or there exists a pair of generators $g_i, g_j \in G$ such that $f_{non-neg}$ pair-matches (g_i, g_j).

In the former case, we use Theorem 6.8 to obtain a certificate of the nonnegative factor in $\mathrm{QM}(\mathrm{compact}_G(g))$. For the second case, we use Theorem 6.5 to obtain a certificate of the nonnegative factor. □

Example 6.13 In this example, we show the construction of elements of the intermediate set of generators $H_{f,G}$ for $G = \{g_1, g_2\}$ where $g_1 = -(X+2)(X+1)^3(X-1)(X-2)$, $g_2 = (X+1)(X-1)^3$, and the input polynomial f being considered is $(X+1)(X-1)$. We also compute a certificates in $\mathrm{QM}(G)$ of the elements in $H_{f,G}$.

1. First, we compute $f_{pos} = 1$, so $f_{non-neg} := f$.
2. We add the following polynomials to $H_{f,G}$ according to the generators matching the roots of $f_{non-neg}$:
   - f matches g_2 at $x = -1$: We add to $H_{f,G}$ the polynomial $(X+1)X$.
   - f matches g_1 at $x = 1$: We add to $H_{f,G}$ the polynomial $X(X-1)$.
3. To make $\mathcal{S}(H_{f,G})$ bounded, we add the polynomials $X+3, -(X-3)$.

4. Since $0 \in \mathcal{S}(H_{f,G})$ and $0 \notin \mathcal{S}(f)$, we then add the polynomial $(X+\frac{1}{2})(X-\frac{1}{2})$ to $H_{f,G}$, so $H_{f,G} = \{X+3, (X+1)X, (X+\frac{1}{2})(X-\frac{1}{2}), X(X-1), -(X-3)\}$.

Now, we compute a certificate in $\mathrm{QM}(G)$ of the elements in $H_{f,G}$.

- $X+3$: This polynomial is strictly positive over $\mathcal{S}(G)$ and was added to $H_{f,G}$ in the last step to ensure that $\mathcal{S}(H_{f,G})$ is bounded to the left. We use Certificate to compute a certificate of it. We obtain $X+3 = s_0 + s_1 \cdot g_1 + s_2 \cdot g_2$ where

  - $s_0 = \frac{173501}{42499}(-\frac{792583}{5552032}X^3 - \frac{470293}{1388008}X^2 + \frac{134507}{347002}X + 1)^2 + \frac{13880608185}{7373618999}(-\frac{1343962716491}{4441794619200}X^3 - \frac{11048051195}{88835892384}X^2 + X)^2 + \frac{911010959414697919}{60406985446841856000}(X^3 + \frac{139905551232275750}{911010959414697919}X^2)^2 + \frac{5131472975492026843391 17}{4955783009813151598026 3680}(X^2)^2$
  - $s_1 = \frac{11501}{42499}$
  - $s_2 = 0$

- $(X+1)X$: This polynomial matches g_2. We compactify the generator g_2 to reduce the problem to a monogenic certificate problem. We obtain $\mathrm{compact}_{\{g_1,g_2\}}(g_2) = -(X+\frac{33}{16})(X+1)(X-1)^3(X-\alpha)$, where α is a radical number $\alpha = \frac{1}{48}(65 - \frac{65*34^{\frac{2}{3}}}{(14477+147\sqrt{10131})^{\frac{1}{3}}} + (34(14477+147\sqrt{10131}))^{\frac{1}{3}})$. The certificate of $\mathrm{compact}_{\{g_1,g_2\}}(g_2)$ in $\mathrm{QM}(g_1,g_2)$ is calculated from the output $\{\sigma,\tau,h\}$ of the Algorithm 8 as σ,τ are the sums of squares, $1 = \sigma g_2 + \tau h$ and $\mathrm{compact}_{\{g_1,g_2\}}(g_2) = g_2 h$. We have

  - $\sigma = \frac{65536}{2000033}$
  - $\tau = \frac{65536}{2000033}((X - \frac{65-16\alpha}{32})^2 + (\frac{1}{32}\sqrt{4355 - 130(16\alpha) + 3(16\alpha)^2})^2)$
  - $h = -(X+\frac{33}{16})(X-\alpha)$

  The polynomial h is strictly positive over $\mathcal{S}(g_1,g_2)$, and we use the Certificate algorithm to obtain a certificate of it. We have $h = s_3 + s_4 \cdot g_1 + s_5 \cdot g_2$ where

- $s_3 = h - \frac{3}{10}g_1$
- $s_4 = \frac{3}{10}$
- $s_5 = 0$

Hence, a certificate of $\text{compact}_{\{g_1,g_2\}}(g_2) \in \text{QM}(g_1, g_2)$ is

$$\begin{aligned}\text{compact}_{\{g_1,g_2\}}(g_2) &= \sigma g_2^2 \cdot h + \tau h^2 \cdot g_2 \\ &= \sigma g_2^2 \cdot (s_3 + s_4 \cdot g_1) + \tau h^2 \cdot g_2 \\ &= \sigma g_2^2 s_3 + \sigma g_2^2 s_4 \cdot g_1 + \tau h^2 \cdot g_2\end{aligned}$$

We have $(X+1)X \in \text{QM}(\text{compact}_{\{g_1,g_2\}}(g_2))$. For the latter, we use the certificate obtained in Example 5.16. Lifting $\text{compact}_{\{g_1,g_2\}}(g_2) \in \text{QM}(g_1, g_2)$ from the previous certificate gives us a certificate of $(X+1)X$ in $\text{QM}(g_1, g_2)$.

- $(X + \frac{1}{2})(X - \frac{1}{2})$: This polynomial is strictly positive on $\mathcal{S}(G)$ and was added to $H_{f,G}$ to avoid the isolated point 0 that is not in the semialgebraic set of the input polynomial f. We use Certificate to compute a certificate of it. We obtain $(X + \frac{1}{2})(X - \frac{1}{2}) = s_6 + s_7 \cdot g_1 + s_8 \cdot g_2$ where

  - $s_6 = (-\frac{1}{2} + X)(\frac{1}{2} + X) - \frac{1679616}{6634204312890625}(-2 - X)(-2 + X)(-1 + X)(1 + X)^3(-\frac{31}{2} + (-2 - X)(-2 + X)(-1 + X)(1 + X)^3)^8$
  - $s_7 = \frac{1679616}{6634204312890625}(-\frac{31}{2} + (-2 - X)(-2 + X)(-1 + X)(1 + X)^3)^8$
  - $s_8 = 0$

- $X(X-1)$: This polynomial matches g_1. Since $X(X-1)$ is nonnegative over $\mathcal{S}(g_1)$, we use the MonogenicCert algorithm to compute a certificate of $X(X-1)$ in $\text{QM}(g_1)$. For the latter, we use the certificate obtained in Example 5.14.

- $-(X-3)$: This polynomial is strictly positive on $\mathcal{S}(G)$ and was added to $H_{f,G}$ in the last step to ensure that $\mathcal{S}(H_{f,G})$ is bounded to the right. We use Certificate to compute a certificate of it. We obtain $-(X-3) = s_9 + s_{10} \cdot g_1 + s_{11} \cdot g_2$ where

- $s_9 = \frac{516341}{156215}(-\frac{82265}{4130728}X^3 - \frac{9910683}{41307280}X^2 - \frac{8689}{147526}X + 1)^2$
  $+ \frac{690574799179}{460915481800}(-\frac{144499855068}{690574799179}X^3 - \frac{668391670287}{2762299196716}X^2 + X)^2$
  $+ \frac{173716254385435964813}{4832940772967887328000}(-\frac{74048954978876122510}{173716254385435964813}X^3 + X^2)^2$
  $+ \frac{12542962326617044398511841}{434193354861134067892204720 0}(X^3)^2$
- $s_{10} = \frac{11924}{156215}$
- $s_{11} = 0$

6.5 Certificates computation with QMCert

In this section, we introduce the QMCert algorithm to compute a of an input polynomial f in a given quadratic module $\mathrm{QM}(G)$ generated by a set of polynomials G. The approach identifies whether the input polynomial is pure with respect to the set of generators G. In such a case, we use the MonogenicCert algorithm to compute a certificate of f; otherwise, we factor out a strictly positive polynomial f_{pos} from the input polynomial and construct the intermediate set of generators $H_{f_{non-neg},G}$ from the remaining factors of the input polynomial f. As discussed in the previous section, the intermediate factors $H_{f,G}$ have the property of we can find certificates for the products of its members; using this property we obtain a certificate of the nonnegative input polynomial by multiplying elements in $H_{f_{non-neg},G}$. Finally, such product of elements in $H_{f_{non-neg},G}$ might include additional factors added not in $f_{non-neg}$; to remove these we use the Basic Lemma.

By construction of the intermediate generators $H_{f,G}$, we have $\prod_{\substack{h\in H_{f,G}\\ \mathcal{Z}(h)\cap\mathcal{Z}(f)\neq\emptyset}} h = f_{non-neg}p$ where p is a polynomial. Since $f_{non-neg}$ and p are coprime and nonnegative over $\mathcal{S}(H_{f,G})$, we can find σ, τ strictly positive over $\mathcal{S}(H_{f,G})$ using $\mathsf{BasicLemma}(f_{non-neg}, p, H_{f,G})$ such that $1 = \sigma f_{non-neg} + \tau p$. Multiplying $f = f_{pos}f_{non-neg}$ by both sides of the previous equation, we obtain $f = f_{non-neg}^2\sigma f_{pos} + \tau f_{pos}\prod_{\substack{h\in H_{f,G}\\ \mathcal{Z}(h)\cap\mathcal{Z}(f)\neq\emptyset}} h$. Since both $\sigma f_{pos}, \tau f_{pos}$ are strictly positive over $\mathcal{S}(H_{f,G})$, we can use the Certificate algorithm to compute their certificates in $\mathrm{QM}(H_{f,G})$.

By distributing the products in $\tau f_{pos} \prod_{\substack{h\in H_{f,G}\\ \mathcal{Z}(h)\cap\mathcal{Z}(f)\neq\emptyset}} h$, a certificate of $\tau f_{pos} \prod_{\substack{h\in H_{f,G}\\ \mathcal{Z}(h)\cap\mathcal{Z}(f)\neq\emptyset}} h$ is obtained in $\mathrm{QM}(H_{f,G})$ after lifting the products of elements in $H_{f,G}$ that might appear using the method in Appendix B. Finally, by Proposition 6.12, we obtain a certificate of f in $\mathrm{QM}(G)$. In Algorithm 18, we formalize these ideas.

Algorithm 18: Algorithm to compute certificates in general case

$\mathsf{QMCert}(f, G)$

Input: $f \in \mathbb{A}[X]$, $G = \{g_1, \ldots, g_s\} \subseteq \mathbb{A}[X]$

Output: $\sigma_0, \sigma_1, \ldots, \sigma_s \in \mathbb{A}[X]$

Requires: $\mathrm{QM}(G)$ is Archimedean, $f \in \mathrm{QM}(G)$

Ensures: $\sigma_i \in \sum \mathbb{A}[X]^2$, $f = \sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i$

$\nu := \mathsf{Reduced}_{\mathrm{input}}(f, G)$, $f_1 := \frac{f}{\nu}$

$gens :=$ RelativeGens(f_1, G)

if f_1 *is pure with respect to* G **then**

  Take $g \in gens$ with smallest degree

  Compute certificate of f_1 in $\mathrm{QM}(\mathrm{compact}_G(g))$ using the $\mathsf{MonogenicCert}$ algorithm, i.e., $f_1 = s_0 + s_1 \cdot \mathrm{compact}_G(g)$

  Lift certificate of f_1 from $\mathrm{QM}(\mathrm{compact}_G(g))$ to $\mathrm{QM}(G)$ using the certificate of $\mathrm{compact}_G(g)$ in $\mathrm{QM}(G)$, i.e., $f_1 = \sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i$

else

  $f_{pos} := \mathsf{StrictPos}(f_1, \mathcal{S}(G))$

  $f_{non-neg} := \frac{f_1}{f_{pos}}$

  Construct $H_{f_1,G}$ from Section 6.4 using $f_{pos}, f_{non-neg}, G$

  $H := \prod_{\substack{h\in H_{f_1,G}\\ \mathcal{Z}(h)\cap\mathcal{Z}(f_1)\neq\emptyset}} h$

  Find σ, τ using $\mathsf{BasicLemma}(f_{non-neg}, \frac{H}{f_{non-neg}}, H_{f_1,G})$ such that $1 = \sigma f_{non-neg} + \tau \frac{H}{f_{non-neg}}$

  Compute a certificate of σf_{pos} in $\mathrm{QM}(H_{f_1,G})$ using $\mathsf{Certificate}(H_{f_1,G}, \sigma f_{pos})$, i.e., $\sigma f_{pos} = \tau_0 + \sum_{h\in H_{f_1,G}} \tau_h \cdot h$

  Compute a certificate of τf_{pos} in $\mathrm{QM}(H_{f_1,G})$ using $\mathsf{Certificate}(H_{f_1,G}, \tau f_{pos})$, i.e., $\tau f_{pos} = \tau_0^{'} + \sum_{h\in H_{f_1,G}} \tau_h^{'} \cdot h$

  Compute a certificate of $(\tau_0^{'} + \sum_{h\in H_{f_1,G}} \tau_h^{'} \cdot h)H$ using the product property of $H_{f_1,G}$ in Appendix B, i.e., $(\tau_0^{'} + \sum_{h\in H_{f_1,G}} \tau_h^{'} \cdot h)H = \tau_0^{''} + \sum_{h\in H_{f_1,G}} \tau_h^{''} \cdot h$

  Combine the certificates above to obtain a certificate of f_1 in $\mathrm{QM}(H_{f_1,G})$, i.e., $f_1 = (f_{non-neg}^2 \tau_0 + \tau_0^{''}) + \sum_{h\in H_{f_1,G}} (f_{non-neg}^2 \tau_h + \tau_h^{''}) \cdot h$

  Lift certificate of f_1 from $\mathrm{QM}(H_{f_1}, G)$ to $\mathrm{QM}(G)$ using the certificates of the members in $H_{f_1,G}$ in $\mathrm{QM}(G)$, i.e., $f_1 = \sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i$

return $\nu\sigma_0, \ldots, \nu\sigma_s$

6.6 Examples

Example 6.14 This example is a continuation of Example 6.13. We use the QMCert algorithm compute the certificate of $f = (X+1)(X-1)$ with the set of generators $G = \{g_1, g_2\}$ where $g_1 = -(X+2)(X+1)^3(X-1)(X-2)$ and $g_2 = (X+1)(X-1)^3$. Previously, we obtained $H_{f,G} = \{X+3, (X+1)X, (X+\frac{1}{2})(X-\frac{1}{2}), X(X-1), -(X-3)\}$ and their certificates in $\mathrm{QM}(G)$.

