Complexity of Real Root Isolation Using Continued Fractions

Vikram Sharma
GALAAD, INRIA
Sophia Antipolis, France
vikram.sharma@sophia.inria.fr

January 25, 2007

Abstract

Akritas had proposed an algorithm, which utilizes the continued fraction expansion of real algebraic numbers, for isolating the real roots of a univariate polynomial. The efficiency of the algorithm depends upon computing tight lower bounds on the smallest positive root of a polynomial. The known complexity bounds for the algorithm rely on the impractical assumption that it is possible to efficiently compute a lower bound that is at a constant distance from the smallest positive root; without this assumption, the worst case bounds are exponential. In this paper, we derive a polynomial worst case bound on the algorithm without relying on the above mentioned assumption. In particular, we show that for a square-free integer polynomial of degree n and coefficients of bit-length L, the bit-complexity of the algorithm is $\widetilde{O}(n^8L^3)$, where $\widetilde{O}$ indicates that we are omitting logarithmic factors.

1 Introduction

A fundamental task in computer algebra is **real root isolation**, i.e., given a polynomial $A(X)$ with real numbers as coefficients, compute disjoint intervals with rational endpoints that contain exactly one real root of $A(X)$, and together contain all the real roots of $A(X)$. In this paper, we assume $A(X)$ is square free and has degree n.

Based upon the **Descartes' rule of signs** (see Proposition 2.1 below), Vincent [Vin36] proposed an algorithm for real root isolation that computes continued fraction expansion of the real roots of the polynomial; see [AG98] for detailed historic information of the algorithm. Given a polynomial $A(X)$, the algorithm constructs a series of polynomials $A_i(X)$ such that $A_i(X) := A_{i-1}(X+1)$, $A_0(X) := A(X)$, until there is a root of $A_i(X)$ in the unit interval; from $A_i(X)$ it then constructs a polynomial whose positive roots correspond with the roots of $A_i(X)$ in the unit interval and another polynomial whose positive roots correspond with the roots of $A_i(X)$ greater than one; the algorithm then recursively proceeds on these two polynomials; it stops when the polynomial at a recursive call has zero or one sign variation

(i.e., a change from positive to negative or vice versa) in its coefficients, and in the latter case it outputs an isolating interval for the input polynomial $A(X)$. Observe that the number of Taylor shifts needed to compute $A_i(X)$ from $A(X)$ is equal to the floor of the smallest positive root of $A(X)$. Thus Vincent's algorithm has an exponential worst case running time.

Two algorithms were proposed to overcome this drawback: the Descartes method by Collins and Akritas [CA76] [1], and a modification of Vincent's algorithm by Akritas [Akr78b]. The latter algorithm is based upon the idea that to obtain $A_i(X)$ from $A_{i-1}(X)$ we should compute a lower bound b on the smallest positive root of $A_{i-1}(X)$, and if $b > 1$ then perform a Taylor shift on $A_{i-1}(X)$ by b instead of by one, as done by Vincent. The algorithm by Collins and Akritas, however, is substantially different from Vincent's: their algorithm starts with an interval containing all the real roots of the input polynomial; it then sub-divides this interval into two equal parts and recursively searches for roots in each of these halves, using Descartes' rule of signs as a stopping criterion.

In practice, Akritas' algorithm performs comparably with the Descartes method [ET06, AS05]. But what is more interesting about the algorithm is that, unlike the Descartes method, it utilizes the distribution of the real roots of the polynomial for isolating them; the advantage of this approach is evident [ET06, Tab. 1] when isolating the real roots of Mignotte's polynomials [Mig81], where it is known [ESY06, Thm. 3.6] that the subdivision approach of the Descartes method is not suitable. Another advantage of the algorithm is that the approximations computed to the real roots are their continued fraction expansion, which is the best one can expect for a given bit-size of the fraction (e.g., see [Yap00, p. 469]). Moreover, combined with Lagrange's method [AG98, Sec. 3], we easily obtain an algorithm for *real root approximation*, i.e., given a polynomial, approximate its real roots to any desired accuracy.

Despite these advantages of Akritas' algorithm, there is a huge gap between the two algorithms when it comes to understanding their worst case complexity. For the Descartes method we know [Joh98, Kra95, ESY06, EMT06] that for an integer polynomial of degree n with L-bit coefficients its worst case bit-complexity is $\widetilde{O}(n^4L^2)$, here $\widetilde{O}$ means we omit logarithmic factors. On the other hand, Akritas has claimed an $\widetilde{O}(n^5L^3)$ bound on the worst case bit-complexity of his algorithm, but his analysis has two drawbacks: first, he assumes the **ideal Positive Lower Bound (**PLB**) function**, i.e., a function that can determine whether a polynomial has positive real roots, and if there are such roots then returns a value that is at a constant distance from the smallest positive root of the polynomial; and second, as mentioned in [ET06], his analysis does not account for the increased coefficient size of $A_i(X)$ after performing Taylor shift on $A_{i-1}(X)$. The latter drawback was partially resolved by Emiris and Tsigaridas [ET06], who derived an $\widetilde{O}(n^4L^2)$ bound on the expected running time of the algorithm, by using bounds by Khinchin [Khi97] on the expected bit-size of the partial quotients appearing in the continued fraction expansion of a real number; however, for bounding the size of the recursion tree of the algorithm, their analysis also assumed the ideal PLB function. In practice we never use the ideal PLB function because of its prohibitive cost (intuitively it is almost equivalent to doing real root isolation). Instead we use functions that are based upon upper bounds on the absolute value of the roots of a polynomial, such

[1] It was proposed to overcome the exponential running time of Uspensky's algorithm [Usp48], which is an inefficient version of Vincent's algorithm, see [Akr86].

as those by Cauchy, Zassenhaus etc. (see Section 2 for details). These bounds have the two advantages: they can be computed efficiently, and they provide a good lower bound on the smallest positive root.

Thus the complexity analysis of Akritas' algorithm in the current literature does not correspond with the actual implementation of the algorithm. Moreover, given the similarity between Akritas' and Vincent's algorithm, we may conclude that the former is also exponential in the worst case, though we do not know of any example where such behaviour is demonstrated.

In this paper we bridge the gap between the understanding of Akritas' algorithm in theory and practice, and in doing so provide the first polynomial bound on its worst case complexity. More precisely, for a square-free integer polynomial of degree n and L-bit coefficients, we derive a worst case bound of $\widetilde{O}(n^8L^3)$ on Akritas' algorithm *without assuming the ideal* PLB *function*. But if we make this assumption then we can improve the bound to $\widetilde{O}(n^5L^2)$. These bounds are derived in Section 4 and are based upon a bound on the size of the recursion tree of Akritas' algorithm, which is derived in Section 3 and is applicable to polynomials with real coefficients. The crucial component for bounding the size of the recursion tree of the algorithm, without assuming the ideal PLB function, is the tightness of the lower bounds on the positive real roots of the polynomial; this is the subject that we treat in Section 2, where we also give the details of Akritas' algorithm and criteria for its termination.

2 The Continued Fraction Algorithm

Given a polynomial $A(X) = a_nX^n + a_{n-1}X^{n-1} + \cdots + a_0$, $a_i \in \mathbb{R}$, let $\texttt{Var}(A)$ represent the number of sign changes (positive to negative and vice versa) in the sequence $(a_n, a_{n-1}, \ldots, a_0)$.

One of the crucial components of Akritas' algorithm is the Descartes' rule of signs:

Proposition 2.1. *Let $A(X)$ be a polynomial with real coefficients. Then the number of positive real roots of $A(X)$ counted with multiplicities is smaller than* $\mathrm{Var}(A)$ *by a positive even number.*

See [KM06] for a proof with careful historic references. Since $\mathrm{Var}(A)$ exceeds the number of positive roots by a positive even number, the Descartes' rule of signs yields the exact number of positive roots whenever $\mathrm{Var}(A)$ is 0 or 1.

The other component is a procedure $\mathrm{PLB}(A)$ that takes as input a polynomial $A(X)$ and returns a lower bound on the smallest positive root of $A(X)$.

Given these two components, Akritas' algorithm for isolating the real roots of a square-free input polynomial $A_{\mathrm{in}}(X)$ uses a recursive procedure $\mathrm{CF}(A, M)$ that takes as input a polynomial $A(X)$ and a Möbius transformation $M(X) = \frac{pX+q}{rX+s}$, where $p, q, r, s \in \mathbb{N}$ and $ps - rq \neq 0$. With the transformation $M(X)$ we can associate the interval I_M that has endpoints p/r and q/s. The relation among $A_{\mathrm{in}}(X)$, $A(X)$ and $M(X)$ is the following:

$$A(X) = (rX + s)^n A_{\mathrm{in}}(M(X)). \tag{1}$$

Given this relation, the procedure $\mathrm{CF}(A, M)$ returns a list of isolating intervals for the roots of $A_{\mathrm{in}}(X)$ in I_M. To isolate all the positive roots of $A_{\mathrm{in}}(X)$ initiate $\mathrm{CF}(A, M)$ with

$A(X) = A_{\text{in}}(X)$ and $M(X) = X$; to isolate the negative roots of $A_{\text{in}}(X)$ initiate CF(A, M) on $A(X) := A_{\text{in}}(-X)$ and $M(X) = X$ and flip the signs of the intervals returned.

The procedure CF(A, M) is as follows:

Procedure CF(A, M)
Input: A square-free polynomial $A(X) \in \mathbb{R}[X]$ and a Möbius transformation $M(X)$ satisfying $(*)$.
Output: A list of isolating intervals for the roots of $A_{\text{in}}(X)$ in I_M.

1. **If** $A(0) = 0$ **then**
   Output the interval $[M(0), M(0)]$.
   $A(X) := A(X)/X$; **return** CF(A, M).
2. **If** $\texttt{Var}(A) = 0$ **then return**.
3. **If** $\texttt{Var}(A) = 1$ **then**
   Output the interval I_M and **return**.
4. $b := \text{PLB}(A)$.
5. **If** $b > 1$ **then** $A(X) := A(X + b)$ and $M(X) := M(X + b)$.
6. $A_R(X) := A(1 + X)$ and $M_R(X) := M(1 + X)$.
7. CF(A_R, M_R).
8. **If** $\texttt{Var}(A_R) < \texttt{Var}(A)$ **then**
9. $A_L(X) := (1 + X)^n A\left(\frac{1}{1+X}\right)$ and $M_L(X) := M\left(\frac{1}{1+X}\right)$, $n = \deg(A)$.
10. **If** $A_L(0) = 0$ **then** $A_L(X) := A_L(X)/X$.
11. CF(A_L, M_L).

Some remarks on the procedure:

- By removing lines 4 and 5, we obtain Vincent's algorithm for isolating positive roots.
- Line 8 avoids unnecessary computations of $A_L(X)$ since if $\texttt{Var}(A_R) = \texttt{Var}(A)$ then from Budan's theorem [Akr82] we know that there are no real roots of $A(X)$ between 0 and $b + 1$. This test is missing in Uspensky's formulation of the algorithm [Usp48, p. 128] and was pointed out in [Akr86].
- Line 10 is necessary to avoid recounting; since $A_L(0) = 0$ if and only if $A_R(0) = 0$, the root would have been already reported in the recursive call at line 7.

We now give the details of the positive lower bound function used in the algorithm above.

2.1 Lower Bounds on Roots

Given a polynomial $A(X) = \sum_{i=0}^{n} a_i X^i$, $a_0 \neq 0$, one way to compute PLB(A) is to take the inverse of an upper bound on the absolute values of the roots of the polynomial

$$R(A)(X) := X^n A(1/X)$$

. Thus the problem of computing lower bounds on the absolute value of the roots of a polynomial $A(X)$ is equivalent to deriving upper bounds on the absolute value of the roots of the polynomial $R(A)(X)$, and so we will focus on these latter bounds.

For a polynomial $A(X)$, let $\mu(A)$ denote the largest absolute value over all the roots of a polynomial $A(X)$. Zassenhaus has given the following upper bound on $\mu(A)$:

$$S(A) := 2 \max_{i=1,\ldots,n} \left| \frac{a_{n-i}}{a_n} \right|^{1/i}.$$

Van der Sluis [vdS70] showed that $S(A)$ is at most twice the optimal bound amongst all bounds based solely upon the absolute value of the coefficients; he also showed that

$$\mu(A) < S(A) \leq 2n\mu(A). \tag{2}$$

The proof for the lower bound on $S(A)$ can be found, for instance, in [Yap00, Lem. 6.5, p. 147]. The upper bound on $S(A)$ will follow if we show for all $0 < i \leq n$ that $\left| \frac{a_{n-i}}{a_n} \right|^{1/i} \leq n\mu(A)$. Let $\alpha_1, \ldots, \alpha_n$ be the roots of $A(X)$. Then we know that

$$\left| \frac{a_{n-i}}{a_n} \right| \leq \sum_{1 \leq j_1 < \cdots < j_i \leq n} |\alpha_{j_1} \cdots \alpha_{j_i}| \leq \binom{n}{i} \mu(A)^i.$$

Taking the i-th root on both sides, along with the observation that for $1 \leq i \leq n$, $\binom{n}{i}^{1/i} \leq \left(\frac{n^i}{i!} \right)^{1/i} \leq n$, proves the upper bound in (2).

Clearly, $S(A)$ cannot be computed exactly in general. Instead, we use a procedure $U(A)$, similar to that suggested by Akritas [Akr89, p. 350], which computes an upper bound on $\mu(A)$ when $A(X) \in \mathbb{R}[X]$.

PROCEDURE U(A)
INPUT: An integer polynomial $A(X) = \sum_{i=0}^{n} a_i x^i$, $a_i \in \mathbb{R}$, $a_n > 0$.
OUTPUT: A power of two that is an upper bound on the roots of $A(X)$.
1. $q' := -\infty$.
2. **For** i **from** 1 **to** n do the following:
   $p := \lfloor \log |a_{n-i}| \rfloor - \lfloor \log |a_n| \rfloor - 1$.
   Let $q = \lfloor p/i \rfloor$.
   $q' := \max(q', q+2)$.
3. **Return** $2^{q'}$.

Remark 2.2. *If $A(X)$ is an integer polynomial with coefficients of bit-length L then the cost of computing $U(A)$ is $\widetilde{O}(nL)$, because the most expensive operation in the loop on Line 4 is computing the floor of the coefficients which can be done in $O(L)$ time; since the loop runs n times we have the desired bound.*

We have the following relation between $U(A)$ and $S(A)$:

Lemma 2.3.

$$\frac{U(A)}{4} < S(A) < U(A).$$

Proof. See Appendix for the proof. □

This lemma along with (2) yields us

$$\mu(A) < U(A) < 8n\mu(A). \tag{3}$$

For a polynomial $A(X)$, $A(0) \neq 0$, define

$$\mathrm{PLB}(A) := \frac{1}{U(R(A))}. \tag{4}$$

Then from (3) we know that

$$\frac{1}{8n\mu(R(A))} < \mathrm{PLB}(A) < \frac{1}{\mu(R(A))}.$$

Let $\kappa(A)$ denote the minimum of the absolute values of the roots of $A(X)$; thus $\kappa(A) > 0$. Then we have

$$\frac{\kappa(A)}{8n} < \mathrm{PLB}(A) < \kappa(A), \tag{5}$$

since $\kappa(A) = \frac{1}{\mu(R(A))}$.

In practice one can use bounds (such as [Kio86, Şte05]) that utilize the sign of the coefficients and give a better estimate than $S(A)$. For instance, Kioustelidis [Kio86] has shown that the bound

$$K(A) := 2 \max_{a_i<0} \left| \frac{a_i}{a_n} \right|^{1/(n-i)},$$

where a_i is a coefficient of $A(X)$, is an upper bound on the largest positive root of $A(X)$; by definition we have $K(A) \leq S(A)$. We do not use this bound in our analysis because we do not know a relation corresponding to (2) between $K(A)$ and the largest positive root of $A(X)$. But such a relation seems unlikely to hold. Consider the situation when there is only one negative root of $A(X)$, which has the largest absolute value amongst all the roots of $A(X)$. Then the summation in

$$\left| \frac{a_j}{a_n} \right| = \left| \sum_{1 \leq i_1 < \cdots < i_{n-j} \leq n} \alpha_{i_1} \cdots \alpha_{i_{n-j}} \right|,$$

where $\alpha_1, \ldots, \alpha_n$ are the roots of $A(X)$, will be dominated by the negative root. Thus, it appears, that the best we can say is $K(A) \leq 2n\mu(A)$. The same argument applies to the bound by Ştefănescu [Şte05].

Hong [Hon98] has given bounds on the positive roots of a polynomial in more than one variable. For univariate polynomials, his bound is

$$2 \max_{a_j<0} \min_{a_k>0, k>j} \left| \frac{a_j}{a_k} \right|^{1/(k-j)}.$$

It is clear that this bound is an improvement over the bound by Kioustelidis. However, again it is not obvious whether a relation similar to (2) holds between the largest positive

root and the bound above. This is because Hong's bound is on the absolute positiveness of a polynomial, i.e., a bound such that the evaluation of the polynomial and *all* its derivatives at any point larger than the bound is strictly positive. In case of univariate polynomials this means Hong's bound is an upper bound on the positive roots of the polynomial and its derivatives. The difficulty in obtaining a tight relation suggested above is that the real roots of the derivatives may be greater than the positive roots of the polynomial, as in the polynomial $3X^3 - 15X^2 + 11X - 7 = 3(X-1)(X-2+i/\sqrt{3})(X-2-i/\sqrt{3})$.

Now that we have all the details of the algorithm, we face the question of its termination. To answer this question we need some notation from the theory of continued fractions; our notation is borrowed from [Yap00, Ch. 15].

An **ordinary continued fraction** (also called simple continued fractions, or regular continued fractions) is of the form

$$q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cdots}}},$$

where $q_i \in \mathbb{N}$. For the ease of writing we express it as

$$[q_0, q_1, q_2, \ldots].$$

If P_i/Q_i denotes the finite continued fraction $[q_0, \ldots, q_i]$ then we have the following recurrence

$$P_i = P_{i-2} + q_i P_{i-1} \text{ and } Q_i = Q_{i-2} + q_i Q_{i-1}, \tag{6}$$

where $P_{-1} := 0$ and $Q_{-1} := 1$. Furthermore, with the finite continued fraction $[q_0, \ldots, q_i] = \frac{P_i}{Q_i}$ we can associate the Möbius transformation

$$M(X) := \frac{P_{i-1} + P_i X}{Q_{i-1} + Q_i X}.$$

We denote by I_M the interval with end points $M(\infty) = P_m/Q_m$ and $M(0) = P_{i-1}/Q_{i-1}$. Since $[q_0, q_1, \ldots, q_i]$ is an ordinary continued fraction, we know that [Yap00, p. 463]

$$|P_i Q_{i-1} - P_{i-1} Q_i| = 1; \tag{7}$$

thus the Möbius transformation associated with an ordinary continued fraction is unimodal. Moreover, for any two numbers $\alpha, \eta \in \overline{\mathbb{C}} := \mathbb{C} \cup \infty$, such that $\alpha = M(\eta)$, we have

$$\eta = -\frac{P_{i-1} - Q_{i-1}\alpha}{P_i - Q_i \alpha}. \tag{8}$$

For a complex number z, let $\Re(z)$ represent its real part and $\Im(z)$ its imaginary part.

2.2 Termination

Consider the recursion tree of the procedure CF(A, M) initiated with $A(X) = A_{\text{in}}(X) \in \mathbb{R}[X]$ and $M(X) = X$, for a square-free polynomial $A_{\text{in}}(X)$. The right child of any node in this tree corresponds to the Taylor shift $X \to X + \delta$, $\delta \geq 1$, and the left child of the node corresponds to the inverse transformation $X \to (X + 1)^{-1}$. A sequence of Taylor shifts $X \to X_0+\delta_0, X_0 \to X_1+\delta_1, \ldots, X_{i-1} \to X_i+\delta_i$ can be thought of as a single Taylor shift $X \to X+q$, $q = \sum_{j=0}^{i} \delta_j$. Moreover, a sequence of Taylor shifts by a total amount q followed by an inverse transformation $X \to (X+1)^{-1}$ is the same as the transformation $X \to q+(1+X)^{-1}$. Thus with each node in the recursion tree we can associate an ordinary continued fraction $[q_0, q_1, \ldots, q_m] = P_m/Q_m$, for some $q_i \in \mathbb{N}$, and hence the Möbius transformation $(P_mX + P_{m-1})/(Q_mX + Q_{m-1})$; note that the nodes on the right most path of the recursion tree are associated with the continued fraction $[q_0]$, for some $q_0 \in \mathbb{N}$, and the Möbius transformation $X + q_0$, because there are no inverse transformations along the right most path. Based upon the Möbius transformation $(P_mX + P_{m-1})/(Q_mX + Q_{m-1})$ associated with a node in the recursion tree, we can further associate the polynomial $A_M(X) := (Q_mX+Q_{m-1})^n A_{\text{in}}(P_mX+P_{m-1})$ with the same node.

Vincent had stated that if m is large enough then $A_M(X)$ will exhibit at most one sign variation. Uspensky [Usp48, p. 298] quantified this by showing the following: Let $A_{\text{in}}(X) \in \mathbb{R}[X]$ be a square-free polynomial of degree n and Δ be the smallest distance between any pair of its roots; if m is such that

$$F_{m-1}\frac{\Delta}{2} > 1 \text{ and } F_{m-1}F_m\Delta > 1 + \epsilon_n^{-1}, \tag{9}$$

where F_i is the i-th Fibonacci number and $\epsilon_n := (1+1/n)^{1/(n-1)} - 1$, then $A_M(X)$ exhibits at most one sign variation [2]. Ostrowski [Ost50] improved and simplified Uspensky's criterion (9) to $F_mF_{m-1}\Delta \geq \sqrt{3}$. Similar criterion were derived by Alesina and Galuzzi [AG98, p. 246] and Yap [Yap00, Thm. 14.5, p. 476]. We next derive a termination criterion that depends on Δ_α, the shortest distance from a root α of $A(X)$ to any another root of $A(X)$. To describe this result, following [ESY06], we associate three open discs in the complex plane with an open interval $J = (c, d)$: the disc C_J is bounded by the circle that has centre $(c+d)/2$, and radius $(d-c)/2$; the disc $\overline{C}_J$ is bounded by the circle that has centre $(c+d)/2 + i(\sqrt{3}/6)(d-c)/2$, and passes through the end-points of J; and the disc $\underline{C}_J$ is bounded by the circle that has centre $(c+d)/2 - i(\sqrt{3}/6)(d-c)/2$, and passes through the end-points of J. In addition to these three discs, following [KM06], we also define the **cone**

$$\mathcal{C} := \left\{ a + ib | a \leq 0 \text{ and } |b| \leq |a|\sqrt{3} \right\}.$$

We have the following key observation which is implicit in Ostrowski's proof and is also used by Alesina and Galuzzi [AG98, p. 249]:

Lemma 2.4. *Let $a, b, c, d \in \mathbb{R}_{>0}$, I be an interval with unordered endpoints $\frac{a}{c}, \frac{b}{d}$, and define the Möbius transformation $M(z) := \frac{az+b}{cz+d}$. Then $M^{-1}(z)$ maps the closed region $\overline{\mathbb{C}} - (\overline{C}_{I_M} \cup \underline{C}_{I_M})$ bijectively on the cone $\mathcal{C}$, and maps the open disc C_{I_M} bijectively on the half plane $\Re(z) > 0$.*

[2] Uspensky's original proof incorrectly states $F_{m-1}\Delta > \frac{1}{2}$. This was later corrected by Akritas [Akr78a].

Proof. The proof follows by verifying the correspondence show in Figure 1. □

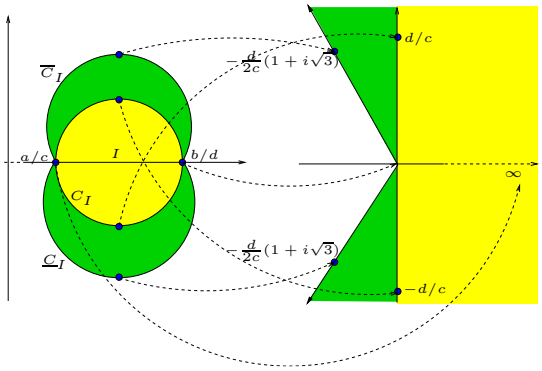

Figure 1: The effect of $M^{-1}(z)$ on the three circles

From Lemma 2.4 and from [KM06, Thm. 3.9] we know the following:

Theorem 2.5. *Let $A(X)$ be a square-free polynomial of degree n,*

$$M(X) := \frac{P_m X + P_{m-1}}{Q_m X + Q_{m-1}}$$

and $A_M(X) := (Q_m X + Q_{m-1})^n A(M(X))$. If α is the only simple root of $A(X)$ in the interval I_M and there are no other roots of $A(X)$ in $\overline{C}_{I_M} \cup \underline{C}_{I_M}$ then $\mathtt{Var}(A_M) = 1$.

The above theorem corresponds to the two-circle theorem in [KM06]. The corresponding one-circle theorem, which again is a direct consequence of Lemma 2.4 and [KM06, Thm. 3.9], is the following :

Theorem 2.6. *Let $A(X)$ be a square-free polynomial of degree n,*

$$M(X) := \frac{P_m X + P_{m-1}}{Q_m X + Q_{m-1}}$$

and $A_M(X) := (Q_m X + Q_{m-1})^n A(M(X))$. If C_{I_M} does not contain any roots then $\mathtt{Var}(A_M) = 0$.

3 The Size of the Recursion Tree

In this section we bound the number of nodes, $\#(T)$, in the recursion tree T of the procedure described in Section 2 initiated with a square-free polynomial $A_{\text{in}}(X) \in \mathbb{R}[X]$ of degree n and the Möbius transformation X.

We partition the leaves of T into two types: **type-0** leaves are those that declare the absence of a real root and **type-1** leaves are those that declare the presence of a real root.

Consider the tree T' obtained by pruning certain leaves from T: prune all the type-0 leaves that have either a non-leaf or a type-1 leaf as sibling; if two leaves are siblings of each

other then arbitrarily prune one of them. Thus, $\#(T') < \#(T) \leq 2\#(T')$ and we bound $\#(T')$.

Let U be the set of leaves of T'. Recall from the beginning of Section 2.2 that with each node in T' we can associate an ordinary continued fraction. In particular, for a leaf $u \in U$ let $[q_0, \ldots, q_{m+1}] = P_{m+1}/Q_{m+1}$ be the associated continued fraction and $I_u := I_{M_u}$ the corresponding interval. We can further associate with u a unique pair (α_u, β_u) of roots of $A_{\text{in}}(X)$. To do this we need to consider the parent v of u. Let the Möbius transformation $M_v(X)$ associated with v be

$$M_v(X) = \frac{P_m X + P_{m-1} + P_m \delta_v}{Q_m X + Q_{m-1} + Q_m \delta_v}, \tag{10}$$

for some $\delta_v \in \mathbb{N}$ such that $1 \leq \delta_v < q_{m+1}$, and the interval associated with v be $I_v := I_{M_v}$. Since v is not a leaf we know that $\texttt{Var}(A_{M_v}) > 1$. To each leaf $u \in U$ we assign a unique pair (α_u, β_u) of roots of $A_{\text{in}}(X)$ as follows:

1. If u is a type-1 leaf then there is a unique root $\alpha_u \in I_u$. Since $\texttt{Var}(A_{M_v}) > 1$, from Theorem 2.5 we know that there must be a root in $\overline{C}_{I_v} \cup \underline{C}_{I_v}$ apart from α_u; let β_u be one such root. Thus with each type-1 leaf we can associate a pair (α_u, β_u). Moreover, this can be done in a unique manner. Suppose u' is another type-1 leaf and v' is its parent then $\alpha_u \neq \alpha_{u'}$. From [ESY06, Lem. 3.2] it is clear that we only need to consider the case when I_v and $I_{v'}$ are adjacent to each other. Moreover, assume β_u and $\overline{\beta_u}$ are the only non-real roots in $\overline{C}_{I_v} \cup \underline{C}_{I_v}$ and $\overline{C}_{I_{v'}} \cup \underline{C}_{I_{v'}}$. Then it must be that either $\beta_u \in \overline{C}_{I_v} \cap \overline{C}_{I_{v'}}$ or $\beta_u \in \underline{C}_{I_v} \cap \underline{C}_{I_{v'}}$. In either case we can choose $\beta_{u'} = \overline{\beta_u}$ distinct from β_u.

2. If u is a type-0 leaf then it had a type-0 leaf as its sibling in T. We consider two sub-cases:

   - If I_v does not contain a real root then we know from Theorem 2.6 that there must be a pair of complex conjugate roots in C_{I_v}. Let (α_u, β_u), $\beta_u := \overline{\alpha_u}$, be one such pair. The uniqueness of the pair is immediate since C_{I_v} does not overlap with $C_{I_{v'}}$ for the parent v' of any other type-0 leaf.
   - If I_v does contain a root then it must be the midpoint of the interval I_v; let α_u denote this root. From Theorem 2.5 we also know that there must be a pair of complex conjugates $(\beta, \overline{\beta})$ in $\overline{C}_{I_v} \cup \underline{C}_{I_v}$; choose $\beta_u := \beta$. The pair is unique because α_u is unique.

We will bound the number of nodes in the path terminating at the leaf u by bounding the number of inverse transformations $X \to 1/(X+1)$ and Taylor shifts $X \to X + b$, $b \geq 1$. Before we proceed further, we have two observations: first, because of the uniqueness of the pair (α_u, β_u) it follows that the size of the set $|U| \leq n$; and second, since $\alpha_u, \beta_u \in \overline{C}_{I_v} \cap \underline{C}_{I_v}$ we know that

$$(Q_m(Q_{m-1} + \delta_v Q_m))^{-1} > \frac{\sqrt{3}}{2}|\alpha_u - \beta_u| > \frac{1}{2}|\alpha_u - \beta_u|, \tag{11}$$

where δ_v is defined as in (10).

3.1 Bounding the Inverse Transformations

From (11) it follows that

$$|\alpha_u - \beta_u| < 2(Q_m Q_{m-1})^{-1}.$$

But from (6) we know that Q_i is greater than the $i+1$-th Fibonacci number; this implies that $Q_i \geq \phi^i$, where $\phi = (1+\sqrt{5})/2$. Thus

$$\phi^{2m-1} \leq 2|\alpha_u - \beta_u|^{-1}$$

and hence

$$m \leq \frac{1}{2}(1 + \log_\phi 2 - \log_\phi |\alpha_u - \beta_u|). \tag{12}$$

So the total number of inverse transformations in T' are bounded by

$$\sum_{u \in U} \frac{1}{2}(1 + \log_\phi 2 - \log_\phi |\alpha_u - \beta_u|) \leq 2n + \sum_{u \in U} \log_\phi (|\alpha_u - \beta_u|)^{-1}. \tag{13}$$

3.2 Bounding the Taylor Shifts

The purpose of the Taylor shifts in the procedure $\mathrm{CF}(A, M)$ was to compute the floor of the smallest positive root of a polynomial. Using property (5) of the $\mathrm{PLB}(A)$ function (defined in (4)) we will bound the number of Taylor shifts required to compute the floor of the smallest positive root of some polynomial $B(X) \in \mathbb{R}[X]$. Before we do so, we have the following observation on the effect of shifts in the complex plane:

Lemma 3.1. *If $\alpha, \beta \in \mathbb{C}$ are such that $|\alpha| \leq |\beta|$ and $|\alpha - \delta| \geq |\beta - \delta|$, for any positive real number δ, then $\Re(\beta) \geq \Re(\alpha)$.*

Proof. $|\alpha - \delta| \geq |\beta - \delta|$ implies

$$2\delta(\Re(\beta) - \Re(\alpha)) \geq |\beta|^2 - |\alpha|^2 \geq 0.$$

Since δ is positive we have our result. □

Intuitively, this lemma says if the origin is shifted to the right then only the complex numbers to the right of the number α and in absolute value greater than α can possibly become smaller than α in absolute value.

We introduce the following notation for convenience: for any $x \in \mathbb{R}_{\geq 0}$ let

$$\mathrm{logm}\,(x) := \log \max(1, x).$$

We start with the following simple case:

Lemma 3.2. *Let $B_1(X) \in \mathbb{R}[X]$ be a polynomial all of whose roots are in the open half plane $\Re(z) > 0$. For $i > 1$ recursively define*

$$B_i(X) := B_{i-1}(X + \delta_{i-1})$$

where

$$\delta_{i-1} := \begin{cases} \mathrm{PLB}(B_{i-1}) + 1 & \text{if } \mathrm{PLB}(B_{i-1}) > 1 \\ 1 & \text{otherwise.} \end{cases}$$

Let α_1 denote a root of $B_1(X)$ with the smallest absolute value, and recursively let $\alpha_i = \alpha_{i-1} - \delta_{i-1}$. Then $\Re(\alpha_i) \leq 1$ if $i \geq 2 + 8n + \gamma_n \mathrm{logm}\, \Re(\alpha_1)$.

Proof. See Appendix for proof. □

We next extend this lemma to the case when the polynomial has roots with negative real parts but zero is still not its root. To derive this result we introduce the following definition:

Definition 3.3. *Let LP(B) denote the root of $B(X)$ in $\Re(z) > 0$ that has the smallest real part and the smallest absolute value, and LN(B) denote the root of $B(X)$ in $\Re(z) \leq 0$ that has the largest real part and the smallest absolute value.*

Lemma 3.4. *Let $B_1(X) \in \mathbb{R}[X]$, $B_1(0) \neq 0$. Recursively define δ_i, and $B_i(X)$ as in the above lemma. Let $\alpha_1 := LP(B_1)$, $\beta_1 := LN(B_1)$ and recursively define $\alpha_i = \alpha_{i-1} - \delta_{i-1}$ and $\beta_i = \beta_{i-1} - \delta_{i-1}$. If*

$$i = \Omega\left(n + \kappa_n \mathrm{logm}\, \frac{|\alpha_1|}{|\beta_1|} + \kappa_n \mathrm{logm}\, |\alpha_1|\right)$$

then $\Re(\alpha_i) \leq 1$, where

$$\kappa_n := (\log(8n+1) - \log 8n)^{-1}. \tag{14}$$

Proof. See Appendix for proof. □

Based upon the above two lemmas we will bound the number of Taylor shifts from the root of T' to the leaf u, with the associated continued fraction $[q_0, \ldots, q_{m+1}]$, by bounding the number of Taylor shifts that compose each q_i, $i = 0, \ldots, m+1$. Recall from the beginning of this section the definitions of the two Möbius transformation $M_u(X)$ and $M_v(X)$, the intervals I_u and I_v, and the pair (α_u, β_u) for a leaf $u \in U$. We further define the following quantities:

Definition 3.5. *For $0 \leq i \leq m+1$ let*

1. $M_i(X) := [q_0, \ldots, q_i, 1 + X] = \frac{P_i X + P_{i-1} + P_i}{Q_i X + Q_{i-1} + Q_i}$;
2. $A_i(X) := (Q_i X + Q_{i-1} + Q_i)^n A_{in}(M_i(X))$, *i.e., the polynomial obtained by performing the ith inverse transformation and on which we will perform a Taylor shift by the amount q_{i+1};*
3. $\eta_i := M_i^{-1}(\alpha_u)$
4. $r_i := P_i/Q_i$, $s_i := \frac{P_i + P_{i-1}}{Q_i + Q_{i-1}}$ *and*
5. $J_i := I_{M_i}$, *i.e., the interval with endpoints r_i and s_i.*

By its definition J_i, for $0 \le i \le m$, contains I_u and hence it follows from (11) that for $0 \le i \le m$

$$(Q_i Q_{i-1})^{-1} \ge \frac{|\alpha_u - \beta_u|}{2}. \tag{15}$$

We now bound the number of Taylor shifts required to obtain q_{i+1}. Let $B_1(X) := A_i(X)$, and recursively define the polynomials $B_i(X)$ as in Lemma 3.2. Define the sequence of indices

$$1 = i_0 \le i_1 < i_2 < \cdots < i_\ell, \tag{16}$$

where the index i_j is such that $\Re(\mathrm{LP}(B_{i_j}))$ is contained in the unit interval; if $i < m$ the last index i_ℓ is such that the real part of the root in $B_{i_\ell}(X)$ corresponding to η_i is in the unit interval; if $i = m$ the index i_ℓ is such that the node that has $B_{i_\ell}(X)$ as the corresponding polynomial is the parent v of the leaf u. Clearly, $\ell \le n$.

From Lemma 3.4 we know that

$$i_{j+1} - i_j = O\left(n + \kappa_n \mathrm{logm}\, \frac{|\mathrm{LP}(B_{1+i_j})|}{|\mathrm{LN}(B_{1+i_j})|} + \kappa_n \mathrm{logm}\, |\mathrm{LP}(B_{1+i_j})|\right).$$

Summing this inequality for $j = 0, \ldots, \ell - 1 < n$ we get that if

$$i_\ell = O(n^2) + O\left(\sum_{j=0}^{\ell-1} \kappa_n \mathrm{logm}\, \frac{|\mathrm{LP}(B_{1+i_j})|}{|\mathrm{LN}(B_{1+i_j})|} + \kappa_n \mathrm{logm}\, |\mathrm{LP}(B_{1+i_j})|\right) \tag{17}$$

then the real part of the root in $B_{i_\ell}(X)$ that corresponds to η_i is in the unit interval, i.e., the number of Taylor shifts which constitute q_{i+1} are bounded by this bound.

The last term in the summation above is smaller than

$$\kappa_n \left(\mathrm{logm}\, \frac{|\eta_i|}{|\mathrm{LN}(B_{1+i_{\ell-1}})|} + \mathrm{logm}\, |\eta_i|\right), \tag{18}$$

because $|\mathrm{LP}(B_{1+i_{\ell-1}})|$ is smaller than $|\eta_i|$. We call this term the *contribution of* α_u *to* q_{i+1}. Our aim now is to bound it primarily as a function of $\log |\alpha_u - \beta_u|^{-1}$; the advantage becomes evident when we try to sum the term over all $u \in U$, since then we can use the Davenport-Mahler bound [ESY06, Thm. 3.1] to give an amortized bound on the $\sum_{u \in U} \log |\alpha_u - \beta_u|^{-1}$. The remaining terms in the summation in (17) are the contributions of different $\alpha_{u'}$ to q_{i+1}, where $u' \in U - \{u\}$ is such that $\eta_{1+i_j} = M_i^{-1}(\alpha_{u'})$, and can be bounded in terms of $|\alpha_{u'} - \beta_{u'}|$. Note that the contribution of α_u to q_0 is not accounted for, but this will be taken care of later.

We next derive an upper bound on $|\eta_i|$ and a lower bound on $|\mathrm{LN}(B_{1+i_{\ell-1}})|$. In deriving these bounds we will often require lower bounds on $|\alpha - P/Q|$, where α is a root of a degree n polynomial $A(X)$ and P/Q is a fraction such that $0 < |\alpha - P/Q| \le 1$ and $A(P/Q) \ne 0$. The lower bounds in the literature can be parametrized by some $N \in \mathbb{R}_{\ge 1}$ as follows:

$$|\alpha - P/Q| \ge C(A, N) \cdot Q^{-N}; \tag{19}$$

note that the lower bound holds for all conjugates of α. For example, in case of Liouville's inequality [Lio40] we have $N = n$; for Roth's theorem [Rot55] we have $N > 2$; for Thue's

result [Thu09] we have $N > 1 + n/2$; and for Dyson's result [Dys47] we have $N > \sqrt{2n}$. However, explicit bounds on $C(A, N)$ are known only for Liouville's inequality.

In bounding $|\eta_i|$ and $|\mathrm{LN}(B_{1+i_{\ell-1}})|$, we need to consider two separate cases depending upon whether Q_i is zero or not. The situation $Q_i = 0$ occurs only on the right-most path of the tree T since there are no inverse transformations along this path. In bounding the length of the right-most path we will also account for the contribution of α_u to q_0, for all $u \in U$. The argument for bounding the length of the path is similar to the argument that was used to derive the bound in (17) above.

Let $B_1(X) := A_{\mathrm{in}}(X)$ and recursively define $B_i(X)$ as in Lemma 3.2. Define the sequence of indices as in (16) and follow the same line of argument used to obtain (17), except now we can replace $|\mathrm{LP}(B_{1+i_j})|$ by $|\eta_{i_j}|$, the absolute value of some root of $A_{\mathrm{in}}(X)$ in $\Re(z) > 0$. Moreover, we know that $|\eta_{i_j}| \leq \mu(A_{\mathrm{in}})$. To obtain a lower bound on $|\mathrm{LN}(B_{1+i_j})|$ we observe that $\mathrm{LN}(B_{1+i_j}) = \alpha - \delta$, where α is some root of $A_{\mathrm{in}}(X)$ and $\delta \in \mathbb{N}$ is such that $B_{1+i_j}(X) = A_{\mathrm{in}}(X + \delta)$, and hence from (19) we get $|\mathrm{LN}(B_{1+i_j})| \geq C(A_{\mathrm{in}}, N)$. Thus the length of the right-most path in the tree T' is bounded by

$$\kappa_n n(\mathrm{logm}\, \mu(A_{\mathrm{in}}) - \mathrm{logm}\, C(A_{\mathrm{in}}, N)). \tag{20}$$

Assuming that $Q_i \geq 1$, we can show the following bounds (see Appendix for details):

$$\log \eta_i \leq -N \log |\alpha_u - \beta_u| - \log C(A_{\mathrm{in}}, N) + N + 1, \tag{21}$$

$$-\log |\mathrm{LN}(B_{1+i_{\ell-1}})| = O(-N \log |\alpha_u - \beta_u| - \log C(A_{\mathrm{in}}, N) + \log \mu(A_{\mathrm{in}})). \tag{22}$$

Note that we may safely assume that $|\mathrm{LN}(B_{1+i_{\ell-1}})| \neq 0$ since if zero is a root of $B_{1+i_{\ell-1}}(X)$ then in the procedure $\mathrm{CF}(A, M)$ we always divide the polynomial by X and remove this degenerate case. Since both $|\alpha_u - \beta_u|$ and $C(A_{\mathrm{in}}, N)$ are smaller than one (see Lemma 3.9), the bound in (22) dominates the bound in (21), and hence the term in (18) is bounded by

$$\kappa_n O(-N \log |\alpha_u - \beta_u| - \log C(A_{\mathrm{in}}, N) + \log \mu(A_{\mathrm{in}})).$$

Thus the total contribution of α_u to each q_i, $i = 1, \ldots, m+1$, is bounded by the sum of this bound from $i = 1, \ldots, m+1$, i.e., by

$$\sum_{i=1}^{m} \kappa_n O(-N \log |\alpha_u - \beta_u| - \log C(A_{\mathrm{in}}, N) + \log \mu(A_{\mathrm{in}})),$$

where m satisfies (12); to show the dependency of m on the choice of the leaf u, from now on we write m as m_u. Thus the total number of Taylor shifts along the path starting from the root of the tree T' and terminating at the leaf $u \in U$, except the leaf of the right-most path, is bounded by

$$\sum_{i=1}^{m_u} \sum_{u' \in U} \kappa_n O(-N \log |\alpha_{u'} - \beta_{u'}| - \log C(A_{\mathrm{in}}, N) + \log \mu(A_{\mathrm{in}})),$$

where u' are the leaves to the left of u and that share a common ancestor with u. The total number of Taylor shifts in the tree T' is obtained by summing the above bound for all $u \in U$

and adding to it the bound in (20) on the length of the right-most path:

$$\sum_{u\in U}\sum_{i=1}^{m_u}\sum_{u'\in U}\kappa_n O(-N\log|\alpha_{u'}-\beta_{u'}|-\log C(A_{\text{in}},N)+\log\mu(A_{\text{in}})). \tag{23}$$

Combined with the bound in (13) on the total number of inverse transformations in the tree T', we get the following bound on the number of nodes in the tree T'

$$\begin{aligned}\#(T') = O(n+\sum_{u\in U}\log_\phi|\alpha_u-\beta_u|^{-1}) \\ +\sum_{u\in U}\sum_{i=1}^{m_u}\sum_{u'\in U}\kappa_n O(-N\log|\alpha_{u'}-\beta_{u'}|-\log C(A_{\text{in}},N)+\log\mu(A_{\text{in}})).\end{aligned} \tag{24}$$

Recall from the beginning of this section that the number of nodes in the recursion tree T of Akritas' algorithm is $\Theta(\#(T'))$, so the above bound applies to $\#(T)$ as well.

3.3 Worst Case Size of the Tree

In order to derive a worst-case bound on the size of the tree T, from the bound given in (24), we need to derive an upper bound on $\sum_{u\in U}-\log|\alpha_u-\beta_u|$. For this purpose we resort to the Davenport-Mahler bound :

Proposition 3.6. *Let $A(X)=a_n\prod_{i=1}^n(X-\alpha_i)$ be a square-free complex polynomial of degree n. Let $G=(V,E)$ be a directed graph whose nodes $\{v_1,\ldots,v_k\}$ are a subset of the roots of $A(X)$ such that*

1. *If $(v_i,v_j)\in E$ then $|v_i|\le|v_j|$.*
2. *G is acyclic.*
3. *The in-degree of any node is at most 1.*

If exactly m of the nodes have in-degree 1, then

$$\prod_{(v_i,v_j)\in E}|v_i-v_j|\ge\sqrt{|\text{discr}(A)|}\cdot\text{M}(A)^{-(n-1)}\cdot(n/\sqrt{3})^{-m}\cdot n^{-n/2}.$$

See [ESY06] for a proof.

Remark 3.7. *Let* $\text{sep}(A)$ *be the minimum distance between two distinct roots of $A(X)$. Then we have*

$$\text{sep}(A)\ge\sqrt{\text{discr}(A)}\text{M}(A)^{-(n-1)}\cdot(n/\sqrt{3})\cdot n^{-n/2}.$$

Consider the graph G whose edge set is $E_1\cup E_0$, where $E_0:=\{(\alpha_u,\beta_u)\}$, u is a type-0 leaf and $E_1:=\{(\alpha_u,\beta_u)\}$, u is a type-1 leaf. We will show that G satisfies the conditions of Proposition 3.6. First of all, for any $u\in U$ we can reorder the pair (α_u,β_u) to ensure that $|\alpha_u|\le|\beta_u|$ without affecting the summation $\sum_{u\in U}-\log|\alpha_u-\beta_u|$. We note that the graph so obtained is similar to the graph described in the proof of [ESY06, Thm. 3.4]; thus after

properly reordering the edges as was mentioned there we may directly apply Proposition 3.6 to G to obtain

$$\sum_{u \in U} -\log |\alpha_u - \beta_u| = O(B(A_{\text{in}})), \tag{25}$$

where

$$B(A_{\text{in}}) := O(n \log \mathrm{M}(A_{\text{in}}) + n \log n - \log \operatorname{discr}(A_{\text{in}})), \tag{26}$$

$\mathrm{M}(A_{\text{in}})$ is the Mahler measure of $A_{\text{in}}(X)$ and $\operatorname{discr}(A_{\text{in}})$ is its discriminant (see [Yap00, Sec. 6.6, Sec. 4.5], [Mc99, Sec. 1.5, Sec. 2.1]). Based upon this bound we have the following:

Theorem 3.8. *Let $A_{in}(X) \in \mathbb{R}[X]$ be a square-free polynomial of degree n and T be the recursion tree of Akritas' algorithm applied to $A_{in}(X)$. The number of nodes in T is*

$$nO(NB(A_{in})^2 - nB(A_{in}) \log C(A_{in}, N) + nB(A_{in}) \log \mu(A_{in})),$$

where $B(A_{in})$ is defined in (26), $C(A_{in}, N)$ is the constant involved in the inequality (19), and $\mu(A_{in})$ is the largest absolute value amongst all the roots of $A(X)$.

Proof. Applying the bound in (25), along with the observation that $|U| \leq n$, to (24) we get that the size of the tree is bounded by

$$O(n + B(A_{\text{in}})) + \sum_{u \in U} \sum_{i=1}^{m_u} \kappa_n O(-NB(A_{\text{in}}) - n \log C(A_{\text{in}}, N) + n \log \mu(A_{\text{in}})).$$

From (12) we further get that the above bound is smaller than

$$\kappa_n O(-NB(A_{\text{in}}) - n \log C(A_{\text{in}}, N) + n \log \mu(A_{\text{in}})) \sum_{u \in U} \frac{1}{2}(1 + \log_\phi 2 - \log_\phi 2|\alpha_u - \beta_u|).$$

Again applying (25), we get that the size of the tree is bounded by

$$\kappa_n O(NB(A_{\text{in}})^2 - nB(A_{\text{in}}) \log C(A_{\text{in}}, N) + nB(A_{\text{in}}) \log \mu(A_{\text{in}})).$$

From the observation that $\kappa_n = \Theta(n)$ (see its definition in (14), we get the desired result. □

We will next give a specialization of the above theorem for integer polynomials, but for achieving this we need to derive bounds on the quantities N and $C(A, N)$ involved in (19).

Lemma 3.9. *Let α be a root of an integer polynomial $A(X)$ of degree n. Suppose $P/Q \in \mathbb{Q}$, $Q > 0$, is such that $0 < |\alpha - P/Q| \leq 1$ and $A(P/Q) \neq 0$, then $|\alpha - P/Q| \geq C(\alpha) \cdot Q^{-n}$ where*

$$C(\alpha) \geq 2^{-n - \log n - (n+1) \log \|A\|_\infty}. \tag{27}$$

Proof. See Appendix for proof. □

We now have the desired specialization of the theorem above.

Corollary 3.10. *Let $A(X)$ be a square-free polynomial of degree n with integer coefficients of magnitude less than 2^L. The number of nodes in the recursion tree of Akritas' algorithm run on $A(X)$ is $\widetilde{O}(n^4 L^2)$.*

Proof. From Landau's inequality (e.g., [Yap00, Lem. 4.14(i)]) and the estimate $\|A\|_2 \leq \sqrt{n+1}\|A\|_\infty$ we get

$$\mathrm{M}(A) \leq \|A\|_2 \leq \sqrt{n+1}\|A\|_\infty < \sqrt{n+1}2^L.$$

Moreover, $|\mathrm{discr}(A)| \geq 1$ since $A(X)$ is square-free and its coefficients are integers. From these observations we conclude that $B(A) = \widetilde{O}(nL)$. Furthermore, from Cauchy's bound [Yap00, Cor. 6.8,p. 149] we know that $\mu(A) \leq 2^L$. Plugging these bounds, along with the bounds in Lemma 3.9, in Theorem 3.8, we obtain the desired result. □

Remark 3.11. *Notice that asymptotically the bounds in (21) and (22) are the same, so even showing the tightness (as in (2)) of the bounds in [Kio86, Şte05, Hon98] does not improve upon the complexity result in the above corollary, though it will definitely lead to a simplification of the analysis.*

How good is this bound? The answer depends upon the tightness of the bounds derived on the number of inverse transformations and Taylor shifts. The bound derived on the former, in (13), is perhaps the best one can expect, considering that the same bound holds for root isolation using Sturm's method [Dav85, DSY05] and for the Descartes method [ESY06, EMT06]. The best possible scenario for the number of Taylor shifts is based upon the ideal PLB function, which we recall is a function that can determine whether a polynomial has positive real roots, and if there are such roots then returns a value that is at a constant distance from the smallest positive root of the polynomial.

Lemma 3.12. *For a square-free polynomial of degree n and integer coefficients of bit-length L, the number of Taylor shift in the recursion tree of Akritas' algorithm, if it uses the ideal PLB function, is $\widetilde{O}(nL)$.*

Proof. The total number of Taylor shifts required along a path is proportional to the number of inverse transformations along the path, because between two consecutive inverse transformations we only perform a constant number of Taylor shifts. Thus the total number of Taylor shifts in the recursion tree is proportional to the number of inverse transformations in the tree and hence the best possible bound on the size of the recursion tree is $\widetilde{O}(nL)$. □

This shows that there is a huge gap to be overcome in the bound derived in Corollary 3.10.

4 The Bit-Complexity

In this section we derive the bit-complexity of Akritas' algorithm for a square-free polynomial $A_{\mathrm{in}}(X)$ such that $\|A_{\mathrm{in}}\|_\infty < 2^L$. To do this we will bound the worst-case complexity at any node in the recursion tree; then along with Corollary 3.10 we have a bound on the bit-complexity of the algorithm.

Recall from starting of Section 3 the definitions of the set U, and of the pair (α_u, β_u) for any $u \in U$. Let

$$M_u(X) = [q_0, \ldots, q_{m+1}, X] = \frac{P_{m+1}X + P_m}{Q_{m+1}X + Q_m},$$

$$M_i(X) = [q_0, \ldots, q_i, X] = \frac{P_i X + P_{i-1}}{Q_i X + Q_{i-1}},$$

$A_i(X) = (Q_i X + Q_{i-1})^n A(M_i(X))$, L_i be such that $\|A_i\|_\infty < 2^{L_i}$, and b_i the bit-length of q_i for $i = 0, \ldots, m+1$.

To construct $A_{i+1}(X)$ from $A_i(X)$ we need to construct a sequence of polynomials $B_j(X)$, $1 \leq j \leq \ell$, such that $B_1(X) := A_i(X)$ and for $j > 1$, $B_j(X) := B_{j-1}(X + \delta_{j-1})$, where

$$\delta_j := 1 + \begin{cases} \mathrm{PLB}(B_{j-1}) & \text{if } \mathrm{PLB}(B_{j-1}) > 1 \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, $q_{i+1} = \sum_{j=1}^{\ell-1} \delta_j$. The two most important operations in computing $B_j(X)$ from $B_{j-1}(X)$ are computing $\mathrm{PLB}(B_{j-1})$ and the Taylor shift by δ_j. We only focus on the latter operation since its cost dominates the cost of computing the former operation, which we know from Remark 2.2 is $\widetilde{O}(nL_i)$. Since $\delta_j \leq q_{i+1}$, for $j < \ell$, the cost of computing each of Taylor shifts, i.e., the cost of computing $B_j(X)$ from $B_{j-1}(X)$ for all $j \leq \ell$, is bounded by the cost of computing $A_i(X + q_{i+1})$; we bound this latter cost.

We know (see [Kra95]) that the computation of the Taylor shift can be arranged in a triangle of depth n; at each depth the multiplication by q_{i+1} increases the bit-length by b_{i+1}, so the bit-length of the coefficients of $A_i(X + q_{i+1})$ is bounded by $L_i + nb_{i+1}$. Moreover, using the classical approach, Taylor shifts can be performed in $O(n^2)$ additions [Kra95, JKR05, vzGG97]. Thus the cost of computing $A_i(X + q_{i+1})$ is $O(n^2(L_i + nb_{i+1}))$. We further claim that $L_i = O(L + n\sum_{j=0}^{i} b_j)$; this is straightforward from the observation that $L_j \leq L_{j-1} + nb_{j-1}$. Thus the bit-complexity of computing $A_i(X + q_{i+1})$ is bounded by $O(n^2(L + n\sum_{j=0}^{i+1} b_j))$, if we use the classical Taylor shift. We next bound $\sum_{j=0}^{i+1} b_j$, $i \leq m$.

We know that $Q_m = q_m Q_{m-1} + Q_{m-2}$; thus $Q_m \geq q_m Q_{m-1}$, and recursively we get that $Q_m \geq \prod_{j=1}^{m} q_j$. Moreover, from (15) and the worst-case separation bound (see Remark 3.7) we know that $\log Q_m = \widetilde{O}(nL)$. Thus $\sum_{j=0}^{m} b_j = \widetilde{O}(nL)$. The troublesome part is bounding q_{m+1}, since Q_{m+1} does not satisfy (15). However, we do know that $q_{m+1} \leq |M_m^{-1}(\alpha_u)|$, and from (21) that

$$\log |M_m^{-1}(\alpha_u)| = O(-N \log |\alpha_u - \beta_u| - \log C(A_{\mathrm{in}}, N)).$$

But from Lemma 3.9 we have $N = n$ and $-\log C(A_{\mathrm{in}}, N) = \widetilde{O}(nL)$, and from the separation bound it follows that $-\log |\alpha_u - \beta_u| = \widetilde{O}(nL)$. Thus $b_{m+1} = \widetilde{O}(n^2 L)$ and hence $\sum_{j=0}^{m+1} b_j = \widetilde{O}(n^2 L)$.

So the worst-case bit-complexity at any node is asymptotically the same as computing $A_m(X)$, which we know is $\widetilde{O}(n^2(L + n\sum_{j=0}^{i+1} b_j)) = \widetilde{O}(n^5 L)$, when we use classical Taylor shifts. Along with the result in Corollary 3.10 we get the following:

Theorem 4.1. *Let $A(X)$ be a square-free integer polynomial of degree n with integer coefficients of magnitude less than 2^L. Then the bit-complexity of isolating all the real roots of $A(X)$ using Akritas' algorithm based upon classical Taylor shift is $\widetilde{O}(n^9 L^3)$.*

We can improve on the above bound by a factor of n using the fast Taylor shift [vzGG97].

Theorem 4.2. *Let $A(X)$ be a square-free integer polynomial of degree n with integer coefficients of magnitude less than 2^L. Then the bit-complexity of isolating all the real roots of $A(X)$ using Akritas' algorithm based upon a fast Taylor shift is $\widetilde{O}(n^8L^3)$.*

Proof. The cost of computing $A_i(X + q_{i+1})$ using the convolution method (method F in [vzGG97]) is $O(M(n^2b_{i+1} + nL_i))$, where $M(n)$ is the complexity of multiplying two n-bit integers. From above we know that $L_i = O(L + n\sum_{j=0}^{i} b_j)$, thus the cost is $O(M(nL + n^2\sum_{j=0}^{i+1} b_j))$. Moreover, we also know that $\sum_{j=0}^{m+1} b_j = \widetilde{O}(n^2L)$. Assuming the Schönhage-Strassen method and the Turing machine as the underlying computational model we have $M(n) = O(n\log(n)\log\log(n)) = \widetilde{O}(n)$. Hence the worst-case bit-complexity of a node is $\widetilde{O}(n^4L)$. Multiplying with the bound $\widetilde{O}(n^4L^2)$ (from Corollary 3.10) on the size of the tree we get the complexity as mentioned in the theorem. □

Remark 4.3. *If we were to use the ideal* PLB *function then the worst case bit complexity of Akritas' algorithm is $\widetilde{O}(n^5L^2)$, since in this case the size of the tree is $\widetilde{O}(nL)$ (Lemma 3.12) and we know that the worst case complexity of each node is $\widetilde{O}(n^4L)$.*

5 Conclusion and Future Work

The bound in Theorem 4.2 is not as impressive as the complexity of the Descartes method, which we know (e.g., see [ESY06, Thm. 4.2]) is $\widetilde{O}(n^4L^2)$. This disparity arises because of the difference between the bounds on the size of the recursion trees of the two algorithms: whereas for the Descartes method the bound is $\widetilde{O}(nL)$, which is known to be almost tight, the corresponding bound for Akritas' algorithm is $\widetilde{O}(n^4L^2)$, as derived in Corollary 3.10. This difference stems from the following reasons:

1. In our analysis we had to use Liouville's inequality instead of Roth's theorem, because for the latter result we do not know any bounds on the constant $C(A, N)$, even though $N = O(1)$. However, if we assume that the constant $C(A, N)$ for Roth's theorem is the same as that in Lemma 3.9 then it follows that the size of the recursion tree of Akritas' algorithm is $\widetilde{O}(n^3L^2)$, the worst case complexity of a node in the recursion tree is $\widetilde{O}(n^3L)$, and hence the worst case complexity of the algorithm is $\widetilde{O}(n^6L^3)$; note that under this assumption $\log q_{m+1} = \widetilde{O}(nL)$ (instead of $\widetilde{O}(n^2L)$) as expected. Moreover, if we additionally assume the ideal PLB function then we would get a worst case complexity of $\widetilde{O}(n^4L^2)$, which matches the expected bound in [ET06] (also derived under the same assumption) and the worst case complexity of the Descartes method. The assumption that the constant $C(A, N)$ in Roth's theorem satisfies the same bound as in Lemma 3.9 is reasonable since it is known that, barring finitely many rationals, $C(A, N) = 1$ in (19). Thus the bound $\widetilde{O}(n^6L^3)$ is a more accurate statement on the actual performance of Akritas' algorithm than the bound in Theorem 4.2. But perhaps we can improve the latter result by a factor of n if we use Roth's theorem whenever possible, and use Liouville's inequality for those rationals that are exceptions to Roth's theorem, along with bounds [Sch95] on the number of these exceptions.

2. It is clear from above that even if we allow ourselves the liberty of using Roth's theorem, we do not achieve as tight a bound on the recursion tree as for the Descartes method.

The bottleneck is bounding the number of Taylor shifts in the tree. Presently we perceive two ways of improving our analysis:

- Show that the bounds described in Section 2 satisfy a tighter inequality compared to the inequality, (2), satisfied by Zassenhaus' bound.
- An alternative to performing a Taylor shift by b in step five of the procedure $\mathrm{CF}(A, M)$ is to scale by b and shift by one; clearly this reduces the number of steps needed to compute the floor of the smallest positive root of the polynomial, but this is achieved at the cost of increasing the bit-size of the coefficients – this trade-off needs to be explored further.

An interesting result is to show that the size of the recursion tree of Akritas' algorithm applied to Mignotte's polynomial, $X^n - 2(ax-1)^2$, $a \in \mathbb{N}$, is always smaller than the size of the corresponding tree for the Descartes method. For instance, we observed that for $a = 101$ and $n = 10, 20, 30, , \ldots, 100, 200, \ldots, 3000, 4000$ the size of the recursion tree increases until a certain point and thereafter remains the same. This holds for both `SYNAPS` [MPTT05], where the implementation uses scaling in step five of $\mathrm{CF}(A, M)$, and for `Core Library`[YLP+04], where the implementation is the procedure $\mathrm{CF}(A, M)$. We know that the size of the recursion tree of the Descartes method applied to Mignotte's polynomial is $\Omega(n \log a)$. We believe that the corresponding bound for Akritas' algorithm is $\widetilde{O}(\log a)$, which would explain our observation and substantiate its superior performance when applied to Mignotte's polynomials.

A likely direction to pursue is to modify Akritas' algorithm so that its complexity bound improves without affecting its efficiency in practice. One way to modify the algorithm is to ensure that at each recursive level the width of the interval decreases by half [3]. Even though this direction is worth pursuing, it is not evident that it will perform better than the current implementation in all scenarios, since subdivision is not always the right approach, as is manifested in the case of Mignotte's polynomials. However, this is a different direction from our pursuit in this paper, namely to understand the worst case behaviour of the original algorithm by Akritas.

Acknowledgements: The author is indebted to Prof. Chee Yap, Prof. Bernard Mourrain, and Elias Tsigaridas for their suggestions and criticisms.

References

[AG98] Alberto Alesina and Massimo Galuzzi. A new proof of Vincent's theorem. *L'Enseignement Mathémathique*, 44:219–256, 1998.

[Akr78a] A.G. Akritas. A correction on a theorem by Uspensky. *Bull. Soc. Math. Gréce (N.S.)*, 19:278–285, 1978.

[3] I am grateful to Bernard Mourrain for this suggestion.

[Akr78b] A.G. Akritas. *Vincent's theorem in algebraic manipulation*. PhD thesis, Operations Research Program, North Carolina State University, Raleigh, North Carolina, 1978.

[Akr82] A.G. Akritas. Reflections on a pair of theorems by Budan and Fourier. *Mathematics Magazine*, 55(5):292–298, 1982.

[Akr86] A.G. Akritas. There is no "Uspensky's method". In *Proceedings of the 1986 Symposium on Symbolic and Algebraic Computation*, pages 88–90, Waterloo, Ontario, Canada, 1986.

[Akr89] Alkiviadis G. Akritas. *Elements of Computer Algebra with Applications*. John Wiley Interscience, New York, 1989.

[AS05] Alkiviadis G. Akritas and A. Strzébonski. A comparative study of two real root isolation methods. *Nonlinear Analysis:Modelling and Control*, 10(4):297–304, 2005.

[CA76] George E. Collins and Alkiviadis G. Akritas. Polynomial real root isolation using Descartes' rule of signs. In R. D. Jenks, editor, *Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation*, pages 272–275. ACM Press, 1976.

[Dav85] J. H. Davenport. Computer algebra for cylindrical algebraic decomposition. Technical report, The Royal Institute of Technology, Department of Numerical Analysis and Computing Science, S-100 44, Stockholm, Sweden, 1985. Reprinted as: Technical Report 88-10, School of Mathematical Sciences, University of Bath, Claverton Down, Bath BA2 7AY, England.

[DSY05] Zilin Du, Vikram Sharma, and Chee Yap. Amortized bounds for root isolation via Sturm sequences. In Dongming Wang and Lihong Zhi, editors, *Proc. Internat. Workshop on Symbolic-Numeric Computation*, pages 81–93, School of Science, Beihang University, Beijing, China, 2005. Int'l Workshop on Symbolic-Numeric Computation, Xi'an, China, Jul 19–21, 2005.

[Dys47] F. J. Dyson. The approximation to algebraic numbers by rationals. *Acta Math. 79*, 1947.

[EMT06] Ioannis Z. Emiris, Bernard Mourrain, and Elias P. Tsigaridas. Real algebraic numbers: Complexity analysis and experimentations. Research Report 5897, INRIA, April 2006. `http://www.inria.fr/rrrt/rr-5897.html`.

[ESY06] Arno Eigenwillig, Vikram Sharma, and Chee Yap. Almost tight complexity bounds for the Descartes method. In *Proc. Int'l Symp. Symbolic and Algebraic Computation (ISSAC'06)*, 2006. Genova, Italy.

[ET06] Ioannis Z. Emiris and Elias P. Tsigaridas. Univariate polynomial real root isolation: Continued fractions revisited. To appear in ESA 2006. Appeared on CS arxiv, Apr. 2006.

[Hon98] Hoon Hong. Bounds for absolute positiveness of multivariate polynomials. *J. of Symbolic Computation*, 25(5):571–585, 1998.

[JKR05] Jeremy R. Johnson, Werner Krandick, and Anatole D. Ruslanov. Architecture-aware classical Taylor shift by 1. In *Proc. 2005 International Symposium on Symbolic and Algebraic Computation (ISSAC 2005)*, pages 200–207. ACM, 2005.

[Joh98] J.R. Johnson. Algorithms for polynomial real root isolation. In B.F. Caviness and J.R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and monographs in Symbolic Computation, pages 269–299. Springer, 1998.

[Khi97] A. Ya. Khinchin. *Continued Fractions*. Dover Publications, 1997.

[Kio86] J. Kioustelidis. Bounds for the positive roots of the polynomials. *Journal of Computational and Applied Mathematics*, 16:241–244, 1986.

[KM06] Werner Krandick and Kurt Mehlhorn. New bounds for the Descartes method. *J. Symbolic Computation*, 41(1):49–66, 2006.

[Kra95] Werner Krandick. Isolierung reeller Nullstellen von Polynomen. In J. Herzberger, editor, *Wissenschaftliches Rechnen*, pages 105–154. Akademie-Verlag, Berlin, 1995.

[Lio40] J. Liouville. Sur l'irrationalite du nombre e. *J. Math. Pures Appl.*, 1840.

[Mc99] Maurice Mignotte and Doru Ştefănescu. *Polynomials: An Algorithmic Approach*. Springer, Singapore, 1999.

[Mig81] Maurice Mignotte. Some inequalities about univariate polynomials. In *Proc. 1981 ACM Symposium on Symbolic and Algebraic Computation (SYMSAC 1981)*, pages 195–199. ACM, 1981.

[MPTT05] B. Mourrain, J.P. Pavone, E. Tsigaridas, and P. Trébuchet. SYNAPS, a library for symbolic-numeric computation. In 8th Int. Symposium on Effective Methods in Algebraic Geometry, MEGA, Sardinia, Italy, May 2005. Software presentation.

[Ost50] A.M. Ostrowski. Note on Vincent's theorem. *The Annals of Mathematics*, 52(3):702–707, Nov 1950.

[Rot55] K.F. Roth. Rational approximations to algebraic numners. *Mathematika 2*, 1955.

[Sch95] Wolfgang M. Schmidt. The number of exceptional approximations in Roth's theorem. *J. Austral. Math. Soc.*, 59:375–383, 1995.

[Şte05] D. Ştefănescu. New bounds for the positive roots of polynomials. *Journal of Universal Computer Science*, 11(12):2132–2141, 2005.

[Thu09] A. Thue. Uber Annaherungswerte algebraischer Zahlen. *J. Reine Angew. Math. 135*, 1909.

[Usp48] J. V. Uspensky. *Theory of Equations*. McGraw-Hill, New York, 1948.

[vdS70] A. van der Sluis. Upper bounds for roots of polynomials. *Numer. Math.*, 15:250–262, 1970.

[Vin36] A.J.H. Vincent. Sur la résolution des équations numériques. *J. Math. Pures Appl.*, 1:341–372, 1836.

[vzGG97] Joachim von zur Gathen and Jürgen Gerhard. Fast algorithms for Taylor shifts and certain difference equations. In *Proc. 1997 International Symposium on Symbolic and Algebraic Computation (ISSAC 1997)*, pages 40–47. ACM, 1997.

[Yap00] Chee K. Yap. *Fundamental Problems of Algorithmic Algebra*. Oxford University Press, 2000.

[YLP+04] C. Yap, C. Li, S. Pion, Z. Du, and V. Sharma. Core library tutorial: a library for robust geometric computation, 1999–2004. Version 1.1 was released in Jan 1999. Latest Version 1.6 (Jun 2003). Download source and documents, `http://cs.nyu.edu/exact/`.

Appendix: Proofs

Lemma 2.3.

$$\frac{U(A)}{4} < S(A) < U(A).$$

Proof. Suppose $S(A) = 2\left(\frac{|a_{n-i}|}{|a_n|}\right)^{1/i}$. Let $p := \lfloor \log |a_{n-i}| \rfloor - \lfloor \log |a_n| \rfloor - 1$, $q = \lfloor p/i \rfloor$ and $r := p - q \cdot i$, $0 \le r < i$. Then we know that

$$2^p < \frac{|a_{n-i}|}{|a_n|} < 2^{p+2}.$$

Taking the i-th root we get

$$2^q < \left(\frac{|a_{n-i}|}{|a_n|}\right)^{1/i} < 2^{q+2},$$

since $q \le p/i$ and $(p+2)/i = q + (r+2)/i \le q+2$. But $U(A) = 2^{q+2}$, and hence we get our desired inequality. □

Lemma 3.2. *Let $B_1(X) \in \mathbb{R}[X]$ be a polynomial all of whose roots are in the open half plane $\Re(z) > 0$. For $i > 1$ recursively define*

$$B_i(X) := B_{i-1}(X + \delta_{i-1})$$

where

$$\delta_{i-1} := \begin{cases} \mathrm{PLB}(B_{i-1}) + 1 & \text{if } \mathrm{PLB}(B_{i-1}) > 1 \\ 1 & \text{otherwise.} \end{cases}$$

Let α_1 denote a root of $B_1(X)$ with the smallest absolute value, and recursively let $\alpha_i = \alpha_{i-1} - \delta_{i-1}$. Then $\Re(\alpha_i) \leq 1$ if $i \geq 2 + 8n + \gamma_n \mathrm{logm}\, \Re(\alpha_1)$.

Proof. Let $b_i := \mathrm{PLB}(B_i)$, and β_i be the root of $B_i(X)$ with the smallest absolute value. Note that β_i may not be the same as α_i, except initially. But there is still some relation between the two, namely $\Re(\alpha_i) \leq \Re(\beta_i)$, for $i \geq 1$. The proof is by induction; the base case holds by the definition of α_1.

Suppose inductively $\Re(\alpha_{i-1}) \leq \Re(\beta_{i-1})$. Let β be the root of $B_{i-1}(X)$ such that $\beta_i = \beta - \delta_{i-1}$. Then we know by the definition of β_{i-1} that $|\beta_{i-1}| \leq |\beta|$. However, we also have

$$|\beta_i| = |\beta - \delta_{i-1}| \leq |\beta_{i-1} - \delta_{i-1}|.$$

Thus from Lemma 3.1 we know that $\Re(\beta) \geq \Re(\beta_{i-1})$ and hence

$$\Re(\beta_i) = \Re(\beta) - \delta_{i-1} \geq \Re(\beta_{i-1}) - \delta_{i-1} \geq \Re(\alpha_{i-1}) - \delta_{i-1} = \Re(\alpha_i).$$

Since β_i is the root of $B_i(X)$ with the smallest absolute value, from (5) we know that $\frac{|\beta_i|}{8n} < b_i < |\beta_i|$. Moreover, because $\Re(\beta_i) \geq \Re(\alpha_i)$ we have $b_i > \Re(\alpha_i)/8n$. Let j be the index such that $\Re(\alpha_i) > 8n$ for $i < j$. Then for $i < j$ we know that $b_i > 1$. Thus $\Re(\alpha_i) = \Re(\alpha_{i-1}) - b_{i-1} - 1 < \Re(\alpha_{i-1})(1 - \frac{1}{8n})$ and recursively $\Re(\alpha_i) < \Re(\alpha)(1 - \frac{1}{8n})^{i-1}$. So $\Re(\alpha_j) \leq 8n$ if $\Re(\alpha_{j-1}) \leq 8n$ or if

$$j \geq 2 + \gamma_n \mathrm{logm}\, \Re(\alpha_1).$$

For $i \geq j$ we know that $\Re(\alpha_i) \leq 8n$, because $\Re(\alpha_i)$ is monotonically decreasing. Thus if $i > j$ is such that $i - j \geq 8n$ then $\Re(\alpha_i) \leq 1$. Combining this lower bound on $i - j$ with the lower bound on j we get the result of the lemma. □

Lemma 3.4. *Let $B_1(X) \in \mathbb{R}[X]$, $B_1(0) \neq 0$, and recursively define δ_i, and $B_i(X)$ as in the above lemma. Let $\alpha_1 := LP(B_1)$, $\beta_1 := LN(B_1)$ and recursively define $\alpha_i = \alpha_{i-1} - \delta_{i-1}$ and $\beta_i = \beta_{i-1} - \delta_{i-1}$. If*

$$i = \Omega\left(n + \kappa_n \mathrm{logm}\, \frac{|\alpha_1|}{|\beta_1|} + \kappa_n \mathrm{logm}\, |\alpha_1|\right)$$

then $\Re(\alpha_i) \leq 1$, where

$$\kappa_n := (\log(8n+1) - \log 8n)^{-1}. \tag{28}$$

Proof. Let $b_i := \mathrm{PLB}(B_i)$. We assume that $|\beta_1| < |\alpha_1|$, otherwise the bound in the lemma trivially follows from the previous lemma. Let γ_i, denote the root of $B_i(X)$ with the smallest absolute value; by definition and our assumption that $|\beta_1| < |\alpha_1|$ we initially have $\gamma_1 = \beta_1$. Let j be the first index i such that $\gamma_i \neq \beta_i$. Then for $i > j$, $\Re(\gamma_i) \geq \Re(\alpha_i)$; this follows from the fact that $\alpha_1 = \mathrm{LP}(B_1)$ and from Lemma 3.1. Thus if $i > j$ is such that

$$i - j > 1 + 8n + \kappa_n \mathrm{logm}\, |\alpha_1| > 1 + 8n + \gamma_n \mathrm{logm}\, |\alpha_i|$$

then from Lemma 3.2 we are sure that $\Re(\alpha_i) \leq 1$. But if we choose j such that $|\beta_i| > |\alpha_i|$, for $i > j$, then $\gamma_i \neq \beta_i$ for $i > j$, because all the roots with negative real parts are to the left of β_i and in absolute value greater than $|\beta_i|$. We next give a lower bound on j.

For $i < j$ we have from (5)

$$b_i > \frac{|\beta_i|}{8n}. \tag{29}$$

Assume that $b_i > 1$, then we know that $\delta_i = b_i + 1$. Since $\beta_{i+1} = \beta_i - \delta_i$ it follows that $\Im(\beta_{i+1}) = \Im(\beta_i)$ and hence

$$\begin{aligned} |\beta_{i+1}| = |\beta_{i-1} - (b_i + 1)| = &= ((|\Re(\beta_{i-1})| + b_i + 1)^2 + \Im(\beta_{i-1})^2)^{\frac{1}{2}} \\ &> (|\Re(\beta_{i-1})|^2 + b_i^2 + 1 + \Im(\beta_{i-1})^2)^{\frac{1}{2}} \\ &= (|\beta_{i-1}|^2 + b_i^2 + 1)^{\frac{1}{2}}. \end{aligned}$$

Applying the bound from (29) we get

$$|\beta_{i+1}| > (|\beta_{i-1}|^2(1 + (8n)^{-2}) + 1)^{\frac{1}{2}} > |\beta_{i-1}|(1 + \frac{1}{8n}),$$

because $2 < 8n$ for $n \geq 1$, which is trivially true. Thus recursively we know that $|\beta_{i+1}| > |\beta_1|(1 + 1/8n)^i$. Hence if

$$j > 1 + 8n + \kappa_n \log \frac{|\alpha_1|}{|\beta_1|} \tag{30}$$

then $|\beta_i| > |\alpha_1| \geq |\alpha_i|$, for $i > j$. From (29) it is clear that we need $8n$ shifts initially to ensure $b_i > 1$. These additional shifts, along with (30) and the bound on $i - j$ above give us the desired lower bound on i which ensures that $\Re(\alpha_i) \leq 1$. □

Upper bound on $|\eta_i|$. Recall from Definition 3.5 that $\eta_i = M_i^{-1}(\alpha_u)$. Thus from (8) we get

$$\eta_i = \frac{Q_i + Q_{i-1}}{Q_i} \frac{|s_i - \alpha_u|}{|r_i - \alpha_u|} \leq \frac{Q_i + Q_{i-1}}{Q_i} \frac{|s_i - \alpha_u|}{C(A_{\text{in}}, N)} Q_i^N,$$

where the second inequality follows from (19). But

$$|s_i - \alpha_u| \leq \sqrt{3}|J_i| < 2(Q_i Q_{i-1})^{-1}.$$

Thus

$$\eta_i < 2C(A_{\text{in}}, N)^{-1} Q_i^N.$$

Moreover, from (15) we know that $Q_i \leq 2|\alpha_u - \beta_u|^{-1}$. Plugging this bound on Q_i into the bound on η_i we obtain

$$\eta_i < 2^{N+1}|\alpha_u - \beta_u|^{-N} C(A_{\text{in}}, N)^{-1}.$$

Taking logarithm on both sides we get

$$\log \eta_i \leq -N \log |\alpha_u - \beta_u| - \log C(A_{\text{in}}, N) + N + 1.$$

A lower bound on $|\mathbf{LN}(B_{1+i_{\ell-1}})|$. We may safely assume that $|\mathrm{LN}(B_{1+i_{\ell-1}})| \neq 0$ since if zero is a root of $B_{1+i_{\ell-1}}(X)$ then in the procedure $\mathrm{CF}(A, M)$ we always divide the polynomial by X and remove this degenerate case. We derive lower bounds for two cases: first, when the root $\mathrm{LN}(B_{1+i_{\ell-1}})$ corresponds to a root of $A_i(X)$ in $\Re(z) > 0$, and second when $\mathrm{LN}(B_{1+i_{\ell-1}})$ corresponds to a root of $A_i(X)$ in $\Re(z) \leq 0$. Let γ be the root of $A_{\mathrm{in}}(X)$ that corresponds to $\mathrm{LN}(B_{1+i_{\ell-1}})$. Then the first case is equivalent to saying that $\gamma \in C_{J_i}$ and the second to the condition that $\gamma \notin C_{J_i}$. We derive bounds on $|\mathrm{LN}(B_{1+i_{\ell-1}})|$ under these two conditions, starting with the first.

1. In this case the polynomial $B_{1+i_{\ell-1}}(X) = A_i(X + \delta)$, where δ is defined as

$$\delta = 1 + \sum_{j=1}^{i_{\ell-1}} 1 + \begin{cases} \mathrm{PLB}(B_j) & \text{if } \mathrm{PLB}(B_j) > 1 \\ 0 & \text{otherwise;} \end{cases} \tag{31}$$

note that δ is a natural number since $\mathrm{PLB}(B_j)$ is a natural number if it is greater than one. The transformation

$$M'(X) := \frac{P_i X + P_{i-1} + P_i \delta}{Q_i X + Q_{i-1} + Q_i \delta}$$

describes the bijective correspondence between the roots of $A_{\mathrm{in}}(X)$ and of $B_{1+i_{\ell-1}}(X)$. In particular,

$$\gamma = M'(\mathrm{LN}(B_{1+i_{\ell-1}}))$$

and hence

$$\begin{aligned} |\mathrm{LN}(B_{1+i_{\ell-1}})| &= |M'^{-1}(\gamma)| \\ &= \left| \frac{P_{i-1} + P_i\delta - (Q_{i-1} + Q_i\delta)\gamma}{P_i - Q_i\gamma} \right| \\ &= \frac{\delta Q_i + Q_{i-1}}{|P_i - Q_i\gamma|} \left| \gamma - \frac{\delta P_i + P_{i-1}}{\delta Q_i + Q_{i-1}} \right| \end{aligned}$$

(observe that $\frac{\delta P_i + P_{i-1}}{\delta Q_i + Q_{i-1}} = M'(0)$). From (19) we get

$$|\mathrm{LN}(B_{1+i_{\ell-1}})| \geq \frac{C(A_{\mathrm{in}}, N)}{|P_i - Q_i\gamma|} (\delta Q_i + Q_{i-1})^{-(N-1)}.$$

Since $\delta Q_i \geq Q_{i-1}$ we further get

$$|\mathrm{LN}(B_{1+i_{\ell-1}})| \geq \frac{C(A_{\mathrm{in}}, N)}{|P_i - Q_i\gamma|} (2\delta Q_i)^{-(N-1)} \geq C(A_{\mathrm{in}}, N) 2^{-N} (\delta Q_i)^{-(N-1)},$$

where the last step follows from the fact that since $\gamma \in C_{J_i}$, $|P_i - Q_i\gamma| \leq (Q_i Q_{i-1})^{-1} \leq 1$. But $\delta \leq q_{i+1} < Q_{i+1}$, for $i < m$, and for $i = m$, $\delta \leq \delta_v$, where δ_v is defined as in (10); along with (15) and (11) it follows that $\delta, Q_i < 2|\alpha_u - \beta_u|^{-1}$. Thus

$$-\log |\mathrm{LN}(B_{1+i_{\ell-1}})| = O(-N \log |\alpha_u - \beta_u| - \log C(A_{\mathrm{in}}, N)). \tag{32}$$

2. If $\mathrm{LN}(B_{1+i_{\ell-1}})$ corresponds to a negative root of $A_i(X)$ then from Lemma 3.1 we know that $\mathrm{LN}(B_j)$, $j = 1, \ldots, 1+i_{\ell-1}$, correspond to the same negative root of $A_i(X)$. Thus we derive a lower bound on $|\mathrm{LN}(B_1)| = |\mathrm{LN}(A_i)|$. From (8) we know that

$$|\mathrm{LN}(A_i)| = \frac{Q_i + Q_{i-1}}{Q_i} \frac{|s_i - \gamma|}{|r_i - \gamma|} \geq \frac{1}{Q_i |r_i - \gamma|} C(A_{\mathrm{in}}, N)(Q_i + Q_{i-1})^{1-N},$$

where the last step follows by applying (19) to $|s_i - \gamma|$. Since γ is outside C_{J_i} and $\alpha_u \in \overline{C}_{J_i} \cup \underline{C}_{J_i}$, we have

$$|r_i - \gamma| \leq |\gamma - \alpha_u| + |\alpha_u - r_i| \leq |\gamma - \alpha_u| + 2(Q_i Q_{i-1})^{-1}.$$

Thus

$$\begin{aligned} |\mathrm{LN}(A_i)| &\geq \frac{Q_{i-1}}{Q_i |\gamma - \alpha_u| + 2Q_{i-1}^{-1}} C(A_{\mathrm{in}}, N) Q_{i-1}^{-N} \\ &\geq \frac{1}{Q_i (2 + |\gamma - \alpha_u|)} C(A_{\mathrm{in}}, N) Q_{i-1}^{-N} \\ &\geq \frac{1}{2 + |\gamma - \alpha_u|} C(A_{\mathrm{in}}, N)(Q_i Q_{i-1})^{-N}. \end{aligned}$$

From (11), and the fact that $I_v \subseteq J_i$, we know that $Q_i Q_{i-1} |\alpha_u - \beta_u| \leq 2$. Thus

$$|\mathrm{LN}(A_i)| \geq \frac{1}{1 + |\gamma - \alpha_u|} C(A_{\mathrm{in}}, N)(2|\alpha_u - \beta_u|)^N.$$

But from the definition of $\mu(A_{\mathrm{in}})$ we know that $|\gamma - \alpha_u| \leq 2\mu(A_{\mathrm{in}})$, and hence we have

$$|\mathrm{LN}(A_i)| \geq \frac{1}{2 + 2\mu(A_{\mathrm{in}})} C(A_{\mathrm{in}}, N)(2|\alpha_u - \beta_u|)^N$$

from which we obtain

$$-\log |\mathrm{LN}(A_i)| = O(-N \log |\alpha_u - \beta_u| - \log C(A_{\mathrm{in}}, N) + \mathrm{logm}\, \mu(A_{\mathrm{in}})). \tag{33}$$

From (32) and (33) we may safely conclude that

$$-\log |\mathrm{LN}(B_{1+i_{\ell-1}})| = O(-N \log |\alpha_u - \beta_u| - \log C(A_{\mathrm{in}}, N) + \log \mu(A_{\mathrm{in}})). \tag{34}$$

Lemma 3.9. *Let α be a root of an integer polynomial $A(X)$ of degree n. Suppose $P/Q \in \mathbb{Q}$, $Q > 0$, is such that $0 < |\alpha - P/Q| \leq 1$ and $A(P/Q) \neq 0$, then $|\alpha - P/Q| \geq C(\alpha) \cdot Q^{-n}$ where*

$$C(\alpha) \geq 2^{-n - \log n - (n+1)\log \|A\|_\infty}. \tag{35}$$

Proof. From the mean value theorem we know that

$$|A(\alpha) - A(P/Q)| = |A'(\beta)||\alpha - P/Q|,$$

where $\beta = (1-t)\alpha + tP/Q$, $0 \le t \le 1$. But $|A(P/Q)| \ge Q^{-n}$, so

$$|\alpha - P/Q| = \left|\frac{A(P/Q)}{A'(\beta)}\right| \ge |A'(\beta)|^{-1} Q^{-n}.$$

Since $|\alpha - P/Q| \le 1$ we know that $|\beta| \le 1 + |\alpha|$. and hence it can be showed that $|A'(\beta)|$ is smaller than $n\|A\|_\infty (1+|\alpha|)^n$. Using Cauchy's upper bound [Yap00, Cor. 6.8,p. 149] on $|\alpha|$ we get the bound on the constant $C(\alpha)$ mentioned in the lemma. □