University of California

Los Angeles

Lower Bounds in Arithmetic Complexity via Asymmetric Embeddings

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mathematics

by

Joseph Emil Busch

2008

The dissertation of Joseph Emil Busch is approved.

Matthias Aschenbrenner

Miloš Ercegovac

Donald A. Martin

Yiannis N. Moschovakis, Committee Chair

University of California, Los Angeles

2008

To my parents

Table of Contents

1 Introduction **1**

1.1 Recursion on partial algebras 1

1.2 Basic parallel complexity 3

1.3 Fundamental notions and results 4

2 Quadratic residuosity and pseudoprimality **6**

2.1 Introduction 6

2.2 An asymmetric embedding 7

2.3 Legendre and Jacobi symbols 10

2.4 Pseudoprimality and modular exponentiation 14

2.5 Lower bounds for non-uniform algorithms 16

2.6 Extending **Z** by finitely many Presburger primitives 16

3 Notions in imaginary norm-Euclidean quadratic integer rings . **19**

3.1 Introduction 19

3.2 Results from algebraic number theory 20

3.3 Imaginary norm-Euclidean quadratic algebras 24

3.4 Primality and squarefreeness 25

3.5 Coprimality and great common divisor 35

4 Lower bounds for PCF **43**

4.1 Logical extensions 43

4.2 Scott-Plotkin PCF 45

4.2.1 Axioms and rules 45

4.2.2 Computation 46

5 Known algorithms and related bounds 52

5.1 Other bounds for quadratic residuosity and pseudoprimality . . . 52

5.2 The binary algorithm for the Jacobi symbol 53

5.3 Binary-like gcd algorithms in imaginary norm-Euclidean quadratic integer rings 56

5.4 Comparison with Boolean circuit complexity 59

References 60

Acknowledgments

I sincerely thank both my parents, I dedicate this dissertation to them in appreciation of their unbounded support and encouragement.

I thank my friends from the UCLA Mathematics Department, especially Steve Chan and Don Larson, for many hours of enjoyable conversation about mathematics and other interesting subjects. To Becky Hickok, thank you so much for your patience, understanding, and for being such a good listener.

Finally, I am deeply grateful for the privelege of having Professor Yiannis Moschovakis as my advisor. It is impossible for me to imagine an advisor who is more patient in teaching, more generous with time, or more willing to spring to the blackboard to present a detailed proof that would aid in my understanding. It was an honor to work with him, and I learned something new each time we spoke.

Vita

1975	Born, Rahway, New Jersey, USA.
1997	B.A., Mathematics, Boston University.
1997–2001	.com boom participant.
2002	M.S., Mathematics, New York University.
2002	National Science Foundation VIGRE Fellowship awarded.
2004	M.A., Mathematics, UCLA.
2006	C.Phil., Mathematics, UCLA.

Publications

On the optimality of the binary algorithm for the Jacobi symbol. Fundamenta Informaticae, 76 (1–2), pp. 1–11, 2007.

Abstract of the Dissertation

Lower Bounds in Arithmetic Complexity via Asymmetric Embeddings

by

Joseph Emil Busch

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2008

Professor Yiannis N. Moschovakis, Chair

In this dissertation, logical methods are used to prove lower bounds for the complexity of computing certain number-theoretic and algebraic functions. The computations are assumed to be relative to a fixed basis of given operations, and lower bounds are proved for a time complexity which measures the number of calls to the primitives. In one case, as a corollary to a lower bound, the optimality of a known algorithm is established.

The lower bounds are obtained through the construction of certain parametric, asymmetric embeddings. In Chapter 2, an embedding is defined on an algebra of (rational) integers. Embeddings which are similar in spirit to this one are constructed in Chapter 3 on algebras of each of the imaginary norm-Euclidean quadratic integer rings, including the rings of Gaussian and Eisenstein integers. A key issue in the definition of such embeddings involves the interplay of the metric and multiplicative structure of the underlying algebra.

The lower bounds are proved for recursive programs, where parameters are assumed to be passed by value. In Chapter 4, it is shown how the same lower bounds hold a version of call-by-name PCF, a simply-typed λ-calculus with re-

cursion operators at each type.

In Chapter 5, results that are related to those of the dissertation are discussed. In addition, relevant known algorithms are presented. A unifying feature of this work is that algorithms for computing many of the functions for which lower bounds are obtained are similar in spirit to the so-called “binary algorithm” of Stein [Ste67], and, like Stein’s algorithm, rely only on piecewise-linear primitive operations.

CHAPTER 1

Introduction

The lower bounds that we prove in the sequel are established in the framework developed by Moschovakis and van den Dries in [DM04]. In this context, algorithms relative to given operations are formalized as recursive programs defined on partial algebras, structures which encapsulate the domain of computation and the primitives.

1.1 Recursion on partial algebras

We consider algorithms expressed as McCarthy-style recursive programs [McC63] on partial algebras of the form

$$\mathbf{A} = (A, 0, 1, \{\phi^{\mathbf{A}}\}_{\phi \in \Phi}). \tag{1.1}$$

Here, A is a non-empty set, $0, 1 \in A$ are distinct, and the signature Φ associates to each formal symbol $\phi \in \Phi$, a partial function $\phi^{\mathbf{A}} : A^{n_\phi} \rightharpoonup A$ of arity n_ϕ. To emphasize the signature Φ, we sometimes refer to $\mathbf{A}$ as in (1.1) as a partial Φ-algebra. If X is any subset of A containing $0, 1$, then the restricted subalgebra

$$\mathbf{A} \upharpoonright X = (X, 0, 1, \{\phi^{\mathbf{A}} \upharpoonright X\}_{\phi \in \Phi}), \tag{1.2}$$

on the same signature Φ, is defined so that each of its partial functions $\phi^{\mathbf{A}} \upharpoonright X$ satisfies

$$(\phi^{\mathbf{A}} \upharpoonright X)(\vec{x}) = w \iff \vec{x}, w \in X \ \&\ \phi^{\mathbf{A}}(\vec{x}) = w. \tag{1.3}$$

Φ-terms E are generated by the recursion

$$E ::= \mathsf{0} \mid \mathsf{1} \mid \mathsf{v} \mid \phi(E_1, \ldots, E_{n_\phi}) \mid \zeta(E_1, \ldots, E_{n_\zeta}) \mid \\ (\mathsf{if}\ (E_1 = \mathsf{0})\ \mathsf{then}\ E_2\ \mathsf{else}\ E_3), \tag{1.4}$$

where v is an individual variable, $\phi \in \Phi$ has arity n_ϕ, and ζ is a function variable with arity n_ζ. A recursive program α on $\mathbf{A}$ is a system of mutually recursive Φ-term equations

$$\alpha : \begin{cases} \zeta_0(\vec{\mathsf{v}}_0) &= E_0(\vec{\mathsf{v}}_0, \zeta_1, \ldots, \zeta_k), \\ \zeta_1(\vec{\mathsf{v}}_1) &= E_1(\vec{\mathsf{v}}_1, \zeta_1, \ldots, \zeta_k), \\ & \vdots \\ \zeta_k(\vec{\mathsf{v}}_k) &= E_k(\vec{\mathsf{v}}_k, \zeta_1, \ldots, \zeta_k), \end{cases} \tag{1.5}$$

where the variables which occur in the terms are among those displayed. Corresponding to each term $E(\vec{\mathsf{v}}, \vec{\zeta})$ is the functional on A

$$(\vec{x}, \vec{f}) \mapsto \mathrm{den}^{\mathbf{A}}(E\{\vec{\mathsf{v}} := \vec{x}, \vec{\zeta} := \vec{f}\}), \tag{1.6}$$

which takes a tuple of individuals and partial functions on A to the denotation in $\mathbf{A}$ of E under the binding of the variables to these objects. Each such functional is monotone and continuous, and therefore, by a classical result, the system of functionals associated to (1.5) has a $\sqsubseteq$-least solution tuple $(\overline{\zeta}_0, \ldots, \overline{\zeta}_k)$. The partial function computed by α in $\mathbf{A}$ is $\overline{\alpha}^{\mathbf{A}}(\vec{x}) = \overline{\zeta}_0(\vec{x})$, the least fixed point associated with the head term.

It will be convenient to use the model-theoretic notation

$$\mathbf{A} \models \alpha(\vec{x}) = w \Longleftrightarrow \overline{\alpha}^{\mathbf{A}}(\vec{x}) = w. \tag{1.7}$$

1.2 Basic parallel complexity

During the course of its execution, a recursive program may perform logical operations, and may also call a given function $\phi^{\mathbf{A}}$, which is regarded as an oracle, and which returns the correct value in one step. If α computes the partial function $\overline{\alpha}$ on a partial Φ-algebra $\mathbf{A}$, and $\overline{\alpha}$ converges on input $\vec{x}$, then the basic parallel complexity $c_\alpha^{\mathbf{A}}(\vec{x})$ (or simply $c_\alpha(\vec{x})$ if the underlying algebra is clear) is the maximum number of nested calls to the given functions $\{\phi^{\mathbf{A}}\}_{\phi\in\Phi}$ made by α in the computation of $\overline{\alpha}^{\mathbf{A}}(\vec{x})$. In more detail, for each term M (possibly with parameters) in the signature of α, the basic parallel complexity $C(M)$ is determined by (C1)–(C4) below. In each of these clauses, the parameters and function variables of a term are suppressed for brevity.

(C1) $C(\mathsf{0}) = C(\mathsf{1}) = C(x) = 0$, if x is any parameter,

(C2) $C(\phi(M_1, \ldots, M_n)) = \max\{C(M_i)\} + 1$, if $\phi \in \Phi$,

(C3) $C(\zeta(M_1, \ldots, M_n)) = \max\{C(M_i)\} + C(E_\zeta(\overline{M}_1, \cdots, \overline{M}_n))$, if ζ is a recursion variable and E_ζ defines ζ in the program α,

(C4) $C(\mathsf{if}\ (M_0 = 0)\ \mathsf{then}\ M_1\ \mathsf{else}\ M_2)$

$$= \begin{cases} \max\{C(M_0), C(M_1)\} & \text{if } \overline{M}_0 = 0, \\ \max\{C(M_0), C(M_2)\} & \text{if } \overline{M}_0 \downarrow\neq 0. \end{cases}$$

The basic parallel complexity of α is the complexity of the head term:

$$c_\alpha(\vec{x}) = C(E_0(\vec{x}, \zeta_1, \ldots, \zeta_k)), \tag{1.8}$$

for α as in (1.5).

Note that c_α does not count logical steps. It is an abstract measure of strict, parallel, call-by-value time complexity which is dominated by many other common measures, e.g., it is no larger than the total number of calls to the given operations in any sequential or parallel implementation. We extend the model-theoretic notation of (1.7) as follows:

$$\mathbf{A} \models \alpha^{(m)}(\vec{x}) = w \iff \overline{\alpha}^{\mathbf{A}}(\vec{x}) = w \text{ and } c_{\alpha}^{\mathbf{A}}(\vec{x}) \leq m. \tag{1.9}$$

1.3 Fundamental notions and results

Given a partial algebra $\mathbf{A} = (A, 0, 1, \{\phi^{\mathbf{A}}\}_{\phi \in \Phi})$ and $X \subseteq A$, define by recursion

$$\begin{aligned} G_0^{\mathbf{A}}(X) &= \{0, 1\} \cup X, \qquad (1.10) \\ G_{m+1}^{\mathbf{A}}(X) &= G_m^{\mathbf{A}}(X) \cup \{\phi^{\mathbf{A}}(\vec{x}) \mid \vec{x} \in G_m^{\mathbf{A}}(X) \ \&\ \phi \in \Phi \\ &\qquad \&\ \phi^{\mathbf{A}}(\vec{x}) \downarrow\}. \end{aligned}$$

Thus, the elements of $G_m^{\mathbf{A}}(X)$ are precisely the values that are generated in $\mathbf{A}$ from X in m steps.