The certificate of the product $(X+1)X, X(X-1)$ in $\mathrm{QM}(H_{f,G})$ is $(X+1)X^2(X-1) = \frac{1}{2}(X-1)^2 \cdot (X+1)X + \frac{1}{2}(X+1)^2 \cdot X(X-1)$. Now, we apply the BasicLemma algorithm with inputs f, X^2 and $H_{f,G}$ to obtain σ, τ strictly positive over $\mathcal{S}(H_{f,G})$ such that $1 = \sigma f + \tau X^2$. We obtain:

- $\sigma = \frac{1}{360}(\frac{-1440-19X^2+19X^4}{1440})^{160}(-\frac{1}{4}+X^2)(-(-1440-19X^2+19X^4))$
- $\tau = 1-4(-1+X)(1+X)+\frac{19}{1440}(-1+4X^2)\sum_{i=0}^{160}(1-\frac{19(-1+X)X^2(1+X)}{1440})^i((X+1)(X-1))^2$

We check τ is a nonnegative polynomial over $\mathbb{R}$ so it can be decomposed into a sums of squares. On the other hand, σ is not nonnegative over $\mathbb{R}$ as $\mathcal{S}(\sigma) \approx [-3.03646, -\frac{1}{2}] \cup [\frac{1}{2}, 3.03646]$ but only strictly positive over $\mathcal{S}(H_{f,G})$; hence, we compute a certificate of σ in $\mathrm{QM}(H_{f,G})$ using the Certificate algorithm. We obtain $\sigma = s_0 + s_1 \cdot (X+3) + s_2 \cdot (X+1)X + s_3 \cdot (X+\frac{1}{2})(X-\frac{1}{2}) + s_4 \cdot X(X-1) + s_5 \cdot (-(X-3))$ where

- $s_0 = \frac{4}{1440}(\frac{-1440-19X^2+19X^4}{1440})^{160}(\frac{130387}{8442}((-\frac{1}{2}-X)^2(\frac{1}{2}-X)^2(3-X)^2(-\frac{1279974}{2567725}-\frac{426658X}{2567725})^{140}) + \frac{130387}{8442}((-\frac{1279974}{2567725}+\frac{426658X}{2567725})^{140}(-\frac{1}{2}+X)^2(\frac{1}{2}+X)^2(3+X)^2))$
- $s_1 = \frac{4}{1440}(\frac{-1440-19X^2+19X^4}{1440})^{160}(\frac{130387}{8442}(-\frac{1}{2}-X)^2(\frac{1}{2}-X)^2(3-X)^2(-\frac{1279974}{2567725}-\frac{426658X}{2567725})^{140}(\frac{71(1-\frac{12X}{71})^2}{210}+\frac{2X^2}{213}) + \frac{130387}{8442}(-\frac{1279974}{2567725}+\frac{426658X}{2567725})^{140}(-3+X)^2(-\frac{1}{2}+X)^2(\frac{1}{2}+X)^2(\frac{71(1+\frac{12X}{71})^2}{210}+\frac{2X^2}{213}))$

- $s_2 = 0$
- $s_3 = \frac{4}{1440}(\frac{-1440-19X^2+19X^4}{1440})^{160}(1440 - (\frac{651935}{804}(3-X)(-\frac{1279974}{2567725} - \frac{426658X}{2567725})^{140}) + 19X^2 - 19X^4 - (\frac{651935}{804}(-\frac{1279974}{2567725} + \frac{426658X}{2567725})^{140}(3+X)) + (\frac{130387}{1407}(3-X)^2(-\frac{1279974}{2567725} - \frac{426658X}{2567725})^{140}(3+X)^2(\frac{71}{210}(1-\frac{12X}{71})^2 + \frac{2X^2}{213})) + (\frac{130387}{1407}(3-X)^2(-\frac{1279974}{2567725} + \frac{426658X}{2567725})^{140}(3+X)^2(\frac{71}{210}(1+\frac{12X}{71})^2 + \frac{2X^2}{213})))$
- $s_4 = 0$
- $s_5 = \frac{4}{1440}(\frac{-1440-19X^2+19X^4}{1440})^{160}((\frac{130387}{8442}(-3-X)^2(-\frac{1}{2}-X)^2(\frac{1}{2}-X)^2(-\frac{1279974}{2567725} - \frac{426658X}{2567725})^{140}((\frac{71}{210}(1-\frac{12X}{71})^2) + \frac{2X^2}{213})) + (\frac{130387}{8442}(-\frac{1279974}{2567725} + \frac{426658X}{2567725})^{140}(-\frac{1}{2}+X)^2(\frac{1}{2}+X)^2(3+X)^2((\frac{71}{210}(1+\frac{12X}{71})^2) + \frac{2X^2}{213})))$

We obtain a certificate of $f \in \mathrm{QM}(H_{f,G})$ by multiplying f to the equation $1 = \sigma f + \tau X^2$. We obtain

$$\begin{aligned}
f &= f^2\sigma + \tau X^2 f \\
&= f^2(s_0 + s_1 \cdot (X+3) + s_3 \cdot (X+\frac{1}{2})(X-\frac{1}{2}) + s_5 \cdot (-(X-3))) \\
&\quad + \tau(\frac{1}{2}(X-1)^2 \cdot (X+1)X + \frac{1}{2}(X+1)^2 \cdot X(X-1)) \\
&= f^2 s_0 + f^2 s_1 \cdot (X+3) + \frac{1}{2}\tau(X-1)^2 \cdot (X+1)X \\
&+ f^2 s_3 \cdot (X+\frac{1}{2})(X-\frac{1}{2}) + \frac{1}{2}\tau(X+1)^2 \cdot X(X-1) + f^2 s_5 \cdot (-(X-3))
\end{aligned}$$

Finally, we obtain a certificate of $f \in \mathrm{QM}(G)$ by lifting from $\mathrm{QM}(H_{f,G})$ to $\mathrm{QM}(G)$; for the latter, we use the certificates computed in Example 6.13 of the elements in $H_{f,G}$ in $\mathrm{QM}(G)$.

Example 6.15 In this example, we compare the results of the proposed method and Shang et. al's general method. Consider the set of generators $G = \{g_1, g_2\}$ where $g_1 = -(X+2)X^3$ and $g_2 = -X^3(X-2)$. Let the input polynomial be $(X+1)X^4(X-1)$. We reduce the problem to computing a certificate of $(X+$

$1)X^2(X-1)$ in $\mathrm{QM}(-(X+2)X, -X(X-2))$. For convenience, we reuse the same polynomials for the above problem, i.e., let $g_1 = -(X+2)X$ and $g_2 = -X(X-2)$ and $f = (X+1)X^2(X-1)$

First, we use our proposed approach. In this case, it is enough to use Theorem 6.5. We compute a certificate of $(X+1)X$ in $\mathrm{QM}(g_2)$. First, we compute the certificate of $X+1$ in $\mathrm{QM}(g_2)$ using the Certificate algorithm as $X+1$ is strictly positive over $\mathcal{S}(g_2)$. We obtain the certificate $X+1 = s_0 + s_1 \cdot g_2$ where

- $s_0 = \frac{35152}{20931}(X - \frac{49373}{70304})^2 + \frac{505372919}{2943066048}$
- $s_1 = \frac{35152}{20931}$

Now, we compute a certificate of X in $\mathrm{QM}(g_2)$ using SOSBasicLemma with inputs X and $-(X-2)$. We obtain $X = s_2 + s_3 \cdot g_2$ where

- $s_2 = \frac{1}{2}X^2$
- $s_3 = \frac{1}{2}$

Thus, multiplying and rearranging the certificates of X and $-(X-2)$, we obtain $(X+1)X = s_4 + s_5 \cdot g_2$ where

- $s_4 = s_0 s_2 + s_3 s_1 g_2^2$
- $s_5 = s_2 s_1 + s_0 s_3$

Similarly, we compute a certificate of $X(X-1)$ in $\mathrm{QM}(g_1)$. We obtain $X(X-1) = s_{10} + s_{11} \cdot g_1$ where:

- $s_6 = \frac{35152}{20931}(X + \frac{49373}{70304})^2 + \frac{505372919}{2943066048}$

- $s_7 = \frac{35152}{20931}$
- $-(X-1) = s_6 + s_7 \cdot g_1$
- $s_8 = \frac{1}{2}X^2$
- $s_9 = \frac{1}{2}$
- $-X = s_8 + s_9 \cdot g_1$
- $s_{10} = s_6 s_8 + s_9 s_7 g_1^2$
- $s_{11} = s_8 s_7 + s_6 s_9$

Then, we apply the procedure in Appendix B.3 to compute a certificate of the product of $(X+1)X$ and $X(X-1)$, which is our reduced problem. We have $(X+1)X^2(X-1) = \frac{1}{2}(X-1)^2 \cdot (X+1)X + \frac{1}{2}(X+1)^2 \cdot X(X-1)$.

Putting all the above together, we have $(X+1)X^4(X-1) = s_{12} + s_{13} \cdot g_1 + s_{14} g_2$ where $g_1 = -(X+2)X^3$, $g_2 = -X^3(X-2)$ are the original generators, and

- $s_{12} = X^2(\frac{1}{2}(X-1)^2 s_4 + \frac{1}{2}(X+1)^2 s_{10})$
- $s_{13} = \frac{1}{2}(X+1)^2 s_{11}$
- $s_{14} = \frac{1}{2}(X-1)^2 s_5$

The degrees of the certificate obtained are $\deg(s_{12}) = 8$, $\deg(s_{13}) = 4$, $\deg(s_{14}) = 4$.

Now, let us use Shang et. al's general method. We fix the input polynomial f at $x = 0$ with $f_1 := f - (4(X+\frac{1}{2})^2 \cdot g_1 + 4(X-\frac{1}{2})^2 \cdot g_2) = 17X^4 + 9X^6$. Since f_1 is a sums of squares, we are done. Hence, the certificate obtained is $f = f_1 + 4(X+\frac{1}{2})^2 \cdot g_1 + 4(X-\frac{1}{2})^2 \cdot g_2$.

Finally, we use the Extended Gram matrix method to check the quality of the certificate obtained with this method. Let $m^d := \begin{pmatrix} 1 \\ X \\ \vdots \\ X^{d-1} \end{pmatrix}$. There are no positive semidefinite matrices A_0 and A_1 of dimension $d \times d$ with $d < 3$ such that $f = ((m^d)^T A_0 m^d) + ((m^d)^T A_1 m^d) \cdot g_1 + ((m^d)^T A_2 m^d) \cdot g_2$. However, there exist positive semidefinite matrices that satisfy these conditions with $d = 3$. We obtain the following.

$$A_0 = \begin{pmatrix} 6.555009 \times 10^{-7} & 0 & -0.405252 \\ 0 & 0.810504 & -27.1442 \\ -0.405252 & -27.1442 & 7.44974 \times 10^6 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 3.72841 \times 10^6 & -888.157 & -0.386659 \\ -888.157 & 0.275157 & 0.0000923 \\ -0.386659 & 0.0000923 & 0 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 3.72844 \times 10^6 & 888.152 & -0.388337 \\ 888.152 & 0.274097 & -0.0000923 \\ -0.388337 & -0.0000923 & 0 \end{pmatrix}$$

The certificate obtained with this approach is

$$\begin{aligned} f &= ((m^3)^T A_0 m^3) + ((m^3)^T A_1 m^3) \cdot g_1 + +((m^3)^T A_2 m^3) \cdot g_2 \\ &\approx (0.000256028 + 4.08976 * 10^-10X - 1582.84X^2)^2 + (0. + 0.90028X - 30.1508X^2)^2 \\ &+ \quad (0. + 2223.38X^2)^2 \\ &+ \quad ((1930.92 + 0.459964X - 0.000201115X^2)^2 + (0. + 0.25006X + 8.44035 * 10^-7X^2)^2 \\ &\qquad + (0. + 0.0000809667X^2)^2) \cdot g_1 \\ &+ ((1930.91 - 0.459968X - 0.000200247X^2)^2 + (0. + 0.252162X + 7.45169 * 10^-7X^2)^2 \\ &\qquad + (0. + 0.0000809673X^2)^2) \cdot g_2 \end{aligned}$$

In expanded form, the degrees of the certificates are $\deg((m^3)^T A_0 m^3) = 4$, $\deg((m^3)^T A_1 m^3) = 4$ and $\deg((m^3)^T A_2 m^3) = 4$.

Example 6.16 In this example, we compare the results of the proposed method and the method in [8]. The method in [8] requires an associated ideal I to be zero-dimensional; a certificate in such a context is of the form $\sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i + \sum_{i=1}^{t} p_i \cdot h_i$ for $G = \{g_1, \ldots, g_s\}$ encoding inequality constraints, and $H = \{h_1, \ldots, h_t\}$ encoding equality constraints; hence σ_i are sums of squares and p_i are polynomials. Although the certificate problem for quadratic modules does not explicitly incorporate polynomial equalities, such equalities can be enforced by encoding an ideal using two inequality constraints. We illustrate this encoding in the example below.

Consider $G = \{g_1, g_2\}$ where $g_1 = (X+1)^3(X-1)$ and $g_2 = -(X+1)(X-1)^3$ and $f = X + 1$ as the input polynomial. The semialgebraic set of G corresponds to two isolated points $\{-1, 1\}$, so we can include the equality constraint $I = \langle h_1 \rangle$ where $h_1 = (X+1)(X-1)$ without modifying the semialgebraic set.

First, we use our proposed method. In this case, we identify f as pure with respect to g_1; in fact, $f \in \mathrm{QM}(g_2)$. Thus, we use the MonogenicCert algorithm. For the latter, we use the SOSBasicLemma algorithm with inputs $(X+1)$to obtain $1 = \sigma \cdot (X+1) + \frac{1}{8} \cdot (-(X-1)^3)$ where $\sigma = \frac{7}{8}(-\frac{2}{7}X + 1)^2 + \frac{3}{56}X^2$ is a sums of squares. We obtain a certificate of f by multiplying f by both sides of the previous equation, i.e., $f = \frac{7}{8}((-\frac{2}{7}X + 1)(X+1))^2 + \frac{3}{56}(X(X+1))^2 + \frac{1}{8} \cdot g_2$.

Now, we use the method in [8]. We obtain a witness for the condition $\langle I : f \rangle + \langle f \rangle = \langle 1 \rangle$, in this case we have $(-\frac{1}{2})(X-1) \in \langle I : f \rangle$ and $\frac{1}{2}(X+1) \in \langle f \rangle$ and $1 = (-\frac{1}{2})(X-1) + \frac{1}{2}(X+1)$. Multiplying f to both sides of the previous equation, we obtain $f = \frac{1}{2}(X+1)^2 + (-\frac{1}{2}) \cdot h_1$.

6.7 Certificate computation using Shang et al's generators

In her Ph.D. thesis[4], the author characterizes Archimedean quadratic modules in one variable using the end points of the semialgebraic set induced by the generators as well as the orders of the generators and the input polynomial. This characterization is formalized as the signature in [81]. There exists an equivalent set of generators consisting of three polynomials encoding the information from the signature.

The authors in [81] reduced the minimal number of generators from three to two by incorporating additional factors into the generators so that the new set of generators describes the same signature. For the rest of this section, we refer to these two generators as g_1 and g_2. The 2Generators algorithm in [81] computes g_1 and g_2 by initially assigning them -1 and 1, respectively, and updating them for each endpoints in the semialgebraic set $\mathcal{S}(G)$, by multiplying polynomials involving the endpoints with the appropriate order according to the signature of $\mathrm{QM}(G)$. In Table 6.1, we describe the $i^{th}-$update done in the main loop of the 2Generators algorithm.

Interval	$g_1^{new} := g_1^{old}(X-a_i)^{\kappa_i^+}(X-b_i)^{\kappa_i^-}$ $g_2^{new} := g_2^{old}$
Type A	$g_1^{new} := g_1^{old}(X-a_i)^{\kappa_i}$ $g_2^{new} := g_2^{old}$
Type B	$g_1^{new} := \begin{cases} g_1^{old}(X-a_i)^{\kappa_i^-} & \text{if } i=1 \\ g_1^{old}(-(X-a_i))^{\kappa_i^+} & \text{if } i=k \\ g_1^{old}(X-a_i)^{\kappa_i^+}(X-(a_i+\frac{a_{i+1}-a_i}{5})) & \text{otherwise} \end{cases}$ $g_2^{new} := \begin{cases} g_2^{old}(X-a_i)^{\kappa_i^+} & \text{if } i=1 \\ g_2^{old}(-(X-a_i))^{\kappa_i^-} & \text{if } i=k \\ g_2^{old}(X-a_i)^{\kappa_i^-}(X-(a_i+2\frac{a_{i+1}-a_i}{5})) & \text{otherwise} \end{cases}$
Type C	$g_1^{new} := g_1^{old}(X-a_i)^{\kappa_i}$ $g_2^{new} := \begin{cases} g_2^{old}(X-a_i)^{\kappa_i^+} & \text{if } i=1 \\ g_2^{old}(X-a_i)^{\kappa_i^+}(X-(a_i-\frac{a_i-b_{i-1}}{5})) & \text{otherwise} \end{cases}$
Type D	$g_1^{new} := g_1^{old}(X-a_i)^{\kappa_i}$ $g_2^{new} := \begin{cases} g_2^{old}(-(X-a_i))^{\kappa_i^-} & \text{if } i=k \\ g_2^{old}(X-a_i)^{\kappa_i^-}(X-(a_i+\frac{a_{i+1}-a_i}{5})) & \text{otherwise} \end{cases}$

Table 6.1: i^{th}-update operation in main loop of 2Generators

In this section, we provide a method for computing certificates for members in

$\mathrm{QM}(g_1, g_2)$. We apply the QMCert algorithm to nonnegative factors of the input polynomial. Then, we directly multiply these nonnegative factors to obtain the original input polynomial. The latter provides a certificate in $\mathrm{QM}(g_1, g_2, g_1g_2)$. To lift the previous certificate to $\mathrm{QM}(g_1, g_2)$, we use a specialized approach to find a certificate of g_1g_2 in $\mathrm{QM}(g_1, g_2)$. In Section 6.7.1, we discuss a method to obtain a certificate of g_1g_2 in $\mathrm{QM}(g_1, g_2)$ distinguishing two cases whether $\mathcal{S}(g_1, g_2)$ contains isolated points of type B or not. Section 6.7.2 presents an algorithm to compute a certificate in Shang's et al's two generators and compares the output in QMCert.

6.7.1 Certificate of product of generators

In this section, we address the problem of computing certificates for g_1g_2 in $\mathrm{QM}(g_1, g_2)$.

$\mathcal{S}(g_1, g_2)$ does not contain isolated points of type B

If the associated semialgebraic set $\mathcal{S}(g_1, g_2)$ does not contain isolated points of type B, the certificate of the product of generators can be reduced to a monogenic certificate problem. The following theorem characterizes the membership problem of monogenic quadratic modules.

Theorem 6.17 *Let g_1, g_2 be the generators produced by the* 2Generators *algorithm. If $\mathcal{S}(g_1, g_2)$ does not have isolated points of type B then $g_1g_2 \in \mathrm{QM}(g_1)$.*

Proof. By inspection of the 2Generators algorithm, if $\mathcal{S}(g_1, g_2)$ does not have isolated points of type B, then g_1 is of the form

$-\prod_{\substack{i \\ a_i<b_i}} (X-a_i)^{\kappa_i^+}(X-b_i)^{\kappa_i^-}\prod_{\substack{i \\ a_i=b_i}}(X-a_i)^{\kappa_i}$; hence,

$$\mathcal{S}(g_1) = \bigcup_{\substack{i \\ a_i<b_i}} [a_i, b_i] \cup \bigcup_{\substack{i \\ a_i=b_i}} \{a_i\}$$

so $\mathcal{S}(g_1)$ is bounded.

We notice that g_2 is nonnegative over $\mathcal{S}(g_1)$ as g_2 evaluates to zero at the isolated points in $\mathcal{S}(g_1)$ and it evaluates to a positive value over the regular intervals in $\mathcal{S}(g_1)$; otherwise, suppose that there exists $x \in [a_i, b_i]$ for some $[a_i, b_i] \subseteq \mathcal{S}(g_1)$ so that $g_2(x) < 0$, then the latter leads to a contradiction as $[a_i, b_i] \subseteq \mathcal{S}(g_1, g_2)$. Therefore, we find that g_1g_2 is also nonnegative over $\mathcal{S}(g_1)$.

At every end point of $\mathcal{S}(g_1)$ we have $\mathrm{ord}_a(g_1g_2) - \mathrm{ord}_a(g_1) = \mathrm{ord}_a(g_2) = 0 \in 2\mathbb{N}$ and $\epsilon_a(g_1g_2) = \epsilon_a(g_1)$ and for every isolated point of $\mathcal{S}(g_1)$, we have $\mathrm{ord}_a(g_1g_2) \geq \mathrm{ord}_a(g_1)$; therefore, by Theorem 2.15, we have $g_1g_2 \in \mathrm{QM}(g_1)$. □

Example 6.18 Consider $g_1 = -(X+1)^2(X-1)^2$ and $g_2 = -(X+1)(X-1)$. We can see that $g_1g_2 = (X+1)^3(X-1)^3 \in \mathrm{QM}(g_1)$. We apply the MonogenicCert algorithm with inputs g_1g_2 and g_1.

First, we reduce both the input and the generator by ignoring the roots of even multiplicities in line 4 of the MonogenicCert algorithm, we have $(X+1)(X-1) \in \mathrm{QM}(-1)$, its certificate is $(X+1)(X-1) = (\frac{(X+1)(X-1)+1}{2})^2 + (\frac{(X+1)(X-1)-1}{2})^2 \cdot (-1)$. Multiplying $(X+1)^2(X-1)^2$ to both sides of the previous equation, we obtain

$$\begin{aligned} g_1g_2 &= \frac{1}{4}\left((X+1)(X-1)+1\right)(X+1)(X-1))^2 \\ &\quad + \frac{1}{4}\left((X+1)(X-1)-1\right)^2 \cdot g_1 \\ &\quad \in \mathrm{QM}(g_1) \end{aligned}$$

$\mathcal{S}(g_1, g_2)$ contains isolated points of type B

If $\mathcal{S}(g_1, g_2)$ contains isolated points of type B we consider the following approach. Let $g_1 = \beta_1 \overline{g_1}$ and $g_2 = \beta_2 \overline{g_2}$ be such that both β_1, β_2 are the product of the updates of Table 6.1 related to isolated points of type B in $\mathcal{S}(g_1, g_2)$.

Example 6.19 We discuss the factors β_1 and β_2 from above of the Shang et al's two generators using a simple example. For the latter, we show step by step the updates of the two generators using the 2Generators algorithm.

Consider $G = \{(X+2)(X+1)^2X, X(X-1)^2(X-2), -(X+2)(X-2)\}$. The semialgebraic set of G is $\{-2, -1, 0, 1, 2\}$; notice that $-1, 1$ are isolated points of type A, while the rest of isolated points of $\mathcal{S}(G)$ are of type B. The generators obtained by 2Generators are $g_1 = (X+2)(X+1)^2X(X-\frac{1}{5})(X-1)^2(X-2)$ and $g_2 = -(X+2)X(X-\frac{2}{5})(X-2)$. Using Table 6.1, we show the updates to g_1, g_2 in the main loop of 2Generators.

- Initial step: $g_1 = -1, g_2 = 1$
- Isolated point at $x = -2$: $g_1 = -(X+2), g_2 = (X+2)$
- Isolated point at $x = -1$: $g_1 = -(X+2)(X+1)^2, g_2 = (X+2)$
- Isolated point at $x = 0$: $g_1 = -(X+2)(X+1)^2X(X-\frac{1}{5}), g_2 = (X+2)X(X-\frac{2}{5})$
- Isolated point at $x = 1$: $g_1 = -(X+2)(X+1)^2X(X-\frac{1}{5})(X-1)^2, g_2 = (X+2)X(X-\frac{2}{5})$
- Isolated point at $x = 2$: $g_1 = (X+2)(X+1)^2X(X-\frac{1}{5})(X-1)^2(X-2), g_2 = -(X+2)X(X-\frac{2}{5})(X-2)$

From this sequence of updates to g_1 (resp. g_2), we see that the polynomials $(X+2)$, $X(X-\frac{1}{5})$ and $-(X-2)$ (resp. $(X+2)$, $X(X-\frac{2}{5})$ and $-(X-2)$) are the

updates related to isolated points of type B. Hence, $\beta_1 = -(X+2)X(X-\frac{1}{5})(X-2)$ and $\beta_2 = -(X+2)X(X-\frac{2}{5})(X-2)$.

In the following proposition, we prove that β_2 is nonnegative over $\mathcal{S}(\text{compact}_{\{g_1,g_2\}}(g_2))$.

Proposition 6.20 *Let β_2 be as defined above. We have that β_2 is nonnegative over $\mathcal{S}(\text{compact}_{\{g_1,g_2\}}(g_2))$. Furthermore, we can compute a certificate of β_2 in $\text{QM}(\text{compact}_{\{g_1,g_2\}}(g_2))$.*

Proof. Let $S := \mathcal{S}(\text{compact}_{\{g_1,g_2\}}(g_2))$. We will prove that each of the polynomials in the updates in Table 6.1 involving isolated points of type B is nonnegative over S. We address each of the following cases:

- The first isolated point of $\mathcal{S}(g_1, g_2)$ is of type B: The factor of g_2 is $(X-a_0)^{\kappa_0^+}$. The semialgebraic set of the latter is $[a_0, \infty) \supseteq S$.
- The last isolated point of $\mathcal{S}(g_1, g_2)$ is of type B: The factor of g_2 is $-(X-a_k)^{\kappa_k^-}$. The semialgebraic set of the latter is $(-\infty, a_k] \supseteq S$.
- The i^{th} isolated point of $\mathcal{S}(g_1, g_2)$ is of type B: The factor of g_2 is $(X - a_i)^{\kappa_i^-}(X - (a_i + 2\frac{a_{i+1}-a_i}{5}))$. The semialgebraic set of the latter is $(-\infty, a_i] \cup [a_i + 2\frac{a_{i+1}-a_i}{5}, \infty) \supseteq S$.

Since the polynomial β_2 is the product of the polynomials in the updates in Table 6.1 that involve isolated points of type B, then β_2 is nonnegative over $\mathcal{S}(\text{compact}_{\{g_1,g_2\}}(g_2))$.

The polynomial $\frac{\text{compact}_{\{g_1,g_2\}}(g_2)}{\beta_2}$ is a product of factors similar to β_2 that involve isolated points of type C and D. Hence, it is nonnegative over $\mathcal{S}(\text{compact}_{\{g_1,g_2\}}(g_2))$

following a similar proof above. Therefore, we can use the SOSBasicLemma algorithm with inputs $\beta_2, \frac{\text{compact}_{\{g_1,g_2\}}(g_2)}{\beta_2}$ to obtain sums of squares σ, τ such that $1 = \sigma\beta_2 + \tau\frac{\text{compact}_{\{g_1,g_2\}}(g_2)}{\beta_2}$. Multiplying β_2 by the previous equation, we obtain $\beta_2 = \sigma\beta_2^2 + \tau \cdot \text{compact}_{\{g_1,g_2\}}(g_2)$. Since the compactification algorithm provides a certificate for $\text{compact}_{\{g_1,g\}}(g_2)$ in $\text{QM}(g_1, g_2)$, $\beta_2 \in \text{QM}(g_2)$ and its certificates are computable. □

Proposition 6.21 *Let β_2 be as defined above and let $\mathcal{B} = \{a_{p_1}, \ldots, a_{p_m}\}$ be isolated points of type B of $\mathcal{S}(g_1, g_2)$. The semialgebraic set of the polynomial $\beta_2 g_1$ is bounded.*

Proof. Inspecting Table 6.1, if $a_0 \notin \mathcal{B}$ and $a_k \notin \mathcal{B}$ the polynomials β_1, β_2 are

$$\beta_1 = \prod_{i=1}^{m} (X - a_{p_i})^{\kappa_{p_i}^+}(X - (a_{p_i} + \frac{d_{p_i}}{5}))$$
$$\beta_2 = \prod_{i=1}^{m} (X - a_{p_i})^{\kappa_{p_i}^-}(X - (a_{p_i} + \frac{2d_{p_i}}{5}))$$

Whether a_0 (resp. a_k) is in $\mathcal{B}$, the polynomials β_1 and β_2 above also include the factor $(X - a_0)^{\kappa_0^-}$ and $(X - a_0)^{\kappa_0^+}$ (resp. $-(X - a_k)^{\kappa_k^+}$ and $-(X - a_k)^{\kappa_k^-}$). In all of the cases, we have the following

$$\beta_1\beta_2 = \prod_{i=1}^{m} (X - a_{p_i})^{\kappa_{p_i}^+ + \kappa_{p_i}^-}(X - (a_{p_i} + \frac{d_{p_i}}{5}))(X - (a_{p_i} + \frac{2d_{p_i}}{5})) \tag{6.3}$$

Similarly, since $\overline{g_1}$ does not contain roots associated with isolated points of type B, we have the following:

$$\overline{g_1} = -\left(\prod_{\substack{i \\ a_i < b_i}} (X - a_i)^{\kappa_i^+}(X - b_i)^{\kappa_i^-}\right)\left(\prod_{\substack{i \\ a_i = b_i, a_i \notin \mathcal{B}}} (X - a_i)^{\kappa_i}\right) \tag{6.4}$$

From the latter, by multiplying Equation (6.3) and Equation (6.4), we obtain $\mathcal{S}(\beta_1\beta_2\overline{g_1}) = \mathcal{S}(\beta_2 g_1) = \bigcup_{\substack{i \\ a_i<b_i}} [a_i, b_i] \cup \bigcup_{\substack{i \\ a_i=b_i}} \{a_i\}$
$\cup \bigcup_{a_{p_i}\in\mathcal{B}} [a_{p_i} + \frac{d_{p_i}}{5}, a_{p_i} + \frac{2d_{p_i}}{5}]$, so $\mathcal{S}(\beta_2 g_1)$ is bounded. □

Finally, we use the following theorem to compute a certificate of g_1g_2 in $\mathrm{QM}(g_1, g_2)$:

Theorem 6.22 *Let $g_1 = \beta_1\overline{g_1}$ and $g_2 = \beta_2\overline{g_2}$ be as defined above and let $\mathcal{B} = \{a_{p_1}, \ldots, a_{p_m}\}$ be the isolated points of type B of $\mathcal{S}(g_1, g_2)$. The certificates of g_1g_2 in $\mathrm{QM}(g_1, g_2)$ are computable.*

Proof. Using Theorem 6.17, we know that $\overline{g_1}\overline{g_2} \in \mathrm{QM}(\overline{g_1})$. Hence, we use the MonogenicCert algorithm to compute $\sigma_0, \sigma_1 \in \sum \mathbb{A}[X]^2$ such that $\overline{g_1}\overline{g_2} = \sigma_0 + \sigma_1\overline{g_1}$. Multiplying $\beta_1\beta_2$ by the previous equation, we obtain

$$\begin{aligned} g_1g_2 &= \beta_1\beta_2\sigma_0 + \sigma_1 \cdot \beta_2 g_1 \\ &= \sigma_0 \prod_{i=1}^{m} (X - a_{p_i})^{K^+_{p_i}+K^-_{p_i}} (X - (a_{p_i} + \frac{d_{p_i}}{5}))(X - (a_{p_i} + \frac{2d_{p_i}}{5})) \\ &+ \sigma_1 \cdot \beta_2 g_1 \end{aligned} \tag{6.5}$$

Since each $K^+_{p_i} + K^-_{p_i} \in 2\mathbb{N}$ then $\sigma_0 \prod_{i=1}^{m} (X - a_{p_i})^{K^+_{p_i}+K^-_{p_i}}$ is a sums of squares. Additionally, each $(X - (a_{p_i} + \frac{d_{p_i}}{5}))(X - (a_{p_i} + \frac{2d_{p_i}}{5})) \in \mathrm{QM}(\mathrm{compact}_{\{g_1,g_2\}}(g_1))$ as they are nonnegative over $\mathcal{S}(\ \mathrm{compact}_{\{g_1,g_2\}}(g_1))$. Hence,

$$\prod_{i=1}^{m} (X - (a_{p_i} + \frac{d_{p_i}}{5}))(X - (a_{p_i} + \frac{2d_{p_i}}{5})) \in \mathrm{QM}(\mathrm{compact}_{\{g_1,g_2\}}(g_1))$$

We use the MonogenicCert algorithm to compute a certificate for the latter. Since the compactification algorithm provides certificates of $\mathrm{compact}_{\{g_1,g_2\}}(g_1)$ in

$\mathrm{QM}(g_1, g_2)$, we have a certificate of $\beta_1\beta_2\sigma_0$ in $\mathrm{QM}(g_1, g_2)$. To conclude our proof, we need to compute a certificate of $\beta_2 g_1$ in $\mathrm{QM}(g_1, g_2)$.

Using the NonNeg algorithm, we compute the nonnegative factors of g_1 and β_2, let $B := \mathsf{NonNeg}(g_1) \cup \mathsf{NonNeg}(\beta_2)$. By Proposition 6.21, we know that $\mathcal{S}(\beta_2 g_1)$ is bounded, so the semialgebraic set $\mathcal{S}(B)$ is also bounded. Thus, we can use the procedure in Appendix B to compute a certificate of the product $\beta_2 g_1$ in $\mathrm{QM}(B)$. We can compute the certificate for each nonnegative factor in B using a combination of the MonogenicCert algorithm and the compactification procedure.

Finally, multiplying σ_1 by the certificate of $\beta_2 g_1$ obtained provides a certificate of the second summand in Equation (6.5). Therefore, $g_1 g_2 \in \mathrm{QM}(g_1, g_2)$ as a certificated is obtained from the procedure described above. □

6.7.2 Certificate of members using Shang et al's generators

Given an input polynomial $h \in \mathrm{QM}(g_1, g_2)$, we use the following approach to compute a certificate of h in $\mathrm{QM}(g_1, g_2)$.

1. Decompose h into two factors. The first factor, h_{pos}, which is strictly positive over $\mathcal{S}(g_1, g_2)$ and the second one, $h_{non-neg}$, which is a nonnegative polynomial over $\mathcal{S}(g_1, g_2)$.
2. Use the Certificate algorithm to compute a certificate of f_{pos} in $\mathrm{QM}(g_1, g_2)$.
3. We use the NonNeg algorithm to obtain the nonnegative factors of $h_{non-neg}$.
4. For each nonnegative factor $h_{non-neg,i}$ in $\mathsf{NonNeg}(h_{non-neg})$, we use $\mathsf{QMCert}(h_{non-neg,i}, \{g_1, g_2\})$ to compute a certificate in $\mathrm{QM}(g_1, g_2)$.
5. Finally, we multiply the certificates of h_{pos} and each $h_{non-neg,i}$ and rearrange the sums of squares multipliers. The product of these certificates might be of

the form $s_0 + s_1 \cdot g_1 + s_2 \cdot g_2 + s_3 \cdot g_1 g_2$ where each s_i is a sums of squares. Using the certificates of the product of the generators $g_1 g_2$ from Section 6.7, we have a representation of h in $\mathrm{QM}(g_1, g_2)$.

6.8 Summary

This chapter provides the QMCert algorithm to compute a certificate for a member f in an Archimedean quadratic module $\mathrm{QM}(G)$. The key idea is due to the concept of $\mathrm{LocalRelativeGens}(f, G, a)$, which identifies a "local" generator with respect to a zero a of f which is also an endpoint in $\mathcal{S}(G)$ and uses the ideas of pure and mixed polynomials to identify whenever the certificates of an input polynomial can be computed using a single polynomial; in this case we use the MonogenicCert algorithm from Chapter 5.

If the polynomial is not pure, the method multiplies additional strictly positive factors to the input polynomial such that the nonnegative factors of the resulting polynomial are all pure. To combine these, we use the property of the intermediate set of generators $H_{f,G}$ that allow us to compute a certificate of the products of its elements.

Previous work in Chapter 5, especially the MonogenicCert algorithm, is widely used whenever the input polynomial is pure or mixed with respect to the original set of generators. For the former case, the problem reduces entirely to a monogenic certificate problem, while the latter case uses a construction to compute certificates of nonnegative polynomials using the generators in $\mathrm{LocalRelativeGens}(f, G, a)$; the latter are used as the monogenic generators.

In Section 6.7, we refine the QMCert algorithm to compute certificates of members using the two generators in [81]. These generators enjoy the property that there is no

overlap in their roots with respect to the signature, which makes it clear the choice of "local" generators addressing each common root of f and the endpoints of $\mathcal{S}(G)$.

Chapter 7

Certifying the Emptiness of Monogenic Quadratic Modules in $\mathbb{A}[\overline{X}]$

A system of polynomial equations has no solution whenever we can certify that 1 belongs to the radical ideal generated by the polynomials involved in the system via Nullstellensatz [38, 73]. Similarly, when a system of polynomial inequalities has no solutions, or equivalently, the semialgebraic $\mathcal{S}(G)$ associated with the generators is empty, the Positivstellensatz (Theorem 2.1) provides an algebraic characterization that -1 belongs to the preordering $\mathrm{PO}(G)$.

This certification perspective naturally leads us to examine the quadratic modules themselves. A quadratic module module Q is called proper if it does not contain -1. The literature on real algebraic geometry concerns mostly problems related to proper quadratic modules, as the non-proper quadratic modules contain all the polynomials in the polynomial ring. However, it is not obvious to formally certify that a quadratic module is non-proper unless a certificate of -1 is given.

Previous chapters addressed certificate problems in the univariate case. In this chapter, we consider the problem of computing a certificate of -1 for monogenic quadratic modules $\mathrm{QM}(g)$ when $\mathcal{S}(g)$ is empty in the multivariate case. We focus on this particular case as $\mathrm{QM}(g)$ is a preordering and thus Positivstellensatz justifies the membership of -1.

In Section 7.1, we describe an algorithm to compute a certificate for -1 in these kinds of quadratic modules. In addition, we discuss examples of several variables. Section 7.2 concludes this chapter summarizing the main results.

7.1 Certificate of -1

Any quadratic module generated by a single polynomial g is a preordering. Hence, by Theorem 2.1, we know that $-1 \in \mathrm{QM}(g)$ if and only if $\mathcal{S}(g)$ is empty. This kind of quadratic module includes the entire set of polynomials in $\mathbb{A}[\overline{X}]$; in fact, this is clear using the formula $f = \left(\frac{f+1}{2}\right)^2 + \left(\frac{f-1}{2}\right)^2 \cdot (-1)$.

The focus of this section therefore becomes on finding a certificate of -1 for the inconsistent case. The approach of the algorithm aims to find two sums of squares in a Bézout-like identity for -1 using the generator g and $\epsilon - g$ for some positive ϵ. To handle the certificates of nonnegative polynomials, our algorithm uses the ExtLasserre algorithm from [82] discussed in Section 2.1.10.

Theorem 7.1 *Let $g \in \mathbb{A}[\overline{X}]$ such that $\mathcal{S}(g)$ is empty. There exists a positive constant ϵ such that $\epsilon - g$ is strictly positive over $\mathbb{R}^n$.*

Proof. Let ϵ_1 be the maximum value of g over $\mathbb{R}^n$, i.e., $\epsilon_1 \geq g(\bar{x})$ for every $\bar{x} \in \mathbb{R}^n$. Since $\mathcal{S}(g)$ is empty, we have $\epsilon_1 < 0$. Set $\epsilon := -\frac{\epsilon_1}{2}$. We will prove that $-\epsilon - g$ is strictly positive over $\mathbb{R}^n$.

Suppose, by contradiction, there exists $\bar{y} \in \mathbb{R}^n$ such that $-\epsilon - g(\bar{y}) \leq 0$, and hence $\epsilon_1 \leq 2g(\bar{y})$. By the choice of ϵ_1, we have $\epsilon_1 \geq g(\bar{x})$ for every $\bar{x} \in \mathbb{R}^n$, so $\epsilon_1 \geq g(\bar{y})$; thus, we have $2g(\bar{y}) \geq \epsilon_1 \geq g(\bar{y})$. Subtracting $g(\bar{y})$ from the previous inequalities we obtain $g(\bar{y}) \geq \epsilon_1 - g(\bar{y}) \geq 0$, which means $\bar{y} \in \mathcal{S}(g)$. This is a contradiction, as $\mathcal{S}(g)$ is empty. □

Our procedure for computing a certificate of -1 in the inconsistent case is the following.

Algorithm 19: Computing a certificate of -1 in inconsistent monogenic quadratic modules

Input: $g \in \mathbb{A}[\overline{X}]$

Output: $\sigma_0, \sigma_1 \in \mathbb{A}[\overline{X}]$

Requires: $\mathcal{S}(g)$ is empty

Ensures: $-1 = \sigma_0 + \sigma_1 \cdot g$ and $\sigma_0, \sigma_1 \in \sum \mathbb{A}[\overline{X}]^2$

1 Let ϵ_1 be the maximum value of g over $\mathbb{R}^n$

2 Let $\epsilon := -\frac{\epsilon_1}{2}$

3 **if** $-\epsilon - g$ *is a sums of squares* **then**

4 $\quad \sigma_0 := \frac{1}{\epsilon}(-\epsilon - g), \sigma_1 := \frac{1}{\epsilon}$

5 **else**

6 $\quad$ Use ExtLasserre to compute a certificate of $-\epsilon - g$ in $\mathrm{QM}(g)$, i.e., $-\epsilon - g = \sigma_0^{'} + \sigma_1^{'} \cdot g$

7 $\quad \sigma_0 := \frac{1}{\epsilon}\sigma_0^{'}, \sigma_1 := \frac{1}{\epsilon}(1 + \sigma_1^{'})$

8 **return** σ_0, σ_1

7.1.1 Examples

In the following examples, we modify the examples in Section 2.1.9 by subtracting a negative value from the generators to obtain an empty case. First, we start with a

simple example in the univariate case.

Example 7.2 Consider the generator $g = -(X^2 + 1)$. The maximum value of g is obtained at $x = 0$ with $g(0) = -1$. Hence, the value of ϵ is $\frac{1}{2}$. In this case, we have that $-\frac{1}{2} - g$ is $\frac{1}{2} + X^2$, which is a sums of squares; hence, we are done. The certificate of -1 obtained is $-1 = 1 + 2X^2 + 2 \cdot g$.

Example 7.3 Let us consider the generator $g = -X^4 - X^2Y^2 + 2X^2Y + XY^2 - Y^2 - 1$. The maximum value of g is $-1 + \frac{1}{72}(44 + \frac{88064}{(5095441+1433475\sqrt{345})^{\frac{1}{3}}} - (5095441 + 1433475\sqrt{345})^{\frac{1}{3}}) \approx -0.921668$ obtained at $x \approx 0.814681, y \approx 0.781727$. Hence, we set ϵ to $\frac{1}{2}(1 - \frac{1}{72}(44 + \frac{88064}{(5095441+1433475\sqrt{345})^{\frac{1}{3}}} - (5095441 + 1433475\sqrt{345})^{\frac{1}{3}}))$. In this case, we have that $-\epsilon - g$ is a sums of squares, i.e.,

$$
\begin{aligned}
-\epsilon - g = &(\frac{(-72X(915X + 34Y) + 1037\alpha_1)\alpha_2}{12444})^2 \\
&+ (-\frac{13X^2}{30} + X(\frac{1}{61} - \frac{Y}{2}) + Y)^2 \\
&+ (\frac{X(1673940 + 13481X - 1059814Y)}{1830\sqrt{948566}})^2 \\
&+ (\frac{X(63257(5327374X - 9953711Y) + 1807855200(915X + 34Y)\alpha_3)}{31110\sqrt{27899(37037758631 - 3615710400\alpha_3)}})^2 \\
&+ (\frac{1}{510}X^2\alpha_4)^2
\end{aligned}
$$

where

- $\alpha_1 = 116 - \frac{88064}{(-5095441+1433475\sqrt{345})^{\frac{1}{3}}} + (-5095441 + 1433475\sqrt{345})^{\frac{1}{3}}$
- $\alpha_2 = \frac{1}{3\sqrt{\frac{271806}{(11280+(6(205703889507-7215316775\sqrt{345}))^{\frac{1}{3}}+(6(205703889507+7215316775\sqrt{345}))^{\frac{1}{3}})}}}$
- $\alpha_3 = \frac{304560+(24293217942997686-852114480493950\sqrt{345})^{\frac{1}{3}}+27(6(205703889507+7215316775\sqrt{345}))^{\frac{1}{3}}}{66048858}$

- $\alpha_4 = 1/(459711166726897933921689399798536378 2/($ $67441107661581399414313997607446567817263 3$
$-(84314905830129330830021597061974504830029901831253328 00*6^{\frac{2}{3}})/(10446626370516172910914694309840959973050437-$
$3661024804021333421268853957645594081890 25\sqrt{345})^{\frac{1}{3}} -$
$385263550098120166279057200(6\ (10446626370516172910914694309840959973050437 -$
$3661024804021333421268853957645594081890 25\sqrt{345}))^{\frac{1}{3}}))^{\frac{1}{2}}$

Hence, we are done; the certificate obtained for -1 in $\mathrm{QM}(g)$ is

$$\begin{aligned}-1 = \frac{1}{\epsilon}((&\frac{(-72X(915X+34Y)+1037\alpha_1)\alpha_2}{12444})^2\\ &+(-\frac{13X^2}{30}+X(\frac{1}{61}-\frac{Y}{2})+Y)^2\\ &+(\frac{X(1673940+13481X-1059814Y)}{1830\sqrt{948566}})^2\\ &+(\frac{X(63257(5327374X-9953711Y)+1807855200(915X+34Y)\alpha_3)}{31110\sqrt{27899(37037758631-3615710400\alpha_3)}})^2\\ &+(\frac{1}{510}X^2\alpha_4)^2)+\frac{1}{\epsilon}\cdot g\end{aligned}$$

Example 7.4 Consider the generator $g = -1+XY-X^2-Y^2-Z^2$. The maximum value of g is -1 attained at $x = y = z = 0$. We set $\epsilon = \frac{1}{2}$. In this case, we have that $-\frac{1}{2} - g$ is a sums of squares, i.e., $-\frac{1}{2} - g = \frac{1}{2} + Z^2 + (Y - \frac{X}{2})^2 + \frac{3}{4}X^2$. Hence, the certificate of -1 we obtain is

$$\begin{aligned}-1 = &(1+2Z^2+2(Y-\frac{X}{2})^2+\frac{3}{2}X^2)\\ &+2\cdot g\end{aligned}$$

7.2 Summary

In this chapter, we discuss a simple method for certifying the emptiness of a monogenic quadratic module.

Section 7.1 discussed the case where $\mathcal{S}(g)$ is empty. The construction relies on a novel idea similar to the Basic Lemma Theorem 2.19. However, the difference between the construction and the Basic Lemma is that the former obtain an identity involving 1 while our construction finds an identity involving -1. Finally, nonnegative polynomials in the identity are lifted to a member in $\mathrm{QM}(g)$ using the ExtLasserre algorithm of [82].

Chapter 8

Conclusions

This thesis develops a method for computing certificates of members in Archimedean univariate quadratic modules specified by a finite set of generators in the univariate case, and addresses the problem of certifying the emptiness of quadratic modules in $\mathbb{A}[\overline{X}]$. For the first part, we started our study with simpler cases, such as the saturated and monogenic case. These cases provided interesting insights to tackle the general problem; the methods developed in the saturated case allow us to compute certificates of products that were possible to generalize to multiplicities greater than 1, and the monogenic case, in combination with the compactification method, provides a framework to compute certificates of nonnegative factors. The proposed methods are novel, symbolic, and obtain certificates in terms of the original generators.

For the second part of the thesis, we applied an optimization algorithm to compute an upperbound of g to obtain a Bézout-like identity for -1 and the ExtLasserre algorithm was used to handle nonnegative polynomials that are not sums of squares.

8.1 Summary of findings

We briefly summarize the techniques developed and the relevance of the work done in this thesis.

8.1.1 Chapter 3

The compactification method described in Algorithm 9 provides a novel approach to find a certificate of a member in a quadratic module such that this member has a bounded semialgebraic set. This is particularly useful for the purposes of the thesis as it allows us to use the SOS variant of the Basic Lemma construction with members that do not necessarily have bounded semialgebraic sets. The compactification algorithm can be seen as a special case of the Theorem 2.19 (Basic Lemma) applied to a simpler case involving a member of a quadratic module and a polynomial of degree at most 2. The similarities are that both methods compute strictly positive polynomials in a Bézout-like identity of 1. However, the compactification algorithm in Algorithm 9 does not use a complex construction as Algorithm 3 since the former is optimized for the univariate case.

One particular application that is of vital importance for this thesis work allows us to compute certificates of nonnegative factors of members in quadratic modules. This technique is used in Chapters 4, 5, and 6 of this thesis, as our typical approach aims to decompose the input polynomial into simpler nonnegative factors and combine them later using the multiplicative closure of quadratic modules in one variable. We emphasize that the compactification algorithm is symbolic and exact and does not rely on numerical approaches; on the other hand, it requires upfront information about the roots of the generators and input polynomial, which are algebraic numbers.

8.1.2 Chapter 4

For the saturated case, the set of generators have a unique equivalent set of natural generators determined by its semialgebraic set. The algorithms developed in this chapter are symbolic and exact. The computation of the natural generators works for different sets of generators defining the same semialgebraic set; this is one of the advantages of the proposed method, as the certificates of the input polynomial can be lifted from different sets of generators given they share the same signature. However, the second part of the proposed method changes for different sets of generators, since certificates of natural generators must be computed in terms of a given set of generators.

The Algorithm 12 has been implemented in `Maple` in the package `CertSatQM` and compared with `RealCertify` and Shang et. al's general method in a collection of examples. The experimental evaluation shows that the quality of the certificates obtained using our Algorithm 12 finds sums of squares with a smaller degree in contrast to the approaches reported in the existing literature.

8.1.3 Chapter 5

For the monogenic case, the proposed method is fully symbolic as it mostly relies on the Basic Lemma (Theorem 2.19). Our solution notices that the even multiplicities can always be "ignored" and reincorporated as the sums of squares multipliers in the representation of members of quadratic modules. This observation is crucial to obtain a practical algorithm which has been thoroughly tested using many examples.

We use a divide-and-conquer approach. The "divide" part splits nonnegative factors of f and finds certificates for each in $\mathrm{QM}(g)$. The "conquer" part multiplies these factors to obtain a certificate of f in $\mathrm{QM}(g)$; the latter is possible because

QM(g) is a preordering. First, we obtain a normal form for both the input polynomial f and the generator g by removing redundant factors, which are factors that do not change the signatures of f nor g with respect to the signature of g. Then, we formulate a subproblem ignoring the isolated points of even multiplicity in g. This allows us to use the constructions from Chapter 4 involving the Basic lemma [42] and Berg's lemma [4]. To obtain a certificate of the original problem, we bring back the isolated points removed by multiplying them directly with the previous certificate and rearrange terms so that they are absorbed in the sums of squares multipliers or the generator g.

We use the same theoretical tools from Chapter 4. In addition, we also used the characterization of monogenic quadratic modules given by Augustin [4]. The experimental comparison with Shang et al's alternative method shows our approach to find certificates with smaller degree.

8.1.4 Chapter 6

The approach for the general case combines the techniques developed in the saturated and monogenic cases. We compute a certificate of a member f in an Archimedean quadratic module QM(G). No restrictions are considered for these two inputs. We preprocess the input polynomial by multiplying additional factors that allow us to use the techniques in Chapter 5 in order to satisfy nonnegativity conditions.

We introduce an intermediate set of generators $H_{f,G}$ with the property that the certificates of products of its elements are computable and QM($H_{f,G}$) $\subseteq$ QM(G). This set contains the generalized natural generators of G. This result is a generalization of item 2.(a) in Chapter 4 above. We use a divide-and-conquer approach to compute certificates of members in QM(G). The "divide" part splits the preprocessed input polynomial into nonnegative factors and "matches" each of these factors

with a generator $h_i \in H_{f,G}$.

Certificates for the latter are obtained using the algorithm in Chapter 5 since this subproblem is a monogenic problem. The "conquer" part combines the certificates of each previous nonnegative factor and applies the property of the intermediate generators H that certificates of their products are computable. Finally, we remove the additional polynomials introduced in the preprocessing step using the Basic lemma [42] as the introduced polynomial satisfies the requirements of the lemma.

8.1.5 Chapter 7

We certify the emptiness of a monogenic quadratic module. For the latter, we refine the construction ExtLasserre of [82] to compute a certificate for -1. We developed a procedure to compute a Bézout-like identity for -1 and to lift the nonnegative polynomial multiplier to a member in $\mathrm{QM}(g)$ to obtain a certificate of -1 in $\mathrm{QM}(g)$.

8.2 Future work

The research presented in this thesis has contributed substantial advances to the verification and certificate computation of Archimedean quadratic modules. Nevertheless, several enhancements can be made to further improve the work presented here.

The Basic Lemma is a fundamental tool in the algorithms presented in this thesis, especially for the problems in the univariate case addressed in the chapters 4, 5,and 6. However, the original result from Theorem 2.19 applies to the multivariate case. We believe that this result can be further optimized for the univariate case. Some reasons to foresee possible improvements of this observation is due to the compactification

Algorithm compact in Algorithm 9; this algorithm showed a similar construction that is less complex than the construction in the original proof of the Basic Lemma and uses some properties applicable only in the univariate case, for example, that non-negative polynomials can be expressed as sums of squares.

Similarly, the Certificate algorithm from [82] is widely used in the algorithms of this thesis. The fundamental constructive results of the Certificate algorithm are a Lemma by Averkov [6] and a technique to approximate non-negative polynomials as sums of squares by Lasserre in [44]; both of these results hold for the multivariate case. Although the univariate case does not benefit from specializing Lasserre's approximation theorem [1], improving the construction of Averkov's lemma to the univariate case would optimize different aspects of certificate computation of strictly positive polynomials over compact semialgebraic sets in the univariate case. Special algorithms of this kind would also make the complexity analysis of them more suitable.

A significant part of the motivation in developing certificates for members in Archimedean quadratic modules is to use them in verification applications. The prototypical implementation of the algorithms developed in Chapter 4 and Chapter 5 requires a significant amount of work to be useful in this kind of application. The Algorithm 18 for the method QMCert has not yet been implemented as it requires a complete implementation of the algorithms for SaturatedCert and MonogenicCert.

Whenever a statement fails to satisfy a given property, formal methods require the production of a counterexample within a model of the theory. In the context of verification, another way to see the contributions of this thesis is in the proof-production of members in an Archimedean quadratic module. However, whenever a polynomial is not a member, currently there are no easy ways to report this result.

[1] This is because every non-negative polynomial in $\mathbb{R}[X]$ can be expressed as a sum of squares.

It will be of interest to develop methods to exhibit the non-membership of these polynomials in verification settings. Currently, the literature usually uses short proofs for examples such as in [86, Example, p. 168]. A suitable model should be easy to represent within formal methods. A good candidate for this task might be the signature method from [4, 81].

Appendix A

Auxiliary Constructions

A.1 The lifting operation

Throughout the thesis, we use the term "lifting" to indicate a substitution in the generators g_i's of a representation in some quadratic module by the certificate of the g_i's in another quadratic module. In this section, we formalize this notion and provide some examples to illustrate the idea.

Definition A.1 Let $G = \{g_1, \ldots, g_s\}$, $H = \{h_1, \ldots, h_t\}$ be such that $\mathrm{QM}(G) \subseteq \mathrm{QM}(H)$ and $f \in \mathrm{QM}(G)$. Assume that we have a certificate of $f = \sigma_0 + \sum_{i=1}^{s} \sigma_i \cdot g_i$ in $\mathrm{QM}(G)$ and certificates for each $g_i = \sigma_{0,i} + \sum_{j=1}^{t} \sigma_{j,i} \cdot h_j$ in $\mathrm{QM}(H)$. We refer to **lifting** f from $\mathrm{QM}(G)$ to $\mathrm{QM}(H)$ as the action of obtaining a certificate of f in $\mathrm{QM}(H)$ by replacing the certificates of each g_i in $\mathrm{QM}(H)$ with the certificate of f in $\mathrm{QM}(G)$, i.e., $f = (\sigma_0 + \sum_{i=1}^{s} \sigma_i \sigma_{0,i}) + \sum_{j=1}^{t} (\sum_{i=1}^{s} \sigma_i \sigma_{j,i}) \cdot h_j$.

Notice that if we consider certificates as vectors of dimension $1 \times n + 1$ where n is the number of generators, lifting can be formulated as a matrix multiplication operation, i.e.,

$$\begin{pmatrix}\sigma_0 & \sigma_1 & \cdots & \sigma_s\end{pmatrix}\begin{pmatrix}1 & 0 & \cdots & 0\\ \sigma_{0,0} & \sigma_{1,1} & \cdots & \sigma_{t,1}\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_{0,s} & \sigma_{1,s} & \cdots & \sigma_{t,s}\end{pmatrix}$$
$$= \begin{pmatrix}\sigma_0 + \sum_{i=1}^{s}\sigma_i\sigma_{0,i} & \sum_{i=1}^{s}\sigma_i\sigma_{1,i} & \cdots & \sum_{i=1}^{s}\sigma_i\sigma_{t,i}\end{pmatrix}$$

We formalize the previous observation in the following algorithm:

Algorithm 20: Lifting operation

Input: A certificate $\sigma_0, \sigma_1, \ldots, \sigma_s$ for some $f \in \mathrm{QM}(G)$ and certificates $\sigma_{0,i}, \sigma_{1,i}, \ldots, \sigma_{t,i}$ for each $g_i \in G$ in $\mathrm{QM}(H)$

Output: A certificate $\tau_0, \tau_1, \ldots, \tau_t$ of $f \in \mathrm{QM}(H)$

1 Let $v = \begin{pmatrix}\sigma_0 & \sigma_1 & \cdots & \sigma_s\end{pmatrix}$

2 Let $M = \begin{pmatrix}1 & 0 & \cdots & 0\\ \sigma_{0,0} & \sigma_{1,1} & \cdots & \sigma_{t,1}\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_{0,s} & \sigma_{1,s} & \cdots & \sigma_{t,s}\end{pmatrix}$

3 Let τ_i the the i^{th} entry in vM for $0 \leq i \leq t$

4 **return** $\tau_0, \tau_1, \ldots, \tau_t$

Example A.2 Consider the monogenic quadratic module $G := \{g\}$ where $g = -(X+3)(X+2)(X-2)(X-3)$. Its semialgebriac set is $S := [-3,-2] \cup [2,3]$ and its natural generators are $\mathrm{Nat}(S) = \{(X+3), (X+2)(X-2), -(X-3)\}$. Using Theorem 2.9, we can prove that $\mathrm{QM}(G)$ is saturated. From the latter and Theorem 2.8, we have $\mathrm{QM}(\mathrm{Nat}(S)) \subseteq \mathrm{QM}(G)$. We want to illustrate the lifting operation by computing a certificate of the input polynomial $f = (X+1)(X-1)$ in $\mathrm{QM}(\mathrm{Nat}(S))$ and then lifting the certificate to $\mathrm{QM}(G)$.

We obtain a certificate of the polynomial $(X+1)(X-1)$ using Lemma 2.21 using the natural generator $(X+2)(X-2)$. From the latter, we obtain $(X+1)(X-1) = \frac{3}{4}X^2 + \frac{1}{4} \cdot (X+2)(X-2)$. To lift the certificate of f in $\mathrm{QM}(\mathrm{Nat}(S))$ to $\mathrm{QM}(G)$, it is enough to find the certificate of the natural generator $(X+2)(X-2)$ in $\mathrm{QM}(G)$. We use the MonogenicCert algorithm for this; we obtain $(X+2)(X-2) = \frac{1}{5}((X+2)(X-2))^2 + \frac{1}{5} \cdot g$.

Finally, we obtain a certificate of f in $\mathrm{QM}(G)$ by lifting. We obtain

$$\begin{aligned}(X+1)(X-1) &= \frac{3}{4}X^2 + \frac{1}{4} \cdot (X+2)(X-2) \\ &= \frac{3}{4}X^2 + \frac{1}{4} \cdot \left(\frac{1}{5}((X+2)(X-2))^2 + \frac{1}{5} \cdot g\right) \\ &= \frac{3}{4}X^2 + \frac{1}{20}((X+2)(X-2))^2 + \frac{1}{20} \cdot g\end{aligned}$$

A.2 Auxiliary proofs for Section 4.3

Lemma A.3 *Let* $p := \alpha(X-a)(X-b)^m + \beta(X-c)^m(X-d)$ *with* $m \in 2\mathbb{N}+1$, $b, c \in \mathbb{A}$ *within* (a,d), *and* $\alpha := \frac{1}{(d-a)(d-b)^m}, \beta := \frac{1}{(a-c)^m(a-d)}$. *Then* p *is of the form* $q \cdot (X-a)(X-d)+1$ *for some* $q \in \sum \mathbb{A}[X]^2$.

Proof. Notice that $p(a) = p(d) = 1$. Using Euclid's division algorithm, the residue of p divided by $(X-a)(X-d)$ is at most of degree 1, i.e., $p = q(X-a)(X-d)+r$ with $\deg(r) \leq 1$. Let $r = \gamma_1 X + \gamma_2$. If $\gamma_1 = 0$ we are done as $\gamma_2 = 1$. Otherwise, $p(a) = r(a) = 1$, similarly $p(d) = r(d) = 1$. Hence, $r(a) = r(d)$, so $\gamma_1 a + \gamma_2 = \gamma_1 d + \gamma_2$, which implies that $a = d$. This contradicts the assumption that $a < d$.

Now, we will show $q \in \sum \mathbb{A}[X]^2$ to conclude the proof. We will show that the only real roots of the polynomial $p-1$ are a, d with multiplicity 1, which shows that

q must be a product of complex conjugate roots, which means that q is a sum of squares.

Suppose q has a real root e different from a and d. This implies $p(e) = 1$. We will prove this leads to a contradiction by showing that p evaluates to a value greater than 1 over $(-\infty, a)$ and (d, ∞) and smaller than 1 over (a, d).

First, write p as $p = \frac{p_1}{p_1(d)} + \frac{p_2}{p_2(a)}$ where $p_1 = (X - a)(X - b)^m$ and $p_2 = (X - c)^m(X - d)$. For the interval $x \in (d, \infty)$, we see that $\frac{p_1(x)}{p_1(d)}$ is strictly greater than 1 and $\frac{p_2(x)}{p_2(a)}$ is nonnegative; hence $p(x)$ is strictly greater than 1. A similar argument holds for the interval $(-\infty, a)$.

Now, let us consider the case where $x \in (a, d)$. If $b = c$ we have $p = \frac{p_2(a)+p_1(d)}{p_2(a)p_1(d)}(X - b)^m (X - r)$ where $r = \frac{ap_2(a)+dp_1(d)}{p_2(a)+p_1(d)}$. We can see that p has at most two real zeros which belong to the interval (a, d) as $a < r < d$. Since $p_1(d), p_2(a)$ are positive, the leading coefficient of p is positive. The minimum value p^* of p is attained within (a, d); this implies that p strictly decreasing the interval (a, p^*) and strictly increases in the interval (p^*, d). As $p(a) = p(d) = 1$, we conclude that p reaches a value less than 1 over (a, d).

Now, let us consider the case where $b < c$. We can check that $p(b) = \frac{b-d}{a-d}(\frac{b-c}{a-c})^m < 1$ and $p(d) = \frac{a-c}{a-d}(\frac{b-c}{b-d})^m < 1$. To finish this case, we analyze the intervals $(a, b), (b, c)$ and (c, d). Consider the interval (a, b). The polynomial $\frac{p_1}{p_1(d)}$ evaluates negative values on (a, b) and the polynomial $\frac{p_2}{p_2(a)}$ strictly decreases over (a, b) with maximum value 1; thus p evaluates a value less than 1 over the interval (a, b); similar reasoning handles the interval (c, d).

Now, we address the interval (b, c). First, we notice that

$$\begin{aligned}\frac{dp}{dX} &= \frac{1}{p_1(d)}((X-b)^m + m(X-a)(X-b)^{m-1}) \\ &+ \frac{1}{p_2(a)}((X-c)^m + m(X-d)(X-c)^{m-1}) \\ &= \frac{(X-b)^{m-1}}{(d-a)(d-b)^m}((m+1)X - (b+ma)) \\ &+ \frac{(X-c)^{m-1}}{(a-d)(a-c)^m}((m+1)X - (c+md))\end{aligned} \tag{A.1}$$

We want to show that $\frac{dp}{dX}$ is strictly increasing over (b,c). The polynomial $q_1 := \frac{(X-b)^{m-1}}{(d-a)(d-b)^m}((m+1)X-(b+ma))$ has at most two real roots, b and $\frac{b+ma}{m+1}$. The last is contained in the interval $[a,b]$, which means that the polynomial q_1 strictly increases for any point after b. Similarly, we can show that the polynomial $\frac{(X-c)^{m-1}}{(a-d)(a-c)^m}((m+1)X-(c+md))$ is strictly increasing for any point before c. Hence, from equation (A.1), we see that $\frac{dp}{dX}$ is strictly increasing over (b,c).

We have $\frac{dp}{dX}(b) = \frac{1}{b-c}\frac{m(b-d)+(b-c)}{a-d}(\frac{b-c}{a-c})^m < 0$ as $b < c$. Similarly, $\frac{dp}{dX}(c) = \frac{1}{c-b}\frac{m(a-c)+(b-c)}{a-d}(\frac{c-b}{d-b})^m > 0$. By the Intermediate Value theorem, we have that $\frac{dp}{dX}$ has a zero η in (b,c) and it is unique since $\frac{dp}{dX}$ is strictly increasing over (b,c). Furthermore, η corresponds to the global minimum value of p over $\mathbb{R}$. Hence $p(x) < 1$ for $x \in (b,c)$, as otherwise there would be a local maximum value of p in (b,c), which is not the case.

Finally, let us consider the case where $c < b$. For the interval (a,c), $\frac{p_1}{p_1(d)}$ is less than 1 and $\frac{p_2}{p_2(a)}$ is always nonnegative over (a,c), hence p is less than 1 over (a,c). A similar argument handles the interval (b,d). For the interval $[c,b]$ we notice that both $\frac{p_1}{p_1(d)}$ and $\frac{p_2}{p_2(a)}$ are non-positive, so p is less than 1 over $[b,d]$. Therefore, p is less than 1 over (a,d).

Therefore, q has no real roots and is a product of irreducible quadratics of the form $q = \prod_j ((X-a_j)^2 + b_j^2)^{e_j}$. Since the leading coefficient of p is positive, we see

that q is a sums of squares. □

Proposition A.4 *Let f be a polynomial such that $\mathcal{S}(f) = \bigcup_{i=0}^{k} [c_i, d_i]$ is bounded, divisible by $(X-d_j)^m(X-c_{j+1})^n$ where m is odd and n is even. Then $(X-d_j)^m(X-c_{j+1})^{n+1}$ belongs to $\mathrm{QM}(f)$. Furthermore, the certificates of $(X-d_j)^m(X-c_{j+1})^{n+1}$ in $\mathrm{QM}(f)$ are computable.*

Proof. First, we notice that $\epsilon_{d_j}(f) = -1$ as d_j is a right end point. Let $I = \{i \in \mathbb{N} \mid j+1 < i \leq k, \mathrm{ord}_{c_i}(f) \text{ is odd } \}$ be a set of indices.

If I is empty, this means that f only has end points of even multiplicity to the right of c_{j+1}. If so, we define the polynomial $s := (X-d_j)^{m-1}\prod_{i=j+1}^{k}(X-c_i)^{\mathrm{ord}_{c_i}(f)}$ and define $\hat{f} := \frac{f}{s}$. Notice that s is a sums of squares so the semialgebraic set of $\hat{f}$ is $\mathcal{S}(f)$ without the isolated points $c_{j+1}, \ldots, c_k$. Additionally, the multiplicity of $(X-d_j)$ in $\hat{f}$ is 1. The polynomial $-(X-d_j)$ is nonnegative over $\mathcal{S}(\hat{f})$, therefore, we compute its certificate using the procedure in Section 3.2.1 in $\mathrm{QM}(\hat{f})$. The polynomial $-(X-c_{j+1})$ is strictly positive over $\mathrm{QM}(\hat{f})$, so we compute its certificate using the Certificate algorithm. Since $\mathrm{QM}(\hat{f})$ is a preordering, we obtain a certificate of $(X-d_j)(X-c_{j+1})$ in $\mathrm{QM}(\hat{f})$ rearranging the certificates of $-(X-d_j)$ and $-(X-c_{j+1})$. Let $(X-d_j)(X-c_{j+1}) = \sigma_0 + \sigma_1 \cdot \hat{f}$. Multiplying s by the previous equation, we obtain $(X-d_j)^m(X-c_{j+1})^{n+1}\prod_{i=j+2}^{k}(X-c_i)^{\mathrm{ord}_{c_i}(f)} = s\sigma_0 + \sigma_1 \cdot f$. Finally, we compute a certificate of $(X-d_j)^m(X-c_{j+1})^{n+1}$ in $\mathrm{QM}(f)$ using the procedure in Section 3.2.1.

If I is not empty, let $l \in I$ be the smallest one. In this case, we define $s := (X-d_j)^{m-1}\prod_{i=j+1}^{l-1}(X-c_i)^{\mathrm{ord}_{c_i}(f)}(X-c_l)^{\mathrm{ord}_{c_l}(f)-1}$. We compute a certificate of $(X-d_j)(X-c_l)$ in $\mathrm{QM}(\hat{f})$ using the procedure in Section 3.2.1. Then, using Lemma 2.21, we compute a certificate of $(X-d_j)(X-c_{j+1})$ in $\mathrm{QM}((X-d_j)(X-c_l))$, lifting the certificate to $\mathrm{QM}(\hat{f})$ as we have a certificate of $(X-d_j)(X-c_l)$ in $\mathrm{QM}(\hat{f})$. We repeat the same procedure of multiplying s from the case when I is empty to compute a certificate of $(X-d_j)^m(X-c_{j+1})^{n+1}$ in $\mathrm{QM}(f)$. □

A.3 Algorithm to split nonnegative factors of input polynomial f

This appendix introduces the Algorithm 21 that provides a method to compute nonnegative factors of a given polynomial. This method is used in the QMCert algorithm in order to split the input into several polynomials that are members of the quadratic module, and their certificates are easier to obtain with our proposed methods.

Algorithm 21: Algorithm to compute nonnegative factors

NonNeg(f)

Input: $f \in \mathbb{A}[X]$
Output: A set $\{f_1, \ldots, f_l\}$ of nonnegative factors of f
Ensures: Each f_i divides f and f_i is nonnegative over $\mathcal{S}(f)$
Let $c_1 < \cdots < c_m$ be the real roots of f and $e_1, \ldots, e_m$ their multiplicities
Let odd_roots, neg_even_roots be empty arrays
for *i from* 1 *to* m **do**
    if $\epsilon_{c_i}(f) = -1$ *and* e_i *is even* **then**
        Add (c_i, e_i) to neg_even_roots
    if e_i *is odd* **then**
        Add (c_i, e_i) to odd_roots
$l := |\texttt{odd\_roots}|$
if $l = 0$ **then return** $\{-\prod_{(c,e)\in \textit{neg\_even\_roots}} (X-c)^e\}$

output := $\emptyset$
Let (c_1, e_1) be the first element of odd_roots
if $\epsilon_{c_1}(f) = 1$ **then**
    Add $(X-c_1)^{e_1} \prod_{\substack{(c,d)\in \texttt{neg\_even\_roots} \\ c<c_1}} (X-c)^e$ to output
    Remove (c_1, e_1) from odd_roots
Let (c_l, e_l) be the last element of odd_roots
if $\epsilon_{c_l}(f) = -1$ **then**
    Add $-(X-c_l)^{e_l} \prod_{\substack{(c,d)\in \texttt{neg\_even\_roots} \\ c>c_l}} (X-c)^e$ to output
    Remove (c_l, e_l) from odd_roots
while *odd_roots is non-empty* **do**
    Let $(c_i, e_i), (c_j, e_j)$ be the first two elements of odd_roots
    Add $(X-c_i)^{e_i}(X-c_j)^{e_j} \prod_{\substack{(c,d)\in \texttt{neg\_even\_roots} \\ c_i<c<c_j}} (X-c)^e$ to output
    Remove $(c_i, e_i), (c_j, e_j)$ from odd_roots
return *output*

A.3.1 Examples

Consider $f_1 := -(X+3)(X+2)$. The real roots of f_1 are -3 and -2 with odd multiplicities without roots of even multiplicities. Since $\epsilon_{-3}(f_1) = 1$, the Algorithm 21 adds $X+3$ to `output`. Since $\epsilon_{-2}(f_1) = -1$, the algorithm adds $-(X+2)$ to `output`.

Consider $f_2 := (X+1)(X+\frac{3}{4})(X+\frac{1}{4})X^2(X-1)$. We have `odd_roots` $= \{(-1,1),(-\frac{3}{4},1),(-\frac{1}{4},1),(1,1)\}$ and `neg_even_roots` $= \{(0,2)\}$. The Algorithm 21 outputs $(X+1)(X+\frac{3}{4})$ and $(X+\frac{1}{4})X^2(X-1)$.

Appendix B

Certificate Computation of Product of Members in the Intermediate Set of Generators $H_{f,G}$

Given two natural generators n_i, n_j of a saturated quadratic module of Nat(G), a certificate of the product $n_i n_j$ is of the form $n_i n_j = \sigma_0 + \sigma_1 \cdot n_i + \sigma_2 \cdot n_j + \sigma_3 \cdot (X - a) + \sigma_4 \cdot (-(X - b))$ where $X - a, -(X - b)$ are strictly positive polynomials over $\mathcal{S}(G)$. In this section, we generalize the previous result by describing a method to compute a certificate of the product of members in the intermediate set of generators $H_{f,G}$ in our approach.

First, we focus on the products of members in $H_{f,G}$ without monomials of even multiplicity. Then, we address the problem of the remaining cases as these can be reduced to the former case. For the rest of this section, we use the polynomials h_i, h_j to denote the two members of $H_{f,G}$ to be considered in the certificate computation

of their product.

B.1 Product between $(X-a_0)^{\kappa_0^+}$ and $(X-b_i)^{\kappa_i^-}(X-a_{i+1})^{\kappa_{i+1}^+}$

Let $h_i = (X-a_0)^{\kappa_0^+}$ and $h_j = (X-b_i)^{\kappa_i^-}(X-a_{i+1})^{\kappa_{i+1}^+}$. If $a_0 = b_i$, from [41][Theorem 2.5], we have $(X-a_0)^2(X-a_{i+1}) = (X-a_{i+1})^2\cdot(X-a_0)+(a_{i+1}-a_0)\cdot(X-a_0)(X-a_{i+1})$. Multiplying both sides by the previous equation by $(X-a_0)^{\kappa_0^+ + \kappa_i^- -2}(X-a_{i+1})^{\kappa_{i+1}^+ -1}$, we obtain

$$\begin{aligned} h_i h_j &= (X-a_0)^{\kappa_0^+ + \kappa_i^- -2}(X-a_{i+1})^{\kappa_{i+1}^+ -1}((X-a_{i+1})^2\cdot(X-a_0) \\ &\quad + (a_{i+1}-a_0)\cdot(X-a_0)(X-a_{i+1})) \\ &= s_0\cdot h_i + s_1\cdot h_j \end{aligned}$$

where $s_0 = (X-a_0)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^+ +1}$ and $s_1 = (a_{i+1}-a_0)(X-a_0)^{\kappa_0^+ -1}$ are both sums of squares.

If $a_0 \neq b_i$, by our previous result in the saturated case, we have $(X-a_0)(X-b_i)(X-a_{i+1}) = s_0 + s_1\cdot(X-a_0) + s_2\cdot(X-b_i)(X-a_{i+1}) + s_3\cdot(-(X-b))$ where $-(X-b)$ is a strictly positive polynomial over $\mathcal{S}(H_{f,G})$. Multiplying both sides of the previous equation by $(X-a_0)^{\kappa_0^+ -1}(X-b_i)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^+ -1}$, we obtain the following.

$$\begin{aligned} h_i h_j &= (X-a_0)^{\kappa_0^+ -1}(X-b_i)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^+ -1} s_0 \\ &+ (X-b_i)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^+ -1} s_1 \cdot h_i \\ &+ (X-a_0)^{\kappa_0^+ -1} s_2 \cdot h_j \\ &+ (X-a_0)^{\kappa_0^+ -1}(X-b_i)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^+ -1} s_3 \cdot (-(X-b)) \end{aligned}$$

B.2 Product between $(X-a_0)^{\kappa_0^+}$ and $-(X-b_k)^{\kappa_k^-}$

Let $h_i = (X-a_0)^{\kappa_0^+}$ and $h_j = -(X-b_k)^{\kappa_k^-}$. If $a_0 = b_k$, by our previous result in the saturated case, we have $-(X-a_0)^2 = (\frac{X-a_0-1}{2})^2 \cdot (X-a_0) + (\frac{X-a_0+1}{2})^2 \cdot (-(X-a_0))$. Multiplying both sides by $(X-a_0)^{\kappa_0^+ + \kappa_k^- -2}$ we obtain the following.

$$\begin{aligned} h_i h_j &= \left(\frac{X-a_0-1}{2}\right)^2 (X-a_0)^{\kappa_k^- -1} \cdot h_i \\ &+ \left(\frac{X-a_0+1}{2}\right)^2 (X-a_0)^{\kappa_0^+ -1} \cdot h_j \end{aligned}$$

If $a_0 \neq b_k$, by our previous result in the saturated case, we have $(X-a_0)(-(X-b_k)) = \frac{1}{b_k-a_0}((X-b_k)^2 \cdot (X-a_0) + (X-a_0)^2 \cdot (-(X-b_k)))$. Multiplying $(X-a_0)^{\kappa_0^+ -1}(X-b_k)^{\kappa_k^- -1}$ to both sides of the previous equation, we obtain:

$$h_i h_j = \frac{1}{b_k - a_0}\left((X-b_k)^{\kappa_k^- +1} \cdot h_i + (X-a_0)^{\kappa_0^+ +1} \cdot h_j\right)$$

B.3 Product between $(X-b_i)^{\kappa_i^-}(X-a_{i+1})^{\kappa_{i+1}^+}$ and $(X-b_j)^{\kappa_j^-}(X-a_{j+1})^{\kappa_{j+1}^+}$

Let $h_i = (X-b_i)^{\kappa_i^-}(X-a_{i+1})^{\kappa_{i+1}^+}$ and $h_j = (X-b_j)^{\kappa_j^-}(X-a_{j+1})^{\kappa_{j+1}^+}$. If $a_{i+1} = b_j$, from [41][Theorem 2.5], we have $(X-b_i)(X-a_{i+1})^2(X-a_{i+2}) = \frac{a_{i+1}-b_i}{a_{i+2}-b_i}(X-a_{i+2})^2 \cdot (X-b_i)(X-a_{i+1}) + \frac{a_{i+2}-a_{i+1}}{a_{i+2}-b_i}(X-b_i)^2 \cdot (X-a_{i+1})(X-a_{i+2})$. Multiplying both sides of the previous equation by $(X-b_i)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^+ + \kappa_{i+2}^- -2}(X-a_{i+2})^{\kappa_{i+2}^+ -1}$, we obtain the following.

$$
\begin{aligned}
h_i h_j = &((X-b_i)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^+ + \kappa_{i+2}^- -2}(X-a_{i+2})^{\kappa_{i+2}^+ -1}) \\
&(\frac{a_{i+1}-b_i}{a_{i+2}-b_i}(X-a_{i+2})^2 \cdot (X-b_i)(X-a_{i+1}) \\
&+ \frac{a_{i+2}-a_{i+1}}{a_{i+2}-b_i}(X-b_i)^2 \cdot (X-a_{i+1})(X-a_{i+2})) \\
&= s_0 \cdot h_i + s_1 \cdot h_j
\end{aligned}
$$

where $s_0 = \frac{a_{i+1}-b_i}{a_{i+2}-b_i}(X-a_{i+1})^{\kappa_{i+2}^- -1}(X-a_{i+2})^{\kappa_{i+2}^+ +1}$ and $s_1 = \frac{a_{i+2}-a_{i+1}}{a_{i+2}-b_i}(X-b_i)^{\kappa_i^- +1}(X-a_{i+1})^{\kappa_{i+1}^+ -1}$ are both sums of squares.

If $a_{i+1} < b_j$, by our previous result in the saturated case, we have

$(X-b_i)(X-a_{i+1})(X-b_j)(X-a_{j+1}) = s_0 + s_1 \cdot (X-a) + s_2 \cdot (X-b_i)(X-a_{i+1}) + s_3 \cdot (X-b_j)(X-a_{j+1}) + s_4 \cdot (-(X-b))$ where $X-a$ and $-(X-b)$ are strictly positive polynomials over $\mathcal{S}(H_{f,G})$. Multiplying both sides by the previous equation by $\sigma := (X-b_i)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^+ -1}(X-b_j)^{\kappa_j^- -1}(X-a_{j+1})^{\kappa_{j+1}^+ -1} \in \sum \mathbb{A}[X]^2$, we obtain

$$\begin{aligned} h_i h_j = \sigma s_0 &+ \sigma s_1 \cdot (X-a) \\ &+ (X-b_j)^{\kappa_j^- -1}(X-a_{j+1})^{\kappa_{j+1}^+ -1} s_2 \cdot h_i \\ &+ (X-b_i)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^+ -1} s_3 \cdot h_j \\ &+ \sigma s_4 \cdot (-(X-b)) \end{aligned}$$

B.4 Product between $(X-b_i)^{\kappa_i^-}(X-d)^{m_1}$ and $(X-c)^{m_2}(X-a_{i+1})^{\kappa_{i+1}^+}$ with $b_i < c < d < a_{i+1}$ and $m_1, m_2 \in 2\mathbb{N}+1$

Let $h_i = (X-b_i)^{\kappa_i^-}(X-d)^{m_1}$, $h_j = (X-c)^{m_2}(X-a_{i+1})^{\kappa_{i+1}^+}$, $\hat{h_i} = (X-b_i)^{\kappa_i^-}(X-c)^{m_2}$, $\hat{h_j} = (X-d)^{m_1}(X-a_{i+1})^{\kappa_{i+1}^+}$, and $H_{\text{fix}} := \{X-a, h_i, h_j, -(X-b)\}$ where $X-a, -(X-b)$ are strictly positive over $\mathcal{S}(H_{f,G})$. Clearly, $h_i h_j = \hat{h_i}\hat{h_j}$. We reduce the problem of computing certificates of $h_i h_j$ in $\text{QM}(H_{\text{fix}})$ to the case addressed in Section B.3 by computing certificates of $h_i h_j$ in $\text{QM}(X-a, \hat{h_i}, \hat{h_j}, -(X-b))$ for some strictly positive polynomials $X-a, -(X-b)$ over $\mathcal{S}(H_{f,G})$. In order to lift the certificates from the new set of generators to H_{fix}, we need to find a certificates of $\hat{h_i}$ and $\hat{h_j}$ in $\text{QM}(H_{\text{fix}})$. We focus on the first case as the method discussed can also handle the second case.

By Lemma 2.21, we compute a certificate of $(X-b_i)(X-c) \in \text{QM}((X-b_i)(X-d))$, i.e, there are computable sums of squares s_0, s_1 such that $(X-b_i)(X-c) = s_0 + s_1 \cdot (X-b_i)(X-d)$. Hence, we can multiply the sums of squares $(X-b_i)^{\kappa_i^- -1}(X-c)^{m_2-1}$ by the previous equation, which gives us a certificate of $\hat{h_i} = (X-b_i)^{\kappa_i^- -1}(X-c)^{m_2-1} s_0 + s_1 \cdot h_i \in \text{QM}(h_i)$. Similarly, we find certificates for $\hat{h_j} \in \text{QM}(h_j)$.

Example B.1 Let $h_1 = X+3, h_2 = (X+2)(X-1), h_3 = (X+1)(X-2), h_4 =$

$-(X-3)$, and $H_{\text{fix}} = \{h_1, h_2, h_3, h_4\}$. We want to compute the certificate of h_2h_3 in $\text{QM}(H_{\text{fix}})$. First, we compute a certificate of $\hat{h_2} = (X+2)(X+1)$ in $\text{QM}(h_2)$. Using Lemma 2.21 we obtain $\hat{h_2} = \frac{2}{3}(X+2)^2 + \frac{1}{3}\cdot h_2$, similarly we have $\hat{h_3} = \frac{2}{3}(X-2)^2 + \frac{1}{3}\cdot h_3$.

Now, we compute a certificate of $\hat{h_2}\hat{h_3}$ in $\text{QM}(h_1, \hat{h_2}, \hat{h_3}, h_4)$ using the method described in Section B.3.

We have $1 = \sigma_1 h_1 \hat{h_2} h_4 + \tau_1 \hat{h_3}$ where:

- $\sigma_{1,1} = \frac{25651419}{1281797120}(-\frac{4415585}{34201892}X^3 - \frac{1251755}{102605676}X^2 + X - \frac{8011232}{76954257})^2$
  $+ \frac{28331277347353}{2311869023304840}(-\frac{29384854743375}{14505614001844736}X^3 - \frac{890819840493969}{7252807000922368}X^2 + 1)^2$
  $+ \frac{1994111861811973593029}{1859325425139625729196032 0}(-\frac{1285483076174859368325}{3988223723623947186058}X^3 + X^2)^2$
  $+ \frac{10676681630420278749526 1701}{3271739957028384938317599 0738944}(X^3)^2$
- $\sigma_{1,2} = \frac{697}{2002808}$
- $\tau_{1,1} = \frac{7224817}{31919480}(-\frac{15161753}{693582432}X^3 - \frac{21619799}{231194144}X^2 + \frac{7181883}{28899268}X + 1)^2$
  $+ \frac{174487837272007}{1844899213881280}(-\frac{302420779830331}{2791805396352112}X^3 + \frac{719486242613227}{8375416189056336}X^2 + X)^2$
  $+ \frac{138684576679824814487417}{1026581489426918153569075 20}(\frac{428504030089969982462 61}{138684576679824814487417}X^3 + X^2)^2$
  $+ \frac{182956542720540303136597309}{3984065614476121112581335 4648440}(X^3)^2$
- $\tau_{1,2} = \frac{2157}{1595974}$
- $\sigma_1 = \sigma_{1,1} + \tau_{1,2}\hat{h_3}^2$
- $\tau_1 = \tau_{1,1} + \tau_{1,2}(h_1\hat{h_2}h_4)^2$

Then, we have $h_1\hat{h_2}\hat{h_3}h_4 = \sigma_1(h_1\hat{h_2}h_4)^2 \cdot \hat{h_3} + \tau_1\hat{h_3}^2 \cdot h_1\hat{h_2}h_4$.

Finally, we have $1 = \sigma_2 h_1 h_4 + \tau_2 \hat{h_2}\hat{h_3}$ where:

- $\sigma_2 = \frac{1}{40}X^2 + \frac{1}{10}$
- $\tau_2 = \frac{1}{40}$

Then, we have

$$\begin{aligned}\hat{h_2}\hat{h_3} &= \tau_2(\hat{h_2}\hat{h_3})^2 + \sigma_2 \cdot h_1\hat{h_2}\hat{h_3}h_4 \\ &= \tau_2(\hat{h_2}\hat{h_3})^2 + \sigma_1\sigma_2(h_1\hat{h_2}h_4)^2 \cdot \hat{h_3} + \sigma_2\tau_1\hat{h_3}^2 \cdot h_1\hat{h_2}h_4\end{aligned}$$

To obtain a certificate in $\mathrm{QM}(h_1, \hat{h_2}, \hat{h_3}, h_4)$, we compute a certificate of $h_1\hat{h_2}h_4$ in $\mathrm{QM}(h_1, \hat{h_2}, h_4)$ using the methods discussed in Section B.2 and Section B.1. First, we find $h_1h_4 = \frac{1}{6}((X-3)^2 \cdot h_1 + (X+3)^2 \cdot h_4)$. Now, we apply the SOSBasicLemma algorithm with inputs h_1h_4 and $\hat{h_2}$ to obtain $1 = \sigma_3 h_1h_4 + \tau_3\hat{h_2}$ where

- $\sigma_3 = \frac{5077893}{74983750}(-\frac{1210241}{3385262}X + 1)^2 + \frac{51983202049}{253839639492500}(X)^2$
- $\tau_3 = \frac{29282713}{149967500}(-\frac{22495125}{117130852}X + 1)^2 + \frac{116498980322719}{70263284189240000}(X)^2$

Hence, $h_1\hat{h_2}h_4 = \sigma_3(h_1h_4)^2 \cdot \hat{h_2} + \tau_3\hat{h_2}^2 \cdot h_1h_4 = \sigma_3(h_1h_4)^2 \cdot \hat{h_2} + \frac{1}{6}\tau_3\hat{h_2}^2(X-3)^2 \cdot h_1 + \frac{1}{6}\tau_3\hat{h_2}^2(X+3)^2 \cdot h_4$, so

$$\begin{aligned}\hat{h_2}\hat{h_3} &= \tau_2(\hat{h_2}\hat{h_3})^2 + \sigma_1\sigma_2(h_1\hat{h_2}h_4)^2 \cdot \hat{h_3} + \sigma_2\tau_1\hat{h_3}^2 \cdot h_1\hat{h_2}h_4 \\ &= \tau_2(\hat{h_2}\hat{h_3})^2 + \sigma_1\sigma_2(h_1\hat{h_2}h_4)^2 \cdot \hat{h_3} + \sigma_2\sigma_3\tau_1(h_1\hat{h_3}h_4)^2 \cdot \hat{h_2} \\ &+ \frac{1}{6}\sigma_2\tau_1\tau_3(\hat{h_2}\hat{h_3})^2(X-3)^2 \cdot h_1 + \frac{1}{6}\sigma_2\tau_1\tau_3(\hat{h_2}\hat{h_3})^2(X+3)^2 \cdot h_4\end{aligned}$$

Finally, we obtain the certificate of $h_2h_3 = \hat{h_2}\hat{h_3}$ in $\mathrm{QM}(h_1, h_2, h_3, h_4)$ by lifting the certificates of $\hat{h_2}$ (resp. $\hat{h_3}$) by h_2 (resp. h_3).

$$
\begin{aligned}
h_2h_3 &= \tau_2(\hat{h_2}\hat{h_3})^2 + \sigma_1\sigma_2(h_1\hat{h_2}h_4)^2 \cdot (\frac{2}{3}(X-2)^2 + \frac{1}{3}\cdot h_3) \\
&\quad + \sigma_2\sigma_3\tau_1(h_1\hat{h_3}h_4)^2 \cdot (\frac{2}{3}(X+2)^2 + \frac{1}{3}\cdot h_2) \\
&\quad + \frac{1}{6}\sigma_2\tau_1\tau_3(\hat{h_2}\hat{h_3})^2(X-3)^2\cdot h_1 + \frac{1}{6}\sigma_2\tau_1\tau_3(\hat{h_2}\hat{h_3})^2(X+3)^2\cdot h_4 \\
&= \tau_2(\hat{h_2}\hat{h_3})^2 + \frac{2}{3}\sigma_1\sigma_2(h_1\hat{h_2}h_4)^2(X-2)^2 + \frac{2}{3}\sigma_2\sigma_3\tau_1(h_1\hat{h_3}h_4)^2(X+2)^2 \\
&\quad + \frac{1}{6}\sigma_2\tau_1\tau_3(\hat{h_2}\hat{h_3})^2(X-3)^2\cdot h_1 + \frac{1}{3}\sigma_2\sigma_3\tau_1(h_1\hat{h_3}h_4)^2\cdot h_2 \\
&\quad + \frac{1}{3}\sigma_1\sigma_2(h_1\hat{h_2}h_4)^2\cdot h_3 + \frac{1}{6}\sigma_2\tau_1\tau_3(\hat{h_2}\hat{h_3})^2(X+3)^2\cdot h_4
\end{aligned}
$$

B.5 Product between $(X-b_i)^{\kappa_i^-}(X-a_{i+2})^{\kappa_{i+2}^+}$ and $(X-a_{i+1})^{\kappa_{i+1}^-}(X-c)^m$ with $b_i < a_{i+1} < c < a_{i+2}$ and $m \in 2\mathbb{N}+1$

Let $h_i = (X-b_i)^{\kappa_i^-}(X-a_{i+2})^{\kappa_{i+2}^+}$, $h_j = (X-a_{i+1})^{\kappa_{i+1}^-}(X-c)^m$, and $H_{\text{fix}} = \{X-a, h_i, h_j, -(X-b)\}$ where $X-a$ and $-(X-b)$ are strictly positive over $\mathcal{S}(H_{f,G})$. Let $\hat{h_i} = (X-b_i)(X-a_{i+2})$ and $\hat{h_j} = (X-a_{i+1})(X-c)$. We use Lemma 2.21 to find a certificate of $\hat{h_j}$ in $\text{QM}(\hat{h_i})$, i.e., we can find sums of squares σ_0, σ_1 such that $\hat{h_j} = \sigma_0 + \sigma_1\cdot\hat{h_i}$. Hence, the product $\hat{h_i}\hat{h_j} = \sigma_1\hat{h_i}^2 + \sigma_0\cdot\hat{h_i}$. Then, we multiply the sums of squares $(X-b_i)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^- -1}(X-c)^{m-1}(X-a_{i+2})^{\kappa_{i+2}^+ -1}$ by the previous equation to obtain the following.

$$
\begin{aligned}
h_ih_j &= \sigma_1\hat{h_i}^2((X-b_i)^{\kappa_i^- -1}(X-a_{i+1})^{\kappa_{i+1}^- -1}(X-c)^{m-1}(X-a_{i+2})^{\kappa_{i+2}^+ -1}) \\
&\quad + \sigma_0(X-a_{i+1})^{\kappa_{i+1}^- -1}\cdot h_i
\end{aligned}
$$

B.6 Product of members in $H_{f,G}$ with even multiplicities

In this section, we extend the cases considered in this appendix so far by including factors of even multiplicities. We consider the problem of computing certificates of the product of the following form $H_i = h_i \prod_{k=1}^{n_1} (X - c_k)^{2m_k}$, $H_j = h_j \prod_{k=1}^{n_2} (X - d_k)^{2m_k}$ where h_i, h_j are the polynomials considered in previous subsections of this appendix and $\epsilon_{c_k}(H_i) = -1$ for $1 \leq k \leq n_1$ (resp. $\epsilon_{d_k}(H_j) = -1$ for $1 \leq k \leq n_2$).

The method is uniform for any of the previous cases. First, we create a subproblem that ignores factors with even multiplicities. To obtain a certificate for the original input; we multiply these factors and rearrange them with the sums of squares multipliers or with the intermediate generators to recover the original generators.

We illustrate the approach by modifying the Example $B.1$:

Example B.2 Let $p_1 = X + 3, p_2 = (X + 2)(X + \frac{1}{2})^2(X - \frac{1}{4})^4(X - 1), p_3 = (X + 1)(X + \frac{1}{2})^4(X + \frac{1}{3})^2(X - 2), p_4 = -(X - 3)$, and $P_{\text{fix}} = \{p_1, p_2, p_3, p_4\}$. We want to compute the certificate of $p_2 p_3$ in $\text{QM}(P_{\text{fix}})$.

First, we ignore the factors with even multiplicities in the inputs. The resulting subproblem is, in fact, Example $B.1$ where $p_1 = h_1, p_2 = (X + \frac{1}{2})^2(X - \frac{1}{4})^4 h_2, p_3 = (X + \frac{1}{2})^4(X + \frac{1}{3})^2 h_3, p_4 = h_4$. Hence, we reuse the certificate of $h_2 h_3 \in \text{QM}(H_{\text{fix}})$ computed in earlier, i.e. $h_2 h_3 = s_0 + s_1 \cdot h_1 + s_2 \cdot h_2 + s_3 \cdot h_3 + s_4 \cdot h_4$ where $s_i' s$ are the sums of squares computed in Example $B.1$. Multiplying the ignored factors with even multiplicities $(X + \frac{1}{2})^6(X + \frac{1}{3})^2(X - \frac{1}{4})^4$ to the previous equation, we obtain:

$$
\begin{aligned}
p_2 p_3 = &\left((X+\frac{1}{2})^3(X+\frac{1}{3})(X-\frac{1}{4})^2\right)^2 s_0 + \left((X+\frac{1}{2})^3(X+\frac{1}{3})(X-\frac{1}{4})^2\right)^2 s_1 \cdot p_1 \\
&+ \left((X+\frac{1}{2})^2(X+\frac{1}{3})\right)^2 s_2 \cdot p_2 + \left((X+\frac{1}{2})(X-\frac{1}{4})^2\right)^2 s_3 \cdot p_3 \\
&+ \left((X+\frac{1}{2})^3(X+\frac{1}{3})(X-\frac{1}{4})^2\right)^2 s_4 \cdot p_4
\end{aligned}
$$

References

[1] Stephen Abbott. *Understanding Analysis*. Springer New York, 2015. ISBN: 9781493927128. DOI: 10.1007/978-1-4939-2712-8. URL: http://dx.doi.org/10.1007/978-1-4939-2712-8.

[2] Emil Artin. "Über die Zerlegung definiter Funktionen in Quadrate". In: *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 5.1 (Dec. 1927), 100–115. ISSN: 1865-8784. DOI: 10.1007/bf02952513. URL: http://dx.doi.org/10.1007/BF02952513.

[3] Doris Augustin. "The Membership Problem for finitely generated quadratic modules in the univariate case". In: *Journal of Pure and Applied Algebra* 216.10 (2012), pp. 2204 –2212. ISSN: 0022-4049. DOI: https://doi.org/10.1016/j.jpaa.2012.02.004. URL: http://www.sciencedirect.com/science/article/pii/S0022404912000436.

[4] Doris Augustin. "The Membership Problem for quadratic modules with focus on the one dimensional case". Ph.D. thesis. PhD thesis. Universität Regensburg, 2008.

[5] Doris Augustin and Manfred Knebusch. "Quadratic modules in R [[X]]". In: *Proceedings of the American Mathematical Society* 138 (Jan. 2010). DOI: 10.1090/S0002-9939-09-10043-6.

[6] Gennadiy Averkov. "Constructive Proofs of Some Positivstellensätze for Compact Semialgebraic Subsets of Źd". In: *J. Optim. Theory Appl.* 158.2 (Aug. 2013), 410–418. ISSN: 0022-3239. DOI: 10.1007/s10957-012-0261-9. URL: https://doi.org/10.1007/s10957-012-0261-9.

[7] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. USA: Cambridge University Press, 1998. ISBN: 0521455200.

[8] Lorenzo Baldi, Teresa Krick, and Bernard Mourrain. "An Effective Positivstellensatz over the Rational Numbers for Finite Semialgebraic Sets". In: *arXiv e-prints* (2024), arXiv–2410.

[9] Lorenzo Baldi, Teresa Krick, and Bernard Mourrain. "An Effective Positivstellensatz over the Rational Numbers for Finite Semialgebraic Sets". In: 2024. URL: https://api.semanticscholar.org/CorpusID:273186358.

[10] Lorenzo Baldi and Bernard Mourrain. *On the Effective Putinar's Positivstellensatz and Moment Approximation*. 2021. DOI: 10.48550/ARXIV.2111.11258. URL: https://arxiv.org/abs/2111.11258.

[11] Lorenzo Baldi and Bernard Mourrain. "On the effective Putinar's Positivstellensatz and moment approximation". In: *Mathematical Programming* 200.1 (Sept. 2022), 71–103. ISSN: 1436-4646. DOI: 10.1007/s10107-022-01877-6. URL: http://dx.doi.org/10.1007/s10107-022-01877-6.

[12] Lorenzo Baldi and Lucas Slot. "Degree Bounds for Putinar's Positivstellensatz on the Hypercube". In: *SIAM Journal on Applied Algebra and Geometry* 8.1 (2024), pp. 1–25. DOI: 10.1137/23M1555430. eprint: https://doi.org/10.1137/23M1555430. URL: https://doi.org/10.1137/23M1555430.

[13] Haniel Barbosa et al. "Flexible Proof Production in an Industrial-Strength SMT Solver". In: *Automated Reasoning*. Ed. by Jasmin Blanchette, Laura Kovács, and Dirk Pattinson. Cham: Springer International Publishing, 2022, pp. 15–35. ISBN: 978-3-031-10769-6.

[14] Clark Barrett, Leonardo de Moura, and Pascal Fontaine. "Proofs in Satisfiability Modulo Theories". In: *All about Proofs, Proofs for All*. Ed. by David Delahaye and Bruno Woltzenlogel Paleo. Vol. 55. Mathematical Logic and Foundations. London, UK: College Publications, Jan. 2015, pp. 23–44. ISBN: 978-1-84890-166-7. URL: http://theory.stanford.edu/~barrett/pubs/BdMF15.pdf.

[15] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 3540330984.

[16] T. Becker et al. *Gröbner Bases: A Computational Approach to Commutative Algebra*. Graduate texts in mathematics. Springer-Verlag, 1993. ISBN: 9780387979717. URL: https://books.google.com/books?id=mgjvAAAAMAAJ.

[17] Grigoriy Blekherman et al. *Semidefinite Optimization and Convex Algebraic Geometry*. USA: Society for Industrial and Applied Mathematics, 2012. ISBN: 1611972280.

[18] Jacek Bochnak, Michel Coste, and Marie-Francoise Roy. *Real Algebraic Geometry*. Springer Berlin Heidelberg, 1998. DOI: 10.1007/978-3-662-03718-8. URL: https://doi.org/10.1007/978-3-662-03718-8.

[19] Brian Borchers. "CSDP, A C library for semidefinite programming". In: *Optimization Methods and Software* 11.1-4 (1999), pp. 613–623. DOI: 10.1080/10556789908805765. eprint: https://doi.org/10.1080/10556789908805765. URL: https://doi.org/10.1080/10556789908805765.

[20] Bruno Buchberger. "Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal". In: *Journal of Symbolic Computation* 41.3 (2006). Logic, Mathematics and Computer Science: Interactions in honor of Bruno Buchberger (60th birthday), pp. 475–511. ISSN: 0747-7171. DOI: https://doi.org/10.1016/

j.jsc.2005.09.007. URL: https://www.sciencedirect.com/science/article/pii/S0747717105001483.

[21] Maria Eugênia Canto Cabral and Alexander Prestel. "Quadratic modules of polynomials in two variables". In: *Advances in Geometry* 8.2 (2008), pp. 189–204. DOI: doi:10.1515/ADVGEOM.2008.014. URL: https://doi.org/10.1515/ADVGEOM.2008.014.

[22] Changbo Chen et al. "Computing with Semi-Algebraic Sets Represented by Triangular Decomposition". In: *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*. ISSAC '11. New York, NY, USA: Association for Computing Machinery, 2011, 75–82. ISBN: 9781450306751. DOI: 10.1145/1993886.1993903. URL: https://doi.org/10.1145/1993886.1993903.

[23] Changbo Chen et al. "Triangular Decomposition of Semi-Algebraic Systems". In: *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*. ISSAC '10. New York, NY, USA: Association for Computing Machinery, 2010, 187–194. ISBN: 9781450301503. DOI: 10.1145/1837934.1837972. URL: https://doi.org/10.1145/1837934.1837972.

[24] S. Chevillard et al. "Efficient and accurate computation of upper bounds of approximation errors". In: *Theoretical Computer Science* 412.16 (Apr. 2011), 1523–1543. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2010.11.052. URL: http://dx.doi.org/10.1016/j.tcs.2010.11.052.

[25] Diego Cifuentes et al. "Sums of squares in *Macaulay2*". In: *The Journal of Software for Algebra and Geometry* 10 (2020).

[26] Diego Cifuentes et al. *SumsOfSquares: A* Macaulay2 *package. Version 2.2*. A *Macaulay2* package available at https://github.com/Macaulay2/M2/tree/stable/M2/Macaulay2/packages.

[27] George E. Collins. "Quantifier elimination for real closed fields by cylindrical algebraic decompostion". In: *Automata Theory and Formal Languages*. Ed. by H. Brakhage. Berlin, Heidelberg: Springer Berlin Heidelberg, 1975, pp. 134–183. ISBN: 978-3-540-37923-2.

[28] James Harold Davenport. "Proving an Execution of an Algorithm Correct?" In: *Intelligent Computer Mathematics*. Ed. by Catherine Dubois and Manfred Kerber. Cham: Springer Nature Switzerland, 2023, pp. 255–269. ISBN: 978-3-031-42753-4.

[29] R Fateman and W Kahan. "Improving exact integrals from symbolic algebra systems". In: *Ctr. for Pure and Appl. Math. Report* 386 (2000).

[30] Benjamin Grégoire, Loïc Pottier, and Laurent Théry. "Proof certificates for algebra and their application to automatic geometry theorem proving". In: *Automated Deduction in Geometry: 7th International Workshop, ADG 2008, Shanghai, China, September 22-24, 2008. Revised Papers 7*. Springer. 2011, pp. 42–59.

[31] Didier Henrion, Simone Naldi, and Mohab Safey El Din. "Exact algorithms for semidefinite programs with degenerate feasible set". In: *Journal of Symbolic Computation* 104 (2021), pp. 942–959. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2020.11.001. URL: https://www.sciencedirect.com/science/article/pii/S0747717120301176.

[32] Marijn J. H. Heule. "The DRAT format and DRAT-trim checker". In: *CoRR* abs/1610.06229 (2016). arXiv: 1610.06229. URL: http://arxiv.org/abs/1610.06229.

[33] Noriyuki Horigome, Akira Terui, and Masahiko Mikawa. "A Design and an Implementation of an Inverse Kinematics Computation in Robotics Using Gröbner Bases". In: *Mathematical Software – ICMS 2020*. Springer Interna-

tional Publishing, 2020, 3–13. ISBN: 9783030522001. DOI: 10.1007/978-3-030-52200-1_1. URL: http://dx.doi.org/10.1007/978-3-030-52200-1_1.

[34] Thomas Jacobi. "Über die Darstellung positiver Polynome auf semialgebraischen Kompakta". PhD thesis. Konstanz: Universität Konstanz, 1999.

[35] Thomas Jacobi and Alexander Prestel. "Distinguished representations of strictly positive polynomials". In: *Crelle's Journal* 2001 (2001).

[36] Erich Kaltofen et al. "Exact Certification of Global Optimality of Approximate Factorizations via Rationalizing Sums-of-Squares with Floating Point Scalars". In: *Proceedings of the Twenty-First International Symposium on Symbolic and Algebraic Computation*. ISSAC '08. New York, NY, USA: Association for Computing Machinery, 2008, 155–164. ISBN: 9781595939043. DOI: 10.1145/1390768.1390792. URL: https://doi.org/10.1145/1390768.1390792.

[37] Deepak Kapur. "Using Gröbner bases to reason about geometry problems". In: *Journal of Symbolic Computation* 2.4 (1986), pp. 399–408. ISSN: 0747-7171. DOI: https://doi.org/10.1016/S0747-7171(86)80007-4. URL: https://www.sciencedirect.com/science/article/pii/S0747717186800074.

[38] Deepak Kapur et al. "The Generalized Rabinowitsch Trick". In: *Applications of Computer Algebra*. Springer International Publishing, 2017, 219–229. ISBN: 9783319569321. DOI: 10.1007/978-3-319-56932-1_14. URL: http://dx.doi.org/10.1007/978-3-319-56932-1_14.

[39] Guy Katz et al. "Reluplex: An efficient SMT solver for verifying deep neural networks". In: *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*. Springer. 2017, pp. 97–117.

[40] Jean-Louis Krivine. "Anneaux préordonnés". In: *Journal d'analyse mathématique* 12 (1964), p. 307–326. URL: https://hal.science/hal-00165658.

[41] S. Kuhlmann and M. Marshall. "Positivity, sums of squares and the multi-dimensional moment problem". In: 354.11 (July 2002), pp. 4285–4301. DOI: 10.1090/s0002-9947-02-03075-1. URL: https://doi.org/10.1090/s0002-9947-02-03075-1.

[42] S. Kuhlmann, M. Marshall, and N. Schwartz. "Positivity, sums of squares and the multi-dimensional moment problem II". In: 5.4 (Sept. 2005), pp. 583–606. DOI: 10.1515/advg.2005.5.4.583. URL: https://doi.org/10.1515/advg.2005.5.4.583.

[43] A. H. Lachlan and E. W. Madison. "Computable Fields and Arithmetically Definable Ordered Fields". In: *Proceedings of the American Mathematical Society* 24.4 (1970), pp. 803–807. ISSN: 00029939, 10886826. URL: http://www.jstor.org/stable/2037328 (visited on 05/06/2025).

[44] Jean B. Lasserre. "A Sum of Squares Approximation of Nonnegative Polynomials". In: *SIAM Review* 49.4 (Jan. 2007), 651–669. ISSN: 1095-7200. DOI: 10.1137/070693709. URL: http://dx.doi.org/10.1137/070693709.

[45] Jean B. Lasserre. "Global Optimization with Polynomials and the Problem of Moments". In: *SIAM Journal on Optimization* 11.3 (Jan. 2001), 796–817. ISSN: 1095-7189. DOI: 10.1137/s1052623400366802. URL: http://dx.doi.org/10.1137/S1052623400366802.

[46] Monique Laurent and Lucas Slot. *An effective version of Schmüdgen's Positivstellensatz for the hypercube*. 2021. DOI: 10.48550/ARXIV.2109.09528. URL: https://arxiv.org/abs/2109.09528.

[47] François Lemaire et al. "Solving semi-algebraic systems with the RegularChains library in Maple". In: *Proceedings of the Fourth International Conference on Mathematical Aspects of Computer Science and Information Sciences (MACIS2011)*. Ed. by Stefan Raschau. Beijing, China, 2011, pp. 38–51. URL: https://hal.science/hal-00825013.

[48] Henri Lombardi. "Une borne sur les degres pour le theoreme des zeros reel effectif". In: *Real Algebraic Geometry*. Springer Berlin Heidelberg, 1992, 323–345. ISBN: 9783540559924. DOI: 10.1007/bfb0084631. URL: http://dx.doi.org/10.1007/BFb0084631.

[49] Henrri Lombardi, Daniel Perrucci, and Marie-Françoise Roy. "An Elementary Recursive Bound for Effective Positivstellensatz and Hilbert's 17th problem". In: *Memoirs of the American Mathematical Society* 263 (Apr. 2014). DOI: 10.1090/memo/1277.

[50] Victor Magron and Mohab Safey El Din. *RealCertify: a Maple package for certifying non-negativity*. 2018. DOI: 10.48550/ARXIV.1805.02201. URL: https://arxiv.org/abs/1805.02201.

[51] Victor Magron and Mohab Safey El Din. "Realcertify: a maple package for certifying non-negativity". In: *ACM Commun. Comput. Algebra* 52.2 (Oct. 2018), 34–37. ISSN: 1932-2232. DOI: 10.1145/3282678.3282681. URL: https://doi.org/10.1145/3282678.3282681.

[52] Victor Magron and Mohab Safey El Din. "On Exact Polya and Putinar's Representations". In: *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation* (July 2018). DOI: 10.1145/3208976.3208986. URL: http://dx.doi.org/10.1145/3208976.3208986.

[53] Victor Magron and Mohab Safey El Din. "On exact Reznick, Hilbert-Artin and Putinar's representations". In: *Journal of Symbolic Computation* 107 (2021), pp. 221–250. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2021.03.005. URL: https://www.sciencedirect.com/science/article/pii/S0747717121000249.

[54] Victor Magron, Mohab Safey El Din, and Markus Schweighofer. "Algorithms for weighted sum of squares decomposition of non-negative univariate polynomials". In: *Journal of Symbolic Computation* 93 (July 2019), 200–220. ISSN:

0747-7171. DOI: 10.1016/j.jsc.2018.06.005. URL: http://dx.doi.org/10.1016/j.jsc.2018.06.005.

[55] Victor Magron et al. "Exact SOHS Decompositions of Trigonometric Univariate Polynomials with Gaussian Coefficients". In: *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*. ISSAC '22. New York, NY, USA: Association for Computing Machinery, 2022, 325–332. ISBN: 9781450386883. DOI: 10.1145/3476446.3535480. URL: https://doi.org/10.1145/3476446.3535480.

[56] Ngoc Hoang Anh Mai, Victor Magron, and Jean-Bernard Lasserre. *A sparse version of Reznick's Positivstellensatz*. 2020. DOI: 10.48550/ARXIV.2002.05101. URL: https://arxiv.org/abs/2002.05101.

[57] M. Marshall. *Positive Polynomials and Sums of Squares*. Mathematical surveys and monographs. American Mathematical Society, 2008. ISBN: 9780821844021. URL: https://books.google.com/books?id=HW_0BwAAQBAJ.

[58] Murray Marshall. *Positive polynomials and sums of squares*. Feb. 2008. ISBN: 978-0-8218-4402-1; 0-8218-4402-4. DOI: 10.1090/surv/146/02.

[59] Oliver Marx et al. "Proof logging for computer algebra based SMT solving". In: *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 2013, pp. 677–684. DOI: 10.1109/ICCAD.2013.6691188.

[60] Kenneth McMillan. "Interpolants from Z3 proofs". In: *Formal Methods in Computer-Aided Design*. Oct. 2011. URL: https://www.microsoft.com/en-us/research/publication/interpolants-from-z3-proofs/.

[61] Bhubaneswar Mishra and Franz Winkler. "Algorithmic algebra". In: *SIAM Review* 37.3 (1995), p. 479.

[62] Leonardo Mendonça de Moura and Nikolaj S Bjørner. "Proofs and Refutations, and Z3." In: *LPAR Workshops*. Vol. 418. Doha, Qatar. 2008, pp. 123–132.

[63] Jiawang Nie and Markus Schweighofer. "On the Complexity of Putinar's Positivstellensatz". In: *J. Complex.* 23.1 (Feb. 2007), 135–150. ISSN: 0885-064X. DOI: 10.1016/j.jco.2006.07.002. URL: https://doi.org/10.1016/j.jco.2006.07.002.

[64] Robert Nieuwenhuis and Albert Oliveras. "Proof-Producing Congruence Closure". In: *Term Rewriting and Applications*. Ed. by Jürgen Giesl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 453–468. ISBN: 978-3-540-32033-3.

[65] A. Papachristodoulou et al. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. A *MATLAB* package available at https://github.com/oxfordcontrol/SOSTOOLS.

[66] A. Papachristodoulou et al. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. Available from https://github.com/oxfordcontrol/SOSTOOLS. http://arxiv.org/abs/1310.4716, 2021.

[67] Pablo A Parrilo. "Semidefinite programming relaxations for semialgebraic problems". In: *Mathematical programming* 96 (2003), pp. 293–320.

[68] Helfried Peyrl and Pablo A. Parrilo. "Computing sum of squares decompositions with rational coefficients". In: *Theoretical Computer Science* 409.2 (2008). Symbolic-Numerical Computations, pp. 269–281. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2008.09.025. URL: https://www.sciencedirect.com/science/article/pii/S0304397508006452.

[69] Victoria Powers. "Positive polynomials and sums of squares: Theory and practice". In: *Real Algebraic Geometry* 1 (2011), pp. 78–149.

[70] Victoria Powers and Bruce Reznick. "A new bound for Pólya's Theorem with applications to polynomials positive on polyhedra". In: *Journal of Pure and Applied Algebra* 164.1–2 (Oct. 2001), 221–229. ISSN: 0022-4049. DOI: 10.1016/

s0022-4049(00)00155-9. URL: http://dx.doi.org/10.1016/S0022-4049(00)00155-9.

[71] Victoria Powers and Thorsten Wörmann. "An algorithm for sums of squares of real polynomials". In: 127.1 (May 1998), pp. 99–104. DOI: 10.1016/s0022-4049(97)83827-3. URL: https://doi.org/10.1016/s0022-4049(97)83827-3.

[72] Mihai Putinar. "Positive Polynomials on Compact Semi-algebraic Sets". In: *Indiana University Mathematics Journal* 42.3 (1993), pp. 969–984. ISSN: 00222518, 19435258. URL: http://www.jstor.org/stable/24897130.

[73] J. L. Rabinowitsch. "Zum Hilbertschen Nullstellensatz". In: *Mathematische Annalen* 102.1 (Dec. 1930), 520–520. ISSN: 1432-1807. DOI: 10.1007/bf01782361. URL: http://dx.doi.org/10.1007/BF01782361.

[74] Pierre Roux, Yuen-Lam Voronin, and Sriram Sankaranarayanan. "Validating numerical semidefinite programming solvers for polynomial invariants". In: *Formal Methods in System Design* 53.2 (Oct. 2017), 286–312. ISSN: 1572-8102. DOI: 10.1007/s10703-017-0302-y. URL: http://dx.doi.org/10.1007/s10703-017-0302-y.

[75] Massimiliano Sala. "Gröbner Bases, Coding, and Cryptography: a Guide to the State-of-Art". In: *Gröbner Bases, Coding, and Cryptography*. Springer Berlin Heidelberg, 2009, 1–8. ISBN: 9783540938064. DOI: 10.1007/978-3-540-93806-4_1. URL: http://dx.doi.org/10.1007/978-3-540-93806-4_1.

[76] C. Scheiderer. "Sums of squares of regular functions on real algebraic varieties". In: *Transactions of the American Mathematical Society* 352 (Mar. 2000). DOI: 10.2307/118100.

[77] Claus Scheiderer. "Sums of squares on real algebraic curves". In: *Mathematische Zeitschrift* 245.4 (Dec. 2003), 725–760. ISSN: 1432-1823. DOI: 10.1007/s00209-003-0568-1. URL: http://dx.doi.org/10.1007/s00209-003-0568-1.

[78] Konrad Schmüdgen. "The K-moment problem for compact semi-algebraic sets". In: *Mathematische Annalen* 289.1 (Mar. 1991), 203–206. ISSN: 1432-1807. DOI: 10.1007/bf01446568. URL: http://dx.doi.org/10.1007/BF01446568.

[79] Markus Schweighofer. "An algorithmic approach to Schmüdgen's Positivstellensatz". In: *Journal of Pure and Applied Algebra* 166.3 (2002), pp. 307–319. ISSN: 0022-4049. DOI: https://doi.org/10.1016/S0022-4049(01)00041-X. URL: https://www.sciencedirect.com/science/article/pii/S002240490100041X.

[80] Markus Schweighofer. "On the complexity of Schmüdgen's Positivstellensatz". In: *Journal of Complexity* 20.4 (2004), pp. 529–543. ISSN: 0885-064X. DOI: https://doi.org/10.1016/j.jco.2004.01.005. URL: https://www.sciencedirect.com/science/article/pii/S0885064X04000093.

[81] Weifeng Shang, Chenqi Mou, and Deepak Kapur. "Algorithms for Testing Membership in Univariate Quadratic Modules over the Reals". In: *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*. ISSAC '22. New York, NY, USA: Association for Computing Machinery, 2022, 429–437. ISBN: 9781450386883. DOI: 10.1145/3476446.3536176. URL: https://doi.org/10.1145/3476446.3536176.

[82] Weifeng Shang et al. *Computing Certificates of Strictly Positive Polynomials in Archimedean Quadratic Modules*. 2025. arXiv: 2503.11119 [math.AC]. URL: https://arxiv.org/abs/2503.11119.

[83] Gagandeep Singh et al. "An abstract domain for certifying neural networks". In: *Proceedings of the ACM on Programming Languages* 3.POPL (2019), pp. 1–30.

[84] Gagandeep Singh et al. "Fast and effective robustness certification". In: *Advances in neural information processing systems* 31 (2018).

[85] Gilbert Stengle. "A nullstellensatz and a positivstellensatz in semialgebraic geometry". In: *Mathematische Annalen* 207.2 (June 1974), 87–97. ISSN: 1432-1807. DOI: 10.1007/bf01362149. URL: http://dx.doi.org/10.1007/BF01362149.

[86] Gilbert Stengle. "Complexity Estimates for the Schmüdgen Positivstellensatz". In: *Journal of Complexity* 12.2 (June 1996), 167–174. ISSN: 0885-064X. DOI: 10.1006/jcom.1996.0011. URL: http://dx.doi.org/10.1006/jcom.1996.0011.

[87] Jos F. Sturm. "Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones". In: *Optimization Methods and Software* 11.1-4 (1999), pp. 625–653. DOI: 10.1080/10556789908805766. eprint: https://doi.org/10.1080/10556789908805766. URL: https://doi.org/10.1080/10556789908805766.

[88] K. C. Toh, M. J. Todd, and R. H. Tütüncü. "SDPT3 — A Matlab software package for semidefinite programming, Version 1.3". In: *Optimization Methods and Software* 11.1-4 (1999), pp. 545–581. DOI: 10.1080/10556789908805762. eprint: https://doi.org/10.1080/10556789908805762. URL: https://doi.org/10.1080/10556789908805762.

[89] Sven Wagner. "Archimedean Quadratic Modules : A Decision Problem for Real Multivariate Polynomials". PhD thesis. Konstanz: Universität Konstanz, 2009.

[90] Nathan Wetzler, Marijn J. H. Heule, and Warren A. Hunt. "DRAT-trim: Efficient Checking and Trimming Using Expressive Clausal Proofs". In: *Theory and Applications of Satisfiability Testing – SAT 2014*. Ed. by Carsten Sinz and Uwe Egly. Cham: Springer International Publishing, 2014, pp. 422–429. ISBN: 978-3-319-09284-3.

[91] Makoto Yamashita, Katsuki Fujisawa, and Masakazu Kojima. "Implementation and evaluation of SDPA 6.0 (Semidefinite Programming Algorithm 6.0)". In: *Optimization Methods and Software* 18.4 (2003), pp. 491–505. DOI: 10.1080/1055678031000118482. eprint: https://doi.org/10.1080/1055678031000118482. URL: https://doi.org/10.1080/1055678031000118482.