For partial algebras $\mathbf{A} = (A, 0^{\mathbf{A}}, 1^{\mathbf{A}}, \{\phi^{\mathbf{A}}\}_{\phi \in \Phi})$ and $\mathbf{B} = (B, 0^{\mathbf{B}}, 1^{\mathbf{B}}, \{\phi^{\mathbf{B}}\}_{\phi \in \Phi})$, an embedding π from $\mathbf{A}$ to $\mathbf{B}$ ($\pi : \mathbf{A} \hookrightarrow \mathbf{B}$) is any injective $\pi : A \rightarrowtail B$ such that $\pi(0^{\mathbf{A}}) = 0^{\mathbf{B}}$, $\pi(1^{\mathbf{A}}) = 1^{\mathbf{B}}$, and for all $\vec{x}, w \in A$ and all $\phi \in \Phi$,

$$\phi^{\mathbf{A}}(\vec{x}) = w \Longrightarrow \phi^{\mathbf{B}}(\pi(\vec{x})) = \pi(w), \tag{1.11}$$

It follows by the Embedding Lemma of [DM04] that if $\pi : \mathbf{A} \hookrightarrow \mathbf{B}$ is an embedding, and α is a recursive program on a partial algebra with signature Φ, then

$$\mathbf{A} \models \alpha^{(m)}(\vec{x}) = w \Longrightarrow \mathbf{B} \models \alpha^{(m)}(\pi(\vec{x})) = \pi(w). \tag{1.12}$$

This result together with the following will be our key technical tools for establishing lower bounds.

Lemma 1.3.1 (Absoluteness Lemma [DM04]). *Let α be a recursive program on a partial algebra $\boldsymbol{A}$ such that $\boldsymbol{A} \models \overline{\alpha}^{(m)}(\vec{x}) = w$. Then*

$$w \in G_m^{\boldsymbol{A}}(\vec{x}) \tag{1.13}$$

and

$$\boldsymbol{A} \restriction G_m^{\boldsymbol{A}}(\vec{x}) \models \overline{\alpha}^{(m)}(\vec{x}) = w. \tag{1.14}$$

The Absoluteness Lemma expresses the fact that convegergent computations of recursive programs on partial algebras are finite objects which "take place" in the subalgebra generated by the input.

CHAPTER 2

Quadratic residuosity and pseudoprimality

2.1 Introduction

In this Chapter, we consider the minimum complexity of computing the following number-theoretic functions and relations: (i) the Legendre and Jacobi symbols, (ii) pseudoprimality, and (iii) modular exponentiation. The computations are assumed to be relative to the set

$$\boldsymbol{Lin}_0 = \{+, -, 2 \cdot x, \mathrm{quo}_2(x), \mathrm{parity}, \chi_<, \chi_=\} \tag{2.1}$$

of given, primitive operations, where $+$ is addition, $-$ is subtraction, $2 \cdot x$ is the product of 2 with x, $\mathrm{quo}_2(x)$ is the integer quotient in the division of x by 2, $\mathrm{parity}(x)$ is the non-negative remainder in the division of x by 2, and $\chi_<$ and $\chi_=$ are the characteristic functions of the binary relations $<$ and $=$, respectively.

$\boldsymbol{Lin}_0$ is a natural, finite subset of $\boldsymbol{Lin}$, the set of piecewise-linear partial functions that are definable in Presburger arithmetic. The lower bounds that we obtain in this Chapter for recursive programs relative to $\boldsymbol{Lin}_0$ also hold, with different multiplicative constants, for programs which have primitives from larger subsets of $\boldsymbol{Lin}$. This is discussed in more detail in Section 2.6. In this Chapter, we consider programs relative to $\boldsymbol{Lin}_0$ both for concreteness, and because the binary algorithm of Shallit and Sorenson [SS93] that computes the Jacobi symbol is naturally expressed with the operations of $\boldsymbol{Lin}_0$ as primitive. This algorithm is discussed in further detail in Chapter 5.

Let $\mathbf{Z} = (\mathbb{Z}, 0, 1, \boldsymbol{Lin}_0)$. We aim to establish lower bounds on the number of calls to the functions in $\boldsymbol{Lin}_0$ in the computation of (i)–(iii) by a recursive program on $\mathbf{Z}$. In Section 2.2, we establish the existence of a particular embedding which is parameterized by a single natural number. We use this embedding, with suitably chosen values for the parameter, to obtain lower bounds for (i)–(iii) in Sections 2.3 and 2.4.

2.2 An asymmetric embedding

For $\vec{a} \in \mathbb{Z}^n$, let

$$B_m^{\mathbf{Z}}(\vec{a}) = \left\{ \frac{c_0 + c_1 a_1 + \cdots + c_n a_n}{2^m} \in \mathbb{Z} \mid c_i \in \mathbb{Z}, |c_i| \leq 2^{2m} \right\}. \tag{2.2}$$

Lemma 2.2.1. *For each $m \in \mathbb{N}$, $G_m^{\mathbf{Z}}(\vec{a}) \subseteq B_m^{\mathbf{Z}}(\vec{a})$.*

Proof. This follows by a straightforward induction on m. □

Lemma 2.2.2. *Let $p > 1$, $q \geq p$, and*

$$a = 2^{\lfloor \log_2 p \rfloor + 1} q. \tag{2.3}$$

Let $x, y, z \in \mathbb{Z}$, $|x|, |y|, |z| < \frac{p}{2}$, $\lambda \geq 1$. Then

$$\begin{aligned} &x + yp + \lambda z a > 0 \\ &\qquad \iff [z > 0] \vee [z = 0 \ \& \ y > 0] \vee [z = y = 0 \ \& \ x > 0]. \end{aligned} \tag{2.4}$$

Proof. Suppose first that $z = 0$. Assume that $x + yp > 0$. If $y = 0$, then $x = x + yp > 0$, and if $y < 0$, then $p \leq |y|p = -yp < x$, contrary to assumption. Conversely, assume that $y > 0$. Then $yp \geq p > 2|x| > |x|$, and so $x + yp > 0$. If $y = 0$ and $x > 0$, then $x + yp = x > 0$.

Now assume that $z \neq 0$. Since

$$a = 2^{\lfloor \log_2 p \rfloor + 1} q > 2^{\log_2 p} q = pq \geq p^2, \tag{2.5}$$

we have that

$$|x + yp| \leq \frac{p}{2} + \frac{p^2}{2} < p^2 \leq |z| p^2 < |z| a \leq \lambda |z| a, \tag{2.6}$$

and so $x + yp + \lambda z a$ has the same sign as z. □

Lemma 2.2.3. *Let $p > 1$, $q \geq p$, and*

$$a = 2^{\lfloor \log_2 p \rfloor + 1} q. \tag{2.7}$$

Suppose that $\lambda \geq 1$, $x_i, y_i \in \mathbb{Z}$ and $|x_i|, |y_i| < \frac{p}{4}$ for $i = 0, 1, 2$. Then

$$\begin{aligned} x_0 + x_1 p + \lambda x_2 a = y_0 + y_1 p + \lambda y_2 a& \\ \iff x_i = y_i \text{ for } i = 0, 1, 2,& \end{aligned} \tag{2.8}$$

and

$$\begin{aligned} &x_0 + x_1 p + \lambda x_2 a > y_0 + y_1 p + \lambda y_2 a \iff \\ &\quad [x_2 > y_2] \vee [x_2 = y_2 \ \& \ x_1 > y_1] \vee [x_2 = y_2 \ \& \ x_1 = y_1 \ \& \ x_0 > y_0]. \end{aligned} \tag{2.9}$$

Proof. Immediate, by Lemma 2.2.2. □

In the next result, we assume that we are given natural numbers p and q which are sufficiently large compared to m. We then use the metric and multiplicative properties of $a = 2^{\lfloor \log_2 p \rfloor + 1} q$ established in the preceding two Lemmas to embed $\mathbf{Z} \restriction G_m^{\mathbf{Z}}(p, a)$ in $\mathbf{Z}$.

Lemma 2.2.4. *If $2^{2m+3} < p$, $q \geq p$,*

$$a = 2^{\lfloor \log_2 p \rfloor + 1} q, \tag{2.10}$$

and $\lambda \geq 1$, then there exists an embedding $\pi : \boldsymbol{Z} \restriction G_m^{\mathbf{Z}}(p, a) \hookrightarrow \boldsymbol{Z}$ given by

$$\pi\left(\frac{x_0 + x_1 p + x_2 a}{2^m}\right) = \frac{x_0 + x_1 p + \lambda x_2 a}{2^m}. \tag{2.11}$$

In particular,

$$\pi(p) = p, \pi(a) = \lambda a. \tag{2.12}$$

Proof. By Lemma 2.2.1, each $x \in G_m^{\mathbf{Z}}(p, a)$ may be expressed

$$x = \frac{x_0 + x_1 p + x_2 a}{2^m}, \tag{2.13}$$

such that for each $i = 0, 1, 2$,

$$|x_i| \leq 2^{2m} < \frac{2^{2m+3}}{4} < \frac{p}{4}. \tag{2.14}$$

Thus, Lemma 2.2.3 implies that π is a well-defined, order-preserving injection on the set $G_m^{\mathbf{Z}}(p, a)$. To see that range$(\pi) \subseteq \mathbb{Z}$, observe that $\lfloor \log_2 p \rfloor \geq 2m + 3$, and therefore $2^{2m+3} \mid a$. Now,

$$2^m \mid x_0 + x_1 p + x_2 a, \tag{2.15}$$

since the members of $B_m^{\mathbf{Z}}(p, a)$ are integers, and so

$$2^m \mid (x_0 + x_1 p + x_2 a) + (\lambda - 1) x_2 a = x_0 + x_1 p + \lambda x_2 a. \tag{2.16}$$

Clearly, $\pi(0) = \pi(0/2^m) = 0$, and $\pi(1) = \pi(2^m/2^m) = 1$.

The proof will be complete once we show that (1.11) holds for each of the functions in $\boldsymbol{Lin}_0$. We treat the case for $\text{quo}_2(x)$, and verification of the other cases is similar.

Suppose that $x, y = \text{quo}_2(x) \in G_m^{\mathbf{Z}}(p, a)$. Let

$$x = \frac{x_0 + x_1 p + x_2 a}{2^m}, \quad y = \frac{y_0 + y_1 p + y_2 a}{2^m}, \tag{2.17}$$

with $|x_i|, |y_i| \leq 2^{2m}$. If x is odd, then

$$\mathrm{quo}_2(x) = \frac{1}{2}\left(\frac{x_0 + x_1 p + x_2 a}{2^m} - 1\right) = \frac{x_0 - 2^m + x_1 p + x_2 a}{2^{m+1}} = y. \tag{2.18}$$

Since

$$\begin{aligned} |x_0 - 2^m| &\leq 2^{2m} + 2^m \\ &\leq 2^{2m+1} \\ &= \frac{2^{2m+3}}{4} \\ &< \frac{p}{4}, \end{aligned} \tag{2.19}$$

and $2|y_i| \leq 2^{2m+1} < p/4$, we see that

$$2y_0 = x_0 - 2^m, \tag{2.20}$$

$$2y_1 = x_1, 2y_2 = x_2. \tag{2.21}$$

Now, using that $\pi(x)$ is also odd,

$$\begin{aligned} \mathrm{quo}_2(\pi(x)) &= \frac{x_0 - 2^m + x_1 p + \lambda x_2 a}{2^{m+1}} \\ &= \frac{2y_0 + 2y_1 p + 2\lambda y_2 a}{2^{m+1}} \\ &= \pi(y). \end{aligned} \tag{2.22}$$

If x is even, then a similar argument yields the desired result. □

2.3 Legendre and Jacobi symbols

Let $a, m \in \mathbb{Z}$. If there is an $x \in \mathbb{Z}$ such that

$$x^2 \equiv a \pmod{m}, \tag{2.23}$$

then a is said to be a quadratic residue modulo m. If p is an odd prime, then the Legendre symbol (a/p) is

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is a quadratic residue modulo } p \text{ and } p \nmid a, \\ 0 & \text{if } p \mid a, \\ -1 & \text{otherwise.} \end{cases} \tag{2.24}$$

If n is a positive, odd number with prime factorization $n = p_1^{k_1} \cdots p_n^{k_n}$, then the Jacobi symbol, also notated (a/n) is defined as the following product of Legendre symbols:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{k_1} \cdots \left(\frac{a}{p_n}\right)^{k_n}. \tag{2.25}$$

The Jacobi symbol has numerous practical applications, e.g., in the Solovay-Strassen probabilistic primality test, and in cryptography. It is not necessary to factor n in order to compute the Jacobi symbol (a/n); a modern, so-called binary algorithm which appears in [SS93] uses quadratic reciprocity to express the computation of the Jacobi symbol as a recursion using $\boldsymbol{Lin}_0$-operations: comparisons of numbers, and the operations of addition, subtraction, multiplication by 2, and division with remainder by 2. This algorithm is presented in full detail in Chapter 5.

The Jacobi symbol satisfies properties (J1)–(J3) below. These properties are used in the proof of the following Theorem, and are also key to the correctness of the binary algorithm for the Jacobi symbol.

(J1) Jacobi's Reciprocity Law: If $a, n > 0$ are odd and coprime, then

$$\left(\frac{a}{n}\right)\left(\frac{n}{a}\right) = (-1)^{\frac{a-1}{2}\frac{n-1}{2}}. \tag{2.26}$$

In particular, $(a/n) = (n/a)$ unless both a and n are $\equiv 3 \pmod 4$.

(J2) Second Supplement to Jacobi's Reciprocity Law: If $n > 0$ is odd, then

$$\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}}. \tag{2.27}$$

In particular, for odd $n > 0$,

$$\left(\frac{2}{n}\right) = \begin{cases} -1 & n \equiv 3, 5 \pmod{8}, \\ 1 & n \equiv 1, 7 \pmod{8}. \end{cases} \tag{2.28}$$

(J3) The Jacobi symbol is multiplicative: If $n > 0$ is odd, then

$$\left(\frac{aa'}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{a'}{n}\right). \tag{2.29}$$

Theorem 2.3.1. *There is an infinite sequence $\{(a_k, p_k)\}_{k\in\mathbb{N}}$, such that if α is a recursive program on $\boldsymbol{Z}$ such that for all $(a, p) \in \mathbb{Z}^2$ with p an odd prime,*

$$\boldsymbol{Z} \models \alpha(a, p) = \left(\frac{a}{p}\right), \tag{2.30}$$

where $(\frac{-}{-})$ is the Legendre symbol, then for all $k \in \mathbb{N}$,

(i) p_k is prime,

(ii) $p_k \nmid a_k$, and

(iii) $c_\alpha(a_k, p_k) > \frac{1}{20}\log_2 \max(a_k, p_k)$.

Proof. Since 3 and 8 are relatively prime, we may fix, by Dirichlet's theorem on the prime numbers of an arithmetic progression, an increasing enumeration $\{p_k\}_{k\in\mathbb{N}}$ of primes such that each $p_k \equiv 3 \pmod{8}$. For each k, let q_k be the least prime greater than p_k, and let

$$a_k = 2^{\lfloor \log_2 p_k \rfloor + 1} q_k. \tag{2.31}$$

Clearly, $p_k \nmid a_k$.

Fix $k \in \mathbb{N}$. Let $(a, p) = (a_k, p_k)$ and $m = c_\alpha(a, p)$. Suppose for a contradiction that $2^{2m+3} \leq p$. Then there is an embedding π as defined in Lemma 2.2.4 with $\lambda = 2$. We have

$$\mathbf{Z} \models \alpha^{(m)}(a, p) = \left(\frac{a}{p}\right). \tag{2.32}$$

So by absoluteness,

$$\mathbf{Z} \restriction G_m^{\mathbf{Z}}(a, p) \models \alpha(a, p) = \left(\frac{a}{p}\right). \tag{2.33}$$

By (1.12),

$$\mathbf{Z} \models \alpha(\pi(a), \pi(p)) = \pi \left(\frac{a}{p}\right), \tag{2.34}$$

and $\pi\left(\frac{a}{p}\right) = \left(\frac{a}{p}\right)$ since π fixes $\{-1, 1\}$. Since α computes the Legendre symbol on $\mathbf{Z}$,

$$\mathbf{Z} \models \alpha(\pi(a), \pi(p)) = \left(\frac{\pi(a)}{\pi(p)}\right), \tag{2.35}$$

and since 2 is not a residue modulo p, we have that

$$\left(\frac{\pi(a)}{\pi(p)}\right) = \left(\frac{2a}{p}\right) = \left(\frac{2}{p}\right)\left(\frac{a}{p}\right) = -\left(\frac{a}{p}\right). \tag{2.36}$$

Thus

$$\mathbf{Z} \models \alpha(\pi(a), \pi(p)) = -\left(\frac{a}{p}\right), \tag{2.37}$$

a contradiction. Therefore $2^{2m+3} > p$, and so $2m + 3 > \log_2 p$. Since $-1 \notin G_0(a, p)$, we have that $m > 0$, and so

$$5m \geq 2m + 3 > \log_2 p. \tag{2.38}$$

Hence

$$c_\alpha(a, p) > \frac{1}{5} \log_2 p. \tag{2.39}$$

By a straightforward computation, this yields

$$c_\alpha(a, p) > \frac{1}{20} \log_2 a = \frac{1}{20} \log_2 \max(a, p). \tag{2.40}$$

□

2.4 Pseudoprimality and modular exponentiation

Let $a, b \in \mathbb{Z}$. If $a^{b-1} \equiv 1 \pmod{b}$, then b is said to be pseudoprime to the base a. Define the binary relation $\mathrm{PSP} \subseteq \mathbb{Z}^2$ by

$$\mathrm{PSP}(a,b) = \begin{cases} 1 & \text{if } b \text{ is pseudoprime to the base } a, \\ 0 & \text{otherwise.} \end{cases} \tag{2.41}$$

Theorem 2.4.1. *There is an infinite sequence $\{(a_k, p_k)\}_{k\in\mathbb{N}}$, such that if α is a recursive program on $\mathbf{Z}$ which decides PSP, then for all $k \in \mathbb{N}$,*

(i) p_k is pseudoprime to the base a_k and

(ii) $c_\alpha(a_k, p_k) > \dfrac{1}{5}\log_2 p_k$.

Proof. Let $\{p_k\}_{k\in\mathbb{N}}$ be an increasing enumeration of the odd primes. For each k, let $q_k = p_{k+1}$, and let

$$a_k = 2^{\lfloor \log_2 p_k \rfloor + 1} q_k. \tag{2.42}$$

Fix $k \in \mathbb{N}$. Let $(a, p) = (a_k, p_k)$, and $m = c_\alpha(a, p)$. Suppose for a contradiction that $2^{2m+3} \leq p$.

Since $\gcd(a, p) = 1$, we have by Fermat's Little Theorem that

$$a^{p-1} \equiv 1 \pmod{p}. \tag{2.43}$$

Since α decides PSP,

$$\mathbf{Z} \models \alpha^{(m)}(a, p) = 1, \tag{2.44}$$

and so by absoluteness,

$$\mathbf{Z} \restriction G_m^{\mathbf{Z}}(a, p) \models \alpha(a, p) = 1. \tag{2.45}$$

Let π be given by Lemma 2.2.4, with $\lambda = p$. Since π is an embedding, we obtain

$$\mathbf{Z} \models \alpha(\pi(a), \pi(p)) = 1. \tag{2.46}$$

Since $p \mid \lambda$, we have that

$$(\lambda a)^{p-1} \equiv 0 \text{ (mod } p), \tag{2.47}$$

and hence $\neg\mathrm{PSP}(\lambda a, p)$. Therefore,

$$\mathbf{Z} \models \alpha(\pi(a), \pi(p)) = 0, \tag{2.48}$$

a contradiction. Thus $c_\alpha(a,p) > \frac{1}{5}\log_2 p$ as in (2.39). □

Corollary 2.4.1. *There is an infinite sequence $\{(b_k, c_k, m_k)\}_{k\in\mathbb{N}}$, such that if α is a recursive program on $\boldsymbol{Z}$ which computes*

$$f(b,c,m) = b^c \ (mod\ m), \tag{2.49}$$

then for all $k \in \mathbb{N}$,

$$c_\alpha(b_k, c_k, m_k) > \frac{1}{10}\log_2 m_k. \tag{2.50}$$

Proof. Let ζ_γ be a fresh function variable, and define a recursive program γ by adding to α the new head equation

$$\zeta_\gamma(\mathsf{x}_0, \mathsf{x}_1) = (\mathsf{if}\ (\zeta_\alpha(\mathsf{x}_0, \mathsf{x}_1 - 1, \mathsf{x}_1) = 1)\ \mathsf{then}\ 1\ \mathsf{else}\ 0). \tag{2.51}$$

Then γ is a binary recursive program which decides PSP, and the $\boldsymbol{Lin}_0$-calls to $\chi_=$ and $-$ in the head contribute 2 to the complexity, so

$$c_\gamma(x,y) = 2 + c_\alpha(x, y-1, y). \tag{2.52}$$

Let $\{(a_k, p_k)\}_{k\in\mathbb{N}}$ be given by Theorem 2.4.1, and for each k, let

$$(b_k, c_k, m_k) = (a_k, p_k - 1, p_k). \tag{2.53}$$

Then

$$c_\alpha(b_k, c_k, m_k) > \frac{1}{5}\log_2 m_k - 2 > \frac{1}{10}\log_2 m_k, \tag{2.54}$$

provided we re-index so that $m_0 > 2^{20}$. □

2.5 Lower bounds for non-uniform algorithms

The lower bounds obtained above for recursive programs hold also in other models of relative computation. We consider here worst-case non-uniform lower bounds.

Corollary 2.5.1. *There is a rational constant $k > 0$, such that for infinitely many n, if α is a recursive program which computes the Legendre symbol for $x, y < 2^n$, then*

$$\sup\{c_\alpha(x,y) \mid x, y < 2^n\} > kn. \tag{2.55}$$

Proof. Let (a,p) be an element of the sequence of Theorem 2.3.1, and let n be least such that $2a < 2^n$. Let α be a recursive program which computes the Legendre symbol for $x, y < 2^n$. Then α must compute the Legendre symbol for (a,p) and $(\pi(a), \pi(p)) = (2a, p)$, and so, by Theorem 2.3.1,

$$\begin{aligned} c_\alpha(a,p) &> \frac{1}{20}\log_2 a \\ &> \frac{1}{20}(n-2) \\ &> \frac{1}{40}n, \end{aligned} \tag{2.56}$$

where we have used that $a > 2^{n-2}$ and $n > 4$. □

2.6 Extending Z by finitely many Presburger primitives

This Chapter contains lower bounds for algorithms that are expressible as recursive programs relative to the functions in the set

$$\boldsymbol{Lin}_0 = \{+, -, 2\cdot x, \mathrm{quo}_2(x), \mathrm{parity}, \chi_<, \chi_=\}. \tag{2.57}$$

Such algorithms do not have access to general multiplication and division operations, but only multiplication and division by 2. Consider extensions of $\boldsymbol{Lin}_0$ of the form

$$\boldsymbol{Lin}_M = \{+, -, 2 \cdot x, 3 \cdot x, \ldots, M \cdot x, \mathrm{quo}_2(x), \mathrm{quo}_3(x), \ldots, \mathrm{quo}_M(x), \quad (2.58)$$
$$\mathrm{rem}_2(x), \mathrm{rem}_3(x), \ldots, \mathrm{rem}_M(x), \chi_<, \chi_=\},$$

where $M \geq 2$, and, as functions of x with n fixed, $\mathrm{quo}_n(x)$ and $\mathrm{rem}_n(x)$ are the quotient and non-negative remainder in the division of x by n, respectively. Let $\mathbf{Z}_M = (\mathbb{Z}, 0, 1, \boldsymbol{Lin}_M)$. The lower bounds proved in this Chapter also hold—with different multiplicative constants depending on M, but by essentially the same proofs—for recursive programs on $\mathbf{Z}_M$. We outline an argument for this claim by tracing the development of this Chapter, and noting the modifications which suffice to establish it.

First, an appropriate analogue of Lemma 2.2.1 shows that each $x \in G_m^{\mathbf{Z}_M}(p, a)$ can be expressed as

$$x = \frac{c_0 + c_1 p + c_2 a}{D^m}, \quad (2.59)$$

where the coefficients c_k grow exponentially with m, and D is a constant which depends on M. Next, the difficult input a, which depends on p and appears in the statements Lemmas 2.2.2, 2.2.3, and 2.2.4, should be taken not as

$$a = 2^{\lfloor \log_2 p \rfloor + 1} q, \quad (2.60)$$

but rather, to have the form

$$a = D^{\lfloor \log_D p \rfloor + 1} q, \quad (2.61)$$

to account for the denominator in the representation (2.59). After making these changes, it is fairly straightforward to construct an embedding of $\mathbf{Z}_M \restriction G_m^{\mathbf{Z}_M}(p, a)$ into $\mathbf{Z}_M$ which maps (p, a) to $(p, \lambda a)$ as above, but in order to obtain such an embedding stronger assumptions must be made on the size of p relative to m; we must assume that p is larger, but still only exponentially large compared to m.

This assumption leads to a smaller multiplicative constant in a logarithmic lower bound.

In the following Chapter, we consider a parameterized set of primitive operations analogous to $\boldsymbol{Lin}_M$ on certain algebraic rings of integers, mainly because there does not seem to be a particularly natural value of M to fix for these rings. Consequently, the next lower bounds we prove will explicitly show the dependence of the multiplicative constant on the size of the set of piecewise-linear primitive operations.

Presburger arithmetic is the first-order theory of the structure

$$\mathscr{Z} = (\mathbb{Z}, 0, 1, <, +). \tag{2.62}$$

The set of functions definable in this structure is denoted by $\boldsymbol{Lin}$, and each such function is piecewise-linear. It is a classical result that $\mathscr{Z}$ admits elimination of quantifiers when it is extended by the relations $2\mathbb{Z}, 3\mathbb{Z}, 4\mathbb{Z}$, etc., and so the functions of the structures $\mathbf{Z}_M$ are referred to as Presburger operations.

CHAPTER 3

Notions in imaginary norm-Euclidean quadratic integer rings

3.1 Introduction

The greatest common divisor (gcd) of two integers is a fundamental number-theoretic notion. It can be computed using the ancient Euclidean algorithm, which iteratively performs division with remainder. Presented in [Ste67] is a modern, so-called binary gcd algorithm which also computes the gcd of two integers, but uses only comparisons, subtraction, multiplication by 2, and division by 2, and requires no general division operation. The optimality of this algorithm relative to these primitive operations is shown in [DM04].

The notion of gcd generalizes to arbitrary Euclidean Domains, because unique factorization holds in such rings. Given two elements a, b of a Euclidean Domain D, $c \in D$ is a gcd of a and b if $c \mid a, b$, and if $d \in D$ is also such that $d \mid a, b$, then $d \mid c$. A gcd is unique up to multiplication by a unit (invertible element). The binary gcd algorithm for $\mathbb{Z}$ has been generalized in [AF04], [DF05] and [Wei00] to each of the five imaginary norm-Euclidean quadratic integer rings, which are the rings of integers of second-degree algebraic extensions of $\mathbb{Q}$ that are Euclidean Domains with respect to the field norm. Such rings are discussed in more detail in Section 3.2. These algorithms have applications in rational complex arithmetic, and in computing a generator of a finitely-generated ideal.

The binary-like algorithms compute a gcd using subtraction, multiplication and division by a small, fixed set of algebraic integers, and roughly, comparison of norms. We prove lower bounds for the complexity of gcd computation by algorithms which use as given a set of primitive operations in terms of which these algorithms can be naturally expressed. Of course, the strength of a complexity lower bound for a class of algorithms increases with the size of the set of primitive operations that algorithms from the class are permitted to call.

We prove an $\Omega_\infty(\log \max(N(x), N(z)))$-lower bound for deciding coprimality. Here, two algebraic integers are coprime iff their gcd is a unit, and $N(z) = |z|^2$ (absolute value squared) is the norm of z. From this lower bound, we obtain a similar one for computing a gcd, because any gcd algorithm also decides coprimality by first computing a gcd, and then checking in constant time whether it is a unit. Each of the known binary-like gcd algorithms has complexity that is $O(\log^2 \max(N(x), N(z)))$ in our model. Before establishing these bounds, we prove $\Omega_\infty(\log N(z))$-lower bounds for some natural unary decision problems which are of independent interest, and certain technical steps followed to establish these lower bounds will set the pattern for coprimality. As in the previous Chapter, both the constant and the infinitely many difficult inputs absorbed by the Ω-notation are independent of the algorithm.

3.2 Results from algebraic number theory

Let $d \in \mathbb{Z}$ be negative and squarefree. The number field $\mathbb{Q}(\sqrt{d}) \subseteq \mathbb{C}$ has degree 2 over $\mathbb{Q}$, and its ring of integers $\mathcal{O}_d$ consists of algebraic numbers of the form

$a + b\omega$, where $a, b \in \mathbb{Z}$, and ω has the following dependence on d,

$$\omega = \begin{cases} \sqrt{d} & \text{if } d \equiv 2, 3 \pmod{4}, \\ \frac{-1+\sqrt{d}}{2} & \text{if } d \equiv 1 \pmod{4}. \end{cases} \tag{3.1}$$

In particular, $\mathbb{Z} \subseteq \mathcal{O}_d$.

These rings inherit from $\mathbb{Q}(\sqrt{d})$ the algebraic field norm $N(z) = z\overline{z}$, where $\overline{z}$ is the Galois conjugate of z, which is the same as the complex conjugate of z since each $\mathbb{Q}(\sqrt{d})$ is a quadratic extension. The ring of integers $\mathcal{O}_d$ is norm-Euclidean if the field norm N is also a Euclidean norm on $\mathcal{O}_d$, i.e., if for all $x, z \in \mathcal{O}_d$, $z \neq 0$, there exist $q, r \in \mathcal{O}_d$ such that

$$x = qz + r, \qquad \text{with } r = 0 \text{ or } N(r) < N(z). \tag{3.2}$$

A classical result, included in standard texts such as [ST87], is that for $d < 0$, the ring of integers $\mathcal{O}_d$ of the imaginary quadratic extension $\mathbb{Q}(\sqrt{d})$ is norm-Euclidean iff $d \in \{-1, -2, -3, -7, -11\}$. For such values of d, $\mathcal{O}_d$ is said to be an *imaginary norm-Euclidean quadratic integer ring.* The Gaussian integers $\mathbb{Z}[i] = \mathcal{O}_{-1}$, and for a primitive cube root of unity ζ, the Eisenstein integers $\mathbb{Z}[\zeta] = \mathcal{O}_{-3}$, are well-known examples of imaginary norm-Euclidean quadratic integer rings.

We list here some properties of the norm of an imaginary norm-Euclidean quadratic integer ring which are easily verified and which will be used frequently, often without explicit mention. If $|z|$ is the absolute value of the complex number z, then $N(z) = |z|^2$ for all $z \in \mathcal{O}_d$, and it follows by the triangle inequality in $\mathbb{C}$ that for all $x, z \in \mathcal{O}_d$,

$$N(x + z) \leq N(x) + N(z) + 2\sqrt{N(x)N(z)}. \tag{3.3}$$

Since $N(a + b\omega) = (a + b\omega)(a + b\overline{\omega})$, we easily obtain from (3.1) the following explicit formula for the norm,

$$N(a+b\omega) = \begin{cases} a^2 - db^2 & \text{if } d \equiv 2,3 \pmod 4, \\ a^2 + \frac{1-d}{4}b^2 - ab & \text{if } d \equiv 1 \pmod 4. \end{cases} \tag{3.4}$$

In particular, if $a \in \mathbb{Z}$, then $N(a) = a^2$. Clearly, $N(-z) = N(z)$ and $N(z) = 0$ iff $z = 0$. It follows easily that the norm of a non-zero quadratic integer is a positive rational integer. The norm satisfies $N(xz) = N(x)N(z)$, and so, in particular, for $x \in \mathbb{Z}$,

$$N(xz) = x^2 N(z). \tag{3.5}$$

Observe that for $z \neq 0$, if $x \mid z$ in $\mathcal{O}_d$, then $N(x) \leq N(z)$, because $z = qx$ implies that $N(z) = N(q)N(x)$ with $N(q) \geq 1$.

The field norm is a reasonable measure of input size for elements of imaginary quadratic integer rings, since we have for almost every $a, b \in \mathbb{Z}$,

$$\log_2 \max(|a|, |b|) \leq \log_2 N(a+b\omega) \leq 3 \log_2 \max(|a|, |b|), \tag{3.6}$$

and so

$$\log_2 N(a+b\omega) = \Theta(\log \max(|a|, |b|)). \tag{3.7}$$

This is not so in the real case, e.g., in $\mathcal{O}_2$, where each of the (infinitely many) solutions $(a,b) \in \mathbb{Z}^2$ of Pell's equation

$$a^2 - 2b^2 = 1 \tag{3.8}$$

satisfy $N(a+b\omega) = 1$. Another important distinction between imaginary quadratic integer rings and the rings $\mathcal{O}_d$, $d > 0$, which are contained in $\mathbb{R}$, is that it does not appear that an analogue of the triangle inequality like (3.3) exists for these rings.

Since $\mathcal{O}_d$ is discrete, every subset of $\mathcal{O}_d$ that consists of elements which are bounded in norm is finite. Every imaginary norm-Euclidean quadratic integer ring $\mathcal{O}_d$ has a finite group of units,

$$\mathcal{O}_d^{\times} = \{z \in \mathcal{O}_d \mid (\exists x \in \mathcal{O}_d) zx = 1\}. \tag{3.9}$$

It is easy to verify that if $z \in \mathcal{O}_d$ is a unit, then $N(z) = 1$.

Suppose that $\mathcal{O}_d$ is an imaginary norm-Euclidean quadratic integer ring. Given non-zero $x, z \in \mathcal{O}_d$, (3.2) guarantees the existence of a quotient q and remainder r in the division of x by z. It is well known, however, that neither q nor r is necessarily unique, e.g., in the Gaussian integers $\mathcal{O}_{-1}$, we have that

$$\begin{aligned} 5 + 12\sqrt{-1} &= 2(4 + 4\sqrt{-1}) + (-3 + 4\sqrt{-1}), \\ &= (2 + \sqrt{-1})(4 + 4\sqrt{-1}) + 1, \end{aligned} \tag{3.10}$$

with $N(1) = 1$, and

$$N(-3 + 4\sqrt{-1}) = 25 < 32 = N(4 + 4\sqrt{-1}). \tag{3.11}$$

We assume that algorithms can call functions to perform division, and our lower bounds are independent of how a particular quotient and remainder are chosen in division by z, provided the choice is made uniformly for each z. For each non-zero $z \in \mathcal{O}_d$, fix an injection

$$e_z : \{r \mid \text{for some } x, q \in \mathcal{O}_d,\ x = qz + r \ \&\ N(x) < N(z)\} \to \mathbb{N} \tag{3.12}$$

that ranks the finite set

$$\{r \mid \text{for some } x, q \in \mathcal{O}_d,\ x = qz + r \ \&\ N(x) < N(z)\} \tag{3.13}$$

of "potential remainders" in division by z. For the present purposes it will be necessary to fix only finitely many such functions e_z. These rank functions capture a convention by which a remainder in the division by z is chosen in the following sense. For each $x \in \mathcal{O}_d$, we let $\mathrm{rem}_z(x)$ be the e_z-least potential remainder, i.e.,

(i) $z \mid x - \mathrm{rem}_z(x)$, and

(ii) if $z \mid x - r$, then $e_z(\mathrm{rem}_z(x)) \leq e_z(r)$.

The quotient in the division by z is completely determined by the remainder. Let

$$\mathrm{quo}_z(x) = \frac{x - \mathrm{rem}_z(x)}{z}, \tag{3.14}$$

and notice that this uniquely defines a quotient because $\mathcal{O}_d$ is an integral domain: for non-zero $z \in \mathcal{O}_d$, if $z \mid t$ and $t = zq = zq'$, then $z(q - q') = 0$, and so $q = q'$.

We clearly have, for all $x \in \mathcal{O}_d$,

$$x = \mathrm{quo}_z(x)z + \mathrm{rem}_z(x). \tag{3.15}$$

For an example of a natural ranking, we may take e_z to rank the elements of its finite domain in order of increasing norm, and to rank elements of the same norm in order of their principal arguments. The corresponding rem_z can be seen as choosing a least remainder.

3.3 Imaginary norm-Euclidean quadratic algebras

Let $\mathcal{O}_d$ be an imaginary norm-Euclidean quadratic integer ring. For each $M > 1$, let

$$\mathbf{O}_d^M = (\mathcal{O}_d, 0, 1, +, -, \chi_=, x \mapsto \overline{x}, \{x \mapsto zx, \mathrm{quo}_z(x), \mathrm{rem}_z(x)\}_{0<N(z)<M}) \tag{3.16}$$

be the imaginary norm-Euclidean quadratic algebra with underlying set $\mathcal{O}_d$, distinguished constants $0, 1 \in \mathcal{O}_d$, and the functions displayed, where $+, -$ are addition and subtraction respectively, $\chi_=$ is the characteristic function of equality, $x \mapsto \overline{x}$ is the function which takes $x \in \mathcal{O}_d$ to its complex conjugate $\overline{x}$, and

$x \mapsto zx$ is the function which takes $x \in \mathcal{O}_d$ to the product of z with x. Notice that $\mathbf{O}_d^M$ contains functions to perform multiplication, and division with remainder, by only those elements from a fixed, finite set of algebraic integers.

Let

$$D'_{\mathbf{O}_d^M} = \prod_{0<N(z)<M} z. \tag{3.17}$$

Then $D'_{\mathbf{O}_d^M} \in \mathbb{Z}$, since $N(z) < M$ implies that $N(\overline{z}) < M$. Let

$$D = D_{\mathbf{O}_d^M} = |D'_{\mathbf{O}_d^M}|, \tag{3.18}$$

and observe that if $N(z) < M$, then $N(z) \leq D$.

3.4 Primality and squarefreeness

Let $\mathbf{O}_d^M$ be an imaginary norm-Euclidean algebra, and define for all $\alpha \in \mathcal{O}_d$ and $m \in \mathbb{N}$,

$$B_m(\alpha) = \left\{ \frac{c_0 + c_1\alpha}{D^m} \in \mathcal{O}_d \mid c_0, c_1 \in \mathcal{O}_d \ \& \ N(c_0), N(c_1) < (2D)^{3m+1} \right\}. \tag{3.19}$$

Observe that all $z \in \mathcal{O}_d$ with norm $< 2D$ belong to $B_0(\alpha)$. In particular, if $N(z) < M$, then for all $x \in \mathcal{O}_d$, $\mathrm{rem}_z(x) \in B_0(\alpha)$.

Lemma 3.4.1. *If $\boldsymbol{O}_d^M$ is an imaginary norm-Euclidean algebra and $\alpha \in \mathbb{N}$, then for all $m \in \mathbb{N}$, $G_m(\alpha) \subseteq B_m(\alpha)$.*

Proof. The proof is by induction on m. The basis is obvious. For the inductive step, assume that $G_m(\alpha) \subseteq B_m(\alpha)$ for some fixed $m \geq 0$. As noted above, the images of the remainder functions are contained in $B_0(\alpha)$, and are therefore included in every $B_m(\alpha)$. We take cases on the other functions f of $\mathbf{O}_d^M$ which generate $f(\vec{x}) \in G_{m+1}(\alpha)$ from $\vec{x} \in G_m(\alpha)$.

Suppose that $x + y \in G_{m+1}(\alpha)$, with $x, y \in G_m(\alpha)$. By induction, we have the expressions

$$x = \frac{c_0 + c_1\alpha}{D^m}, \quad y = \frac{d_0 + d_1\alpha}{D^m}, \tag{3.20}$$

with $N(c_k), N(d_k) < (2D)^{3m+1}$, $k = 0, 1$. Then

$$x + y = \frac{D(c_0 + d_0) + D(c_1 + d_1)\alpha}{D^{m+1}}, \tag{3.21}$$

and by (3.3),

$$\begin{aligned} N(D(c_k + d_k)) &< 4D^2(2D)^{3m+1} \\ &= (2D)^{3m+3} \\ &< (2D)^{3(m+1)+1}, \end{aligned} \tag{3.22}$$

for $k = 0, 1$, so $x + y \in B_{m+1}(\alpha)$. A similar computation shows that $x - y \in B_{m+1}(\alpha)$ whenever $x, y \in G_m(\alpha)$.

If $\overline{x} \in G_{m+1}(\alpha)$ with $x \in G_m(\alpha)$ expressed as in (3.20), then

$$\overline{x} = \overline{\frac{c_0 + c_1\alpha}{D^m}} = \frac{D\overline{c_0} + D\overline{c_1}\alpha}{D^{m+1}}. \tag{3.23}$$

Since $N(\overline{z}) = N(z)$ for all $z \in \mathcal{O}_d$, we have that

$$N(D\overline{c_k}) < D^2(2D)^{3m+1} < (2D)^{3(m+1)+1}, \quad k = 0, 1, \tag{3.24}$$

and therefore $\overline{x} \in B_{m+1}(\alpha)$.

For multiplication by some fixed $z \in \mathcal{O}_d$, $N(z) < M$, suppose that $zx \in G_{m+1}(\alpha)$ with $x \in G_m(\alpha)$ expressed as in (3.20). Then

$$zx = \frac{zDc_0 + zDc_1\alpha}{D^{m+1}}, \tag{3.25}$$

and

$$N(zDc_k) < D^3(2D)^{3m+1} < (2D)^{3(m+1)+1}, \quad k = 0, 1, \tag{3.26}$$

so $zx \in B_{m+1}(\alpha)$.

For the quotient functions, let $z \in \mathcal{O}_d$ be non-zero and such that $N(z) < M$. Suppose that $\mathrm{quo}_z(x) \in G_{m+1}(\alpha)$ with $x \in G_m(\alpha)$ expressed as in (3.20). We have that

$$\begin{aligned} \mathrm{quo}_z(x) &= \frac{c_0 - D^m \mathrm{rem}_z(x) + c_1\alpha}{zD^m} \\ &= \frac{\frac{D}{z}(c_0 - D^m \mathrm{rem}_z(x)) + \frac{D}{z}c_1\alpha}{D^{m+1}}, \end{aligned} \tag{3.27}$$

and

$$\begin{aligned} &N\left(\frac{D}{z}\left(c_0 - D^m \mathrm{rem}_z(x)\right)\right) \\ &\qquad < D^2\left((2D)^{3m+1} + D^{2m+1} + 2\sqrt{(2D)^{3m+1}D^{2m+1}}\right) \\ &\qquad \leq 4D^2(2D)^{3m+1} \\ &\qquad < (2D)^{3(m+1)+1}. \end{aligned} \tag{3.28}$$

It is even simpler to establish that $N(\frac{D}{z}c_1) < (2D)^{3(m+1)+1}$, hence

$$\mathrm{quo}_z(x) \in B_{m+1}(\alpha). \tag{3.29}$$

□

Lemma 3.4.1 shows that elements of $G_m(\alpha)$ have convenient representations when α is real. The following Lemma implies that this representation is unique when α is sufficiently large relative to m.

Lemma 3.4.2. *Suppose that $\mathcal{O}_d$ is an imaginary norm-Euclidean quadratic ring of integers.*

(i) If $\alpha, c_0, c_1 \in \mathcal{O}_d$ are such that

$$N(c_0) < N(\alpha), \tag{3.30}$$

then

$$c_0 + c_1\alpha = 0 \Longleftrightarrow c_0 = c_1 = 0. \tag{3.31}$$

(ii) *If* $\alpha, c_k \in \mathcal{O}_d$, $k = 0, 1, 2, 3$, *are such that*

$$N(c_0), N(c_2) < N(\alpha)/4, \tag{3.32}$$

then

$$c_0 + c_1\alpha = c_2 + c_3\alpha \Longleftrightarrow c_k = c_{k+2}, \text{ for } k = 0, 1. \tag{3.33}$$

Proof. (i) If $c_1 = 0$, then $c_0 = 0$, so assume $c_1 \neq 0$ so that $N(c_1) \geq 1$. If $c_0 + c_1\alpha = 0$, then $c_0 = -c_1\alpha$. Taking norms yields the contradictory

$$N(c_0) = N(c_1)N(\alpha) \geq N(\alpha). \tag{3.34}$$

(ii) If $c_0 + c_1\alpha = c_2 + c_3\alpha$, then $c_0 - c_2 + (c_1 - c_3)\alpha = 0$. Now,

$$\begin{aligned} N(c_0 - c_2) &\leq N(c_0) + N(c_2) + 2\sqrt{N(c_0)N(c_2)} \\ &< \frac{N(\alpha)}{4} + \frac{N(\alpha)}{4} + 2\sqrt{\frac{N(\alpha)^2}{16}} \\ &= N(\alpha), \end{aligned} \tag{3.35}$$

so the result follows by (i). □

The key to obtaining lower bounds for the unary relations that we consider in this Section is the embedding furnished by the following Lemma.

Lemma 3.4.3. *Let* $\boldsymbol{O}_d^M$ *be an imaginary norm-Euclidean quadratic algebra, with* D *defined as in* (3.18). *If* $m > 0$ *and* $\alpha \in \mathbb{N}$ *satisfies* $N(\alpha) > (2D)^{3m+8}$, *then for all* $\lambda \in \mathbb{N}$ *such that* $\lambda \equiv 1$ $(mod\ D^{m+2})$, *there exists an embedding*

$$\pi : \boldsymbol{O}_d^M \upharpoonright G_m(\alpha) \hookrightarrow \boldsymbol{O}_d^M, \tag{3.36}$$

given by

$$\pi\left(\frac{c_0 + c_1\alpha}{D^{m+1}}\right) = \frac{c_0 + \lambda c_1\alpha}{D^{m+1}}. \tag{3.37}$$

In particular, $\pi(\alpha) = \lambda\alpha$.

Proof. Let m and α be as in the statement of the Lemma. We begin by defining a function

$$\pi : G_{m+1}(\alpha) \to \mathcal{O}_d \tag{3.38}$$

on the larger set $G_{m+1}(\alpha) \supseteq G_m(\alpha)$, on which it is not necessarily an embedding, but does respect differences. We will show that the restriction of π to $G_m(\alpha)$ is an embedding. Fix any $\lambda \in \mathbb{N}$ such that $\lambda \equiv 1 \pmod{D^{m+2}}$, and let q^* be such that $\lambda = q^* D^{m+2} + 1$. Let

$$\pi\left(\frac{c_0 + c_1\alpha}{D^{m+1}}\right) = \frac{c_0 + \lambda c_1\alpha}{D^{m+1}}. \tag{3.39}$$

By Lemmas 3.4.1 and 3.4.2, π is a well-defined injection. It is clear that $\pi(0) = 0$ and $\pi(1) = 1$ because, in general, algebraic numbers of the form

$$\frac{c_0 + c_1\alpha}{D^{m+1}} \tag{3.40}$$

with $c_1 = 0$ are fixed by π. We have that image$(\pi) \subseteq \mathcal{O}_d$, because if

$$\frac{c_0 + c_1\alpha}{D^{m+1}} \in G_{m+1}(\alpha), \tag{3.41}$$

then $D^{m+1} \mid c_0 + c_1\alpha$, so

$$D^{m+1} \mid c_0 + c_1\alpha + q^* D^{m+2} c_1\alpha = c_0 + \lambda c_1\alpha, \tag{3.42}$$

hence

$$\frac{c_0 + \lambda c_1\alpha}{D^{m+1}} \in \mathcal{O}_d. \tag{3.43}$$

Let $x, y \in G_{m+1}(\alpha)$, and suppose that $x - y \in G_{m+1}(\alpha)$. These algebraic integers have representations

$$x = \frac{c_0 + c_1\alpha}{D^{m+1}}, \quad y = \frac{c_2 + c_3\alpha}{D^{m+1}}, \quad x - y = \frac{c_4 + c_5\alpha}{D^{m+1}}, \tag{3.44}$$

with $N(c_k) < (2D)^{3(m+1)+1}$, $0 \le k \le 5$. For $k = 0, 1$, we have that

$$\begin{aligned} N(c_k - c_{k+2}) &< 4(2D)^{3(m+1)+1} \\ &< \frac{(2D)^{3(m+1)+5}}{4} \\ &< \frac{N(\alpha)}{4}, \end{aligned} \tag{3.45}$$

and therefore, by Lemma 3.4.2,

$$c_0 - c_2 = c_4, \quad c_1 - c_3 = c_5. \tag{3.46}$$

Thus

$$\begin{aligned} \pi(x - y) &= \frac{c_4 + \lambda c_5\alpha}{D^{m+1}} \\ &= \frac{c_0 - c_2 + \lambda(c_1 - c_3)\alpha}{D^{m+1}} \\ &= \pi(x) - \pi(y). \end{aligned} \tag{3.47}$$

We proceed to show that $\pi \restriction G_m(\alpha)$ is an embedding. First, observe that for $x \in G_m(\alpha)$,

$$\begin{aligned} \pi(x) &= \pi\left(\frac{c_0 + c_1\alpha}{D^m}\right) \\ &= \pi\left(\frac{Dc_0 + Dc_1\alpha}{D^{m+1}}\right) \\ &= \frac{Dc_0 + \lambda Dc_1\alpha}{D^{m+1}} \\ &= \frac{c_0 + \lambda c_1\alpha}{D^m}, \end{aligned} \tag{3.48}$$

since

$$N(Dc_k) < D^2(2D)^{3m+1} < (2D)^{3(m+1)+1}, \quad k = 0, 1, \tag{3.49}$$

so the representation of x in the second identity above is unique.

To see that π respects complex conjugation, suppose that

$$\frac{c_0 + c_1\alpha}{D^m}, \overline{\frac{c_0 + c_1\alpha}{D^m}} = \frac{d_0 + d_1\alpha}{D^m} \in G_m(\alpha), \tag{3.50}$$

so that $\overline{c_k} = d_k, k = 0, 1$. Then

$$\overline{\pi\left(\frac{c_0 + c_1\alpha}{D^m}\right)} = \overline{\frac{c_0 + \lambda c_1\alpha}{D^m}} = \frac{\overline{c_0} + \lambda\overline{c_1}\alpha}{D^m} = \pi\left(\frac{d_0 + d_1\alpha}{D^m}\right). \tag{3.51}$$

If $N(z) < M$, then the remainder in the division by z is fixed by π, because for every x,

$$\text{rem}_z(x) = \frac{D^m\text{rem}_z(x) + 0\alpha}{D^m}, \tag{3.52}$$

with

$$N(D^m\text{rem}_z(x)) \le D^{2m+1} < (2D)^{3m+1}. \tag{3.53}$$

Therefore,

$$\pi(\text{rem}_z(x)) = \text{rem}_z(x). \tag{3.54}$$

We show next that π is a congruence for residue classes of small moduli in the following sense: if $z \in \mathcal{O}_d$ is such that $N(z) < M$, and $x \in G_{m+1}(\alpha)$, then,

$$\begin{aligned} z \mid x = \frac{c_0 + c_1\alpha}{D^{m+1}} & \qquad (3.55) \\ \iff zD^{m+1} \mid c_0 + c_1\alpha & \\ \iff zD^{m+1} \mid c_0 + c_1\alpha + q^*D^{m+2}c_1\alpha, & \\ \iff zD^{m+1} \mid c_0 + \lambda c_1\alpha & \\ \iff z \mid \frac{c_0 + \lambda c_1\alpha}{D^{m+1}} & \\ \iff z \mid \pi(x). & \end{aligned}$$

Now, to see that π respects the remainder functions, let $x \in G_m(\alpha)$, and fix some non-zero $z \in \mathcal{O}_d$ with $N(z) < M$. Now,

$$N(\mathrm{rem}_z(x)), N(\mathrm{rem}_z(\pi(x))) < N(z) < M, \tag{3.56}$$

hence

$$\mathrm{rem}_z(x), \mathrm{rem}_z(\pi(x)) \in G_1(\alpha) \subseteq G_m(\alpha). \tag{3.57}$$

Since $\pi(\mathrm{rem}_z(x)) = \mathrm{rem}_z(x)$, we need to show that $\mathrm{rem}_z(x) = \mathrm{rem}_z(\pi(x))$. Because $\mathrm{rem}_z(x)$ is the remainder in a division of x by z, we have that $z \mid x - \mathrm{rem}_z(x)$. Clearly, $x - \mathrm{rem}_z(x) \in G_{m+1}(\alpha)$, and so we have by (3.47), (3.54), and (3.55) that

$$\begin{aligned} z \mid \pi(x - \mathrm{rem}_z(x)) &= \pi(x) - \pi(\mathrm{rem}_z(x)) && (3.58)\\ &= \pi(x) - \mathrm{rem}_z(x), && (3.59) \end{aligned}$$

hence $e_z(\mathrm{rem}_z(\pi(x))) \leq e_z(\mathrm{rem}_z(x))$. Similarly,

$$\begin{aligned} z \mid \pi(x) - \mathrm{rem}_z(\pi(x)) &= \pi(x) - \pi(\mathrm{rem}_z(\pi(x))) && (3.60)\\ &= \pi(x - \mathrm{rem}_z(\pi(x))), && (3.61) \end{aligned}$$

thus $z \mid x - \mathrm{rem}_z(\pi(x))$, and therefore

$$e_z(\mathrm{rem}_z(x)) \leq e_z(\mathrm{rem}_z(\pi(x))). \tag{3.62}$$

Since the rank function e_z is a one-one function, we conclude that

$$\mathrm{rem}_z(x) = \mathrm{rem}_z(\pi(x)). \tag{3.63}$$

To see that π respects the quotient functions, let $z \in \mathcal{O}_d$ be such that $N(z) < M$, and assume that $x, y = \mathrm{quo}_z(x) \in G_m(\alpha)$, where

$$x = \frac{c_0 + c_1\alpha}{D^m}, y = \frac{d_0 + d_1\alpha}{D^m}. \tag{3.64}$$

Then

$$\begin{aligned}\operatorname{quo}_z(x) &= \frac{c_0 - D^m \operatorname{rem}_z(x) + c_1\alpha}{zD^m} \\ &= \frac{\frac{D}{z}(c_0 - D^m \operatorname{rem}_z(x)) + \frac{D}{z}c_1\alpha}{D^{m+1}} \\ &= \frac{Dd_0 + Dd_1\alpha}{D^{m+1}}.\end{aligned} \tag{3.65}$$

As in (3.28),

$$N\left(\frac{D}{z}(c_0 - D^m \operatorname{rem}_z(x))\right) < (2D)^{3(m+1)+1} < \frac{N(\alpha)}{4}, \tag{3.66}$$

so we have that

$$\frac{D}{z}(c_0 - D^m \operatorname{rem}_z(x)) = Dd_0, \tag{3.67}$$

and

$$\frac{D}{z}c_1 = Dd_1. \tag{3.68}$$

Therefore

$$\begin{aligned}\operatorname{quo}_z(\pi(x)) &= \frac{c_0 - D^m \operatorname{rem}_z(\pi(x)) + \lambda c_1\alpha}{zD^m} \\ &= \frac{\frac{D}{z}(c_0 - D^m \operatorname{rem}_z(\pi(x))) + \lambda\frac{D}{z}c_1\alpha}{D^{m+1}} \\ &= \frac{\frac{D}{z}(c_0 - D^m \operatorname{rem}_z(x)) + \lambda\frac{D}{z}c_1\alpha}{D^{m+1}} \\ &= \frac{d_0 + \lambda d_1\alpha}{D^m} \\ &= \pi(\operatorname{quo}_z(x)).\end{aligned} \tag{3.69}$$

For multiplication by $z \in \mathcal{O}_d$ with $N(z) < M$, suppose that

$$x, y = zx \in G_m(\alpha). \tag{3.70}$$

We have the representations

$$zx = \frac{zc_0 + zc_1\alpha}{D^m} = \frac{d_0 + d_1\alpha}{D^m} = y, \tag{3.71}$$

and since

$$N(zc_k) \leq D(2D)^{3m+1} < \frac{N(\alpha)}{4}, \quad k = 0, 1, \tag{3.72}$$

we have that $zc_k = d_k$, $k = 0, 1$. Therefore,

$$z\pi(x) = \frac{zc_0 + z\lambda c_1 \alpha}{D^m} = \frac{d_0 + \lambda d_1 \alpha}{D^m} = \pi(zx). \tag{3.73}$$

□

Suppose that $\mathcal{O}_d$ is an imaginary norm-Euclidean quadratic integer ring. A non-zero algebraic integer $z \in \mathcal{O}_d$ is prime in $\mathcal{O}_d$ if z is not a unit, and is irreducible, i.e., if $z = uv$ in $\mathcal{O}_d$ then one of u, v is a unit. An algebraic integer $z \in \mathcal{O}_d$ is squarefree if there is no prime $p \in \mathcal{O}_d$ such that $p^2 \mid z$.

Notice that rational primes $p \in \mathbb{N}$ may fail to be prime in certain $\mathcal{O}_d$, e.g., $2 = (1 + \sqrt{-1})(1 - \sqrt{-1})$ in $\mathcal{O}_{-1}$. Despite this, if $\mathcal{O}_d$ is an imaginary norm-Euclidean quadratic integer ring, then there are infinitely many $p \in \mathbb{N}$ such that p is prime in $\mathcal{O}_d$. In the case $d = -1$, this can be seen as follows. By Dirichlet's Theorem on the primes of an arithmetic progression, there are infinitely many rational primes $p \in \mathbb{N}$ such that $p \equiv 3 \pmod 4$. If p factors as $p = xz$ in $\mathcal{O}_{-1}$ with neither x nor z a unit, then we must have for $x = a + b\sqrt{-1}$ that $N(x) = a^2 + b^2 = p$. It is easy to check that $a^2 + b^2 \equiv 3 \pmod 4$ has no integer solutions, so p is prime in $\mathcal{O}_d$.

Theorem 3.4.1. *There is an infinite set $\{\alpha_k\}_{k \in \mathbb{N}} \subseteq \mathcal{O}_d$ such that if γ is a recursive program in the imaginary norm-Euclidean quadratic algebra $\boldsymbol{O}_d^M$ that decides either of the relations "α is prime" or "α is squarefree", then for all $k \in \mathbb{N}$,*

$$c_\gamma(\alpha_k) > \frac{1}{4 \log_2 2D} \log_2 N(\alpha_k), \tag{3.74}$$

where D is defined as in (3.18).

Proof. Let $\{\alpha_k\}_{k\in\mathbb{N}}$ consist of infinitely many primes in $\mathbb{N}$ such that, for each k, α_k is prime in $\mathcal{O}_d$, and $N(\alpha_k) > (2D)^{32}$. Let γ be a recursive program in $\mathbf{O}_d^M$ that decides either primality or squarefreeness. Fix $k \in \mathbb{N}$, and let $\alpha = \alpha_k$. Let $m = c_\gamma(\alpha) > 0$ and assume for a contradiction that $(2D)^{3m+8} < N(\alpha)$. Let π be an embedding as in the statement of Lemma 3.4.3, with $\lambda = (1 + D^{m+2})^2$.

Since γ decides primality or squarefreeness in $\mathbf{O}_d^M$,

$$\mathbf{O}_d^M \models \gamma(\alpha)^{(m)} = 1, \tag{3.75}$$

hence, by absoluteness,

$$\mathbf{O}_d^M \restriction G_m(\alpha) \models \gamma(\alpha)^{(m)} = 1. \tag{3.76}$$

Therefore, by the Embedding Lemma,

$$\mathbf{O}_d^M \models \gamma(\pi(\alpha))^{(m)} = 1, \tag{3.77}$$

i.e., $\pi(\alpha) = \lambda\alpha$ is either prime or squarefree, but it is neither. Thus $(2D)^{3m+8} \geq N(\alpha)$, hence

$$\begin{aligned} c_\gamma(\alpha) &\geq \frac{\log_{2D} N(\alpha)}{3} - \frac{8}{3} \\ &> \frac{1}{4}\log_{2D} N(\alpha) \\ &= \frac{1}{4\log_2 2D}\log_2 N(\alpha). \end{aligned} \tag{3.78}$$

□

3.5 Coprimality and great common divisor

The main result of this Section is a lower bound for deciding coprimality in imaginary norm-Euclidean quadratic algebras. As above, this is obtained by defining an appropriate embedding, and the one we define here generalizes the one from the previous Chapter.

Let $\mathbf{O}_d^M$ be an imaginary norm-Euclidean algebra, and define for all $\alpha, \beta \in \mathcal{O}_d$ and $m \in \mathbb{N}$,

$$B_m(\alpha, \beta) = \left\{ \frac{c_0 + c_1\alpha + c_2\beta}{D^m} \in \mathcal{O}_d \mid c_k \in \mathcal{O}_d \ \& \ N(c_k) < (2D)^{3m+1} \right\}. \tag{3.79}$$

The elements of the subalgebras of $\mathbf{O}_d^M$ which are generated in m steps by a pair of input values (α, β) have convenient representations as elements of $B_m(\alpha, \beta)$. The following Lemma is analogous to Lemma 3.4.1, and its proof is essentially the same.

Lemma 3.5.1. *If $\boldsymbol{O}_d^M$ is an imaginary norm-Euclidean algebra, D is defined as in* (3.18)*, $\alpha \in \mathbb{N}$ is non-zero, and*

$$\beta = D^{\lfloor \log_D 2N(\alpha) \rfloor + 1}, \tag{3.80}$$

then for all $m \in \mathbb{N}$,

$$G_m(\alpha, \beta) \subseteq B_m(\alpha, \beta). \tag{3.81}$$

The following Lemma implies that the representation of an element of $G_m(\alpha, \beta)$ as a member of $B_m(\alpha, \beta)$ is unique, provided certain conditions on α and β are satisfied.

Lemma 3.5.2. *Suppose that $\boldsymbol{O}_d^M$ is an imaginary norm-Euclidean algebra, $\alpha, \lambda \in \mathcal{O}_d$ are non-zero, and*

$$\beta = D^{\lfloor \log_D 2N(\alpha) \rfloor + 1}. \tag{3.82}$$

(*i*) *If $c_0, c_1, c_2 \in \mathcal{O}_d$ are such that*

$$N(c_0), N(c_1) < N(\alpha), \tag{3.83}$$

then

$$c_0 + c_1\alpha + \lambda c_2\beta = 0 \iff c_k = 0 \textit{ for } k = 0, 1, 2. \tag{3.84}$$

(ii) *If* $c_k, d_k \in \mathcal{O}_d,\ k = 0, 1, 2$ *are such that*

$$N(c_0), N(d_0), N(c_1), N(d_1) < N(\alpha)/4, \tag{3.85}$$

then

$$c_0 + c_1\alpha + \lambda c_2\beta = d_0 + d_1\alpha + \lambda d_2\beta \iff c_k = d_k,\ \textit{for } k = 0, 1, 2. \tag{3.86}$$

Proof. (i) If $c_2 = 0$, then $c_0 = c_1 = 0$ as in Lemma 3.4.2. So assume that $c_2 \neq 0$. Then

$$-\lambda c_2\beta = c_0 + c_1\alpha, \tag{3.87}$$

so, by taking norms,

$$\begin{aligned} N(\beta) &\leq N(\lambda c_2\beta) \\ &\leq N(c_0) + N(c_1)N(\alpha) + 2\sqrt{N(c_0)N(c_1)N(\alpha)} \\ &< N(\alpha) + N(\alpha)^2 + 2N(\alpha)\sqrt{N(\alpha)} \\ &\leq 4N(\alpha)^2. \end{aligned} \tag{3.88}$$

On the other hand,

$$\begin{aligned} N(\beta) &= N(D^{\lfloor \log_D 2N(\alpha)\rfloor + 1}) \\ &\geq (2N(\alpha))^2 \\ &= 4N(\alpha)^2, \end{aligned} \tag{3.89}$$

a contradiction.

(ii) follows from (i) as in the proof of Lemma 3.4.2. □

As in the case of primality and squarefreeness, the key to obtaining a lower bound for the binary relation of coprimality is an embedding. The embedding that we define on $G_m(\alpha, \beta)$ below is asymmetric in the sense that it fixes α, but moves β.

Lemma 3.5.3. *Let $\boldsymbol{O}_d^M$ be an imaginary norm-Euclidean quadratic algebra, with D defined as in* (3.18)*. If $m > 0$, $\alpha \in \mathbb{N}$ satisfies $N(\alpha) > (2D)^{3m+8}$, and*

$$\beta = D^{\lfloor \log_D 2N(\alpha) \rfloor + 1}, \tag{3.90}$$

then for all non-zero $\lambda \in \mathbb{N}$, there exists an embedding

$$\pi : \boldsymbol{O}_d^M \restriction G_m(\alpha, \beta) \hookrightarrow \boldsymbol{O}_d^M, \tag{3.91}$$

given by

$$\pi\left(\frac{c_0 + c_1\alpha + c_2\beta}{D^{m+1}}\right) = \frac{c_0 + c_1\alpha + \lambda c_2\beta}{D^{m+1}}. \tag{3.92}$$

In particular,

$$\pi(\alpha) = \alpha, \quad \pi(\beta) = \lambda\beta. \tag{3.93}$$

Proof. Let m, α, β be as in the statement of the Lemma. As in the proof of Theorem 3.4.3, we begin by defining a function

$$\pi : G_{m+1}(\alpha, \beta) \to \mathcal{O}_d \tag{3.94}$$

on a slightly larger set and then show that the restriction of this map to $G_m(\alpha, \beta)$ is an embedding. Let

$$\pi\left(\frac{c_0 + c_1\alpha + c_2\beta}{D^{m+1}}\right) = \frac{c_0 + c_1\alpha + \lambda c_2\beta}{D^{m+1}}. \tag{3.95}$$

We first observe that

$$\begin{aligned} \lfloor \log_D 2N(\alpha) \rfloor + 1 &\geq \log_D 2N(\alpha) \\ &> \log_D N(\alpha) > \log_{2D} N(\alpha) > 3m + 8, \end{aligned} \tag{3.96}$$

hence $D^{m+2} \mid \beta = D^{\lfloor \log_D 2N(\alpha) \rfloor + 1}$. From this we conclude that

$$\begin{aligned} D^{m+1} \mid c_0 + c_1\alpha + c_2\beta &\Longrightarrow D^{m+1} \mid c_0 + c_1\alpha + c_2\beta + (\lambda - 1)c_2\beta \\ &\Longrightarrow D^{m+1} \mid c_0 + c_1\alpha + \lambda c_2\beta, \end{aligned} \tag{3.97}$$

so image$(\pi) \subseteq \mathcal{O}_d$.

By Lemma 3.5.2, π is an injection. It is clear that π fixes 0 and 1, because in general, π fixes all elements of $G_{m+1}(\alpha, \beta)$ of the form

$$\frac{c_0 + c_1\alpha + c_2\beta}{D^{m+1}}, \tag{3.98}$$

with $c_1 = c_2 = 0$.

It is straightforward to verify, through calculations similar to those in the proof of Lemma 3.4.3, that π respects differences on $G_{m+1}(\alpha, \beta)$, and that the restriction of π to $G_m(\alpha, \beta)$ respects sums, complex conjugates, and products. It is easy to verify that $\pi \restriction G_m(\alpha, \beta)$ fixes all the values of the remainder functions $\mathrm{rem}_z(x)$, $N(z) < M$.

It will follow, as in the proof of Lemma 3.4.3, that the restriction of π to $G_m(\alpha, \beta)$ respects the quotient functions, once we show that it respects the remainder functions. This in turn will follow once we show for all $z \in \mathcal{O}_d$, $N(z) < M$, and for all $x \in G_{m+1}(\alpha, \beta)$,

$$z \mid x \Longleftrightarrow z \mid \pi(x). \tag{3.99}$$

So fix $z \in \mathcal{O}_d$ with $N(z) < M$, and let $x \in G_{m+1}(\alpha, \beta)$. Again using that

$D^{m+2} \mid \beta$, we have that

$$\begin{aligned} z \mid x = \frac{c_0 + c_1\alpha + c_2\beta}{D^{m+1}} &\iff zD^{m+1} \mid c_0 + c_1\alpha + c_2\beta \\ &\iff zD^{m+1} \mid c_0 + c_1\alpha + c_2\beta + (\lambda - 1)c_2\beta \\ &\iff zD^{m+1} \mid c_0 + c_1\alpha + \lambda c_2\beta \\ &\iff z \mid \frac{c_0 + c_1\alpha + \lambda c_2\beta}{D^{m+1}} \\ &\iff z \mid \pi(x). \end{aligned} \tag{3.100}$$

□

Since $\mathcal{O}_d$ admits unique factorization, if $p \in \mathcal{O}_d$ is a prime such that $p \mid z^m$, for some non-zero $m \in \mathbb{N}$ and $z \in \mathcal{O}_d$, then $p \mid z$. This fact is used in the proof of the following Theorem.

Theorem 3.5.1. *There is an infinite set $\{(\alpha_k, \beta_k)\}_{k\in\mathbb{N}} \subseteq \mathcal{O}_d$ such that if γ is a recursive program in the imaginary norm-Euclidean quadratic algebra $\boldsymbol{O}_d^M$ that decides coprimality, then for all $k \in \mathbb{N}$,*

$$c_\gamma(\alpha_k, \beta_k) > \frac{1}{9\log_2 2D} \log_2 \max(N(\alpha_k), N(\beta_k)), \tag{3.101}$$

where D is defined as in (3.18).

Proof. For each $k \in \mathbb{N}$, let $\alpha_k = (D+1)^k$ and

$$\beta_k = D^{\lfloor \log_D 2N(\alpha_k) \rfloor + 1}, \tag{3.102}$$

so that

$$N(\beta_k) = \max(N(\alpha_k), N(\beta_k)). \tag{3.103}$$

Now, D and $D+1$ are coprime because

$$(D+1) - D = 1, \tag{3.104}$$

and so for every k, α_k is coprime to β_k. Suppose that γ decides coprimality in $\mathbf{O}_d^M$. Fix $k \in \mathbb{N}$, and let $(\alpha, \beta) = (\alpha_k, \beta_k)$. Let $m = c_\gamma(\alpha, \beta) > 0$, and suppose that $(2D)^{3m+8} < N(\alpha)$. Let π be an embedding as in the statement of Lemma 3.5.3 with $\lambda = \alpha$.

Since γ decides coprimality,

$$\mathbf{O}_d^M \models \gamma(\alpha, \beta)^{(m)} = 1, \tag{3.105}$$

so by absoluteness,

$$\mathbf{O}_d^M \restriction G_m(\alpha, \beta) \models \gamma(\alpha, \beta)^{(m)} = 1. \tag{3.106}$$

By the Embedding Lemma,

$$\mathbf{O}_d^M \models \gamma(\pi(\alpha), \pi(\beta))^{(m)} = 1, \tag{3.107}$$

which contradicts that γ decides coprimality, because $\pi(\alpha) = \alpha$ and $\pi(\beta) = \alpha\beta$ are not coprime. Thus $(2D)^{3m+8} \geq N(\alpha)$, and if we assume that $N(\alpha) > (2D)^{32}$, then, as in the proof of Theorem 3.4.1,

$$c_\gamma(\alpha, \beta) > \frac{1}{4} \log_{2D} N(\alpha). \tag{3.108}$$

A simple estimate yields $N(\beta) \leq 4D^2 N(\alpha)^2$, and so by a straightforward calculation,

$$\frac{1}{4} \log_{2D} N(\alpha) > \frac{1}{9} \log_{2D} N(\beta) = \frac{1}{9 \log_2 2D} \log_2 N(\beta), \tag{3.109}$$

where we have used that $N(\beta) > N(\alpha) > (2D)^{18}$. The result follows once we delete a finite subset of $\{(\alpha_k, \beta_k)\}_{k \in \mathbb{N}}$ and re-index so that for all $k \in \mathbb{N}$, $N(\alpha_k) > (2D)^{32}$. □

Since a gcd algorithm in $\mathbf{O}_d^M$ decides coprimality by first computing a gcd and then checking in constant time whether it is a unit, Theorem 3.5.1 immediately gives the following.

Corollary 3.5.1. *Let $\boldsymbol{O}_d^M$ be an imaginary norm-Euclidean quadratic algebra. There is an infinite set $\{(x_k, z_k)\}_{k\in\mathbb{N}} \subseteq \mathcal{O}_d$ and a rational constant $r > 0$, such that if γ is a recursive program in $\boldsymbol{O}_d^M$ that computes a gcd of its arguments, then for all $k \in \mathbb{N}$,*

$$c_\gamma(x_k, z_k) > r \log_2 \max(N(x_k), N(z_k)). \tag{3.110}$$

CHAPTER 4

Lower bounds for PCF

In this Chapter of the Dissertation, we show that lower bounds gotten through the embedding method hold also for a relativatized version of PCF [Plo78], a simply-typed λ-calculus with general recursion at each type. PCF programs are closed terms in a certain language, and we consider a call-by-name evaluation strategy given by a set of axioms and rules for rewriting terms. We show that PCF computations can be faithfully represented by recursive programs on a partial algebra which contains functions for performing the syntactic operations of analyzing, parsing, and reducing terms. In a sense made precise below, these operations are logical.

4.1 Logical extensions

Let

$$\mathbf{A} = (A, 0^{\mathbf{A}}, 1^{\mathbf{A}}, \{\phi^{\mathbf{A}}\}_{\phi\in\Phi}) \tag{4.1}$$

be a partial Φ-algebra, and let

$$\mathbf{B} = (B, 0^{\mathbf{B}}, 1^{\mathbf{B}}, \{\phi^{\mathbf{B}}\}_{\phi\in\Phi}, \{\psi^{\mathbf{B}}\}_{\psi\in\Psi}) \tag{4.2}$$

be a partial $(\Phi \cup \Psi)$-algebra. $\mathbf{B}$ is a logical extension (or inessential extension) of $\mathbf{A}$ if (LOG1)–(LOG3) below are satisfied:

(LOG1) $A \subseteq B$, $0^{\mathbf{A}} = 0^{\mathbf{B}}$, and $1^{\mathbf{A}} = 1^{\mathbf{B}}$, (LOG2) For n-ary $\phi \in \Phi$,

$$\phi^{\mathbf{B}}(x_1, \dots, x_n) = \begin{cases} \phi^{\mathbf{A}}(x_1, \dots, x_n) & \text{if } x_1, \dots, x_n \in A, \\ \bot & \text{if some } x_i \notin A. \end{cases} \tag{4.3}$$

(LOG3) If $\pi : A \to A$ is a bijection such that $\pi(0^{\mathbf{A}}) = 0^{\mathbf{A}}$, $\pi(1^{\mathbf{A}}) = 1^{\mathbf{A}}$, then there is a bijection $\rho : B \to B$ such that $\pi \sqsubseteq \rho$, and for each $\psi \in \Psi$,

$$\rho(\psi^{\mathbf{B}}(x_1, \dots, x_n)) = \psi^{\mathbf{B}}(\rho(x_1), \dots, \rho(x_n)). \tag{4.4}$$

It is shown in [DM04] that if a lower bound for computing a function in a partial algebra $\mathbf{A}$ is obtained through the embedding method, then the same lower bound holds in all logical extensions of $\mathbf{A}$, essentially because embeddings can be lifted to logical extensions using (LOG1)–(LOG3).

Typical logical extensions of a partial algebra $\mathbf{A} = (A, 0^{\mathbf{A}}, 1^{\mathbf{A}}, \{\phi^{\mathbf{A}}\}_{\phi \in \Phi})$ extend A with data structures such as queues, lists, trees, etc., and expand $\mathbf{A}$ to include operations such as accessor and mutator functions for working with these data structures. Of course, the introduction and efficient use of data structures occasionally leads to algorithms with substantially smaller time complexity than those which only make use of more primitive data types. However, if a lower bound for the basic parallel complexity for computing a function f in $\mathbf{A}$ is obtained using the embedding method, then the use of logical data structures does nothing to reduce the minimum number of calls to the non-logical primitives of $\mathbf{A}$ that are necessary to compute f, because the lower bound persists in all logical extensions of $\mathbf{A}$.

4.2 Scott-Plotkin PCF

Let ι be a type symbol, intended to denote a set of individuals. The set of finite simple types is generated by the recursion

$$\sigma ::= \ \iota \mid (\sigma_1 \to \sigma_2). \tag{4.5}$$

Let $\mathbf{A}$ be a partial Φ-algebra. The set of (typed) Φ-PCF terms over $\mathbf{A}$ is generated by the recursion,

$$\begin{aligned} M ::= \mathsf{0}^\iota \mid \mathsf{1}^\iota \mid a^\iota \mid \mathsf{v}^\sigma \mid (\lambda \mathsf{v}^\sigma . M^\tau) \mid (M^{\sigma \to \tau} N^\sigma) \mid \\ \phi^{(\iota \times \iota \times \cdots \times \iota) \to \iota}(M_1^\iota \cdots M_n^\iota) \mid \\ \supset^{(\iota \times \iota \times \iota) \to \iota} M_1^\iota M_2^\iota M_3^\iota \mid \\ \mathsf{fix}^{(\sigma \to \sigma) \to \sigma} M^{\sigma \to \sigma}. \end{aligned} \tag{4.6}$$

Here, $a^\iota \in A$ is regarded as a parameter, v^σ is a variable of type σ,

$$\supset^{(\iota \times \iota \times \iota) \to \iota} M_1^\iota M_2^\iota M_3^\iota \tag{4.7}$$

is a conditional expression, and $\mathsf{fix}^{(\sigma \to \sigma) \to \sigma}$ is a fixpoint (recursion) operator. When there is no danger of confusion, the type annotations will not be explicitly shown, and in the absence of parentheses, function application is assumed to be left-associative.

4.2.1 Axioms and rules

A call-by-name immediate reduction strategy ("leftmost outermost") for PCF terms (relative to $\mathbf{A}$) is given by the following sets of axioms and rules for the binary relation $\Rightarrow$ on terms.

Axioms.

(CBN1) $\supset 0MN \Rightarrow N, \supset 1MN \Rightarrow M$.

(CBN2) $\mathsf{fix}\, M \Rightarrow M(\mathsf{fix}\, M)$.

(CBN3) $(\lambda \mathsf{v}.M)N \Rightarrow [N/\mathsf{v}]M$,

where $[N/\mathsf{v}]M$ is the result of substituting N for each free occurrence of v in M after changing bound variables in M so that no free variable of N becomes bound.

(CBN4) If $\phi \in \Phi$ is k-ary, and $a_1, \ldots, a_k$ are parameters, then

$$\phi(a_1, a_2, \cdots, a_k) \Rightarrow \phi^{\mathbf{A}}(a_1, a_2, \ldots, a_k). \tag{4.8}$$

Note that the term on the right-hand-side is a parameter, and that this is the only axiom which depends on the underlying algebra $\mathbf{A}$.

Rules.

(CBN5) $\dfrac{M \Rightarrow M'}{MN \Rightarrow M'N}$.

(CBN6) $\dfrac{M \Rightarrow M'}{\supset M \Rightarrow \supset M'}$.

(CBN7) If $\phi \in \Phi$ is k-ary, $1 \leq m \leq k$, and $a_1, \ldots a_{m-1}$ are parameters, then

$$\frac{N_m \Rightarrow N'_m}{\phi(a_1, a_2, \cdots, a_{m-1}, N_m, \ldots, N_k) \Rightarrow \phi(a_1, a_2, \cdots a_{m-1}, N'_m, \ldots, N_k)}.$$

4.2.2 Computation

Let $\Rightarrow^*$ be the transitive closure of $\Rightarrow$. A closed PCF-term M of type

$$\underbrace{\iota \to \iota \to \cdots \to \iota}_{n} \to \iota \tag{4.9}$$

is a PCF program of arity n. M computes the partial function $f : A^n \rightharpoonup A$ in $\mathbf{A}$, if for all $\vec{x}, y \in A$,

$$f(\vec{x}) = y \text{ iff } M\vec{x} \Rightarrow^* y. \tag{4.10}$$

If M computes a partial function, then for all $\vec{x}$ such that there is a parameter y with $M\vec{x} \Rightarrow^* y$, let

$$\mathscr{L}(\vec{x}) = n, \tag{4.11}$$

if there are terms $M_1, \ldots, M_n \equiv y$ such that

$$M\vec{x} \Rightarrow M_1 \Rightarrow M_2 \Rightarrow \cdots \Rightarrow y, \tag{4.12}$$

and let $\mathscr{L}(\vec{x}) = \infty$ if M does not converge on $\vec{x}$. Since $\Rightarrow$ is deterministic, $\mathscr{L}$ is well-defined, and is a measure of reduction sequence length.

Let

$$\begin{aligned}\boldsymbol{Par} =\{&\textsf{is-0}, \textsf{is-1}, \textsf{is-parameter}, \textsf{is-variable}, \textsf{is-abstraction},\\ &\textsf{is-general-app}, \textsf{is-given-app}, \textsf{is-conditional},\\ &\textsf{is-fixpoint}\},\end{aligned} \tag{4.13}$$

where, is-0, is-1, is-parameter, etc. are predicates which decide if their argument is 0, 1, a parameter, etc.

Let

$$\begin{aligned}\boldsymbol{Lex} =\{&\lambda\textsf{-abstraction-variable}, \lambda\textsf{-abstraction-term},\\ &\textsf{function-in-app}, \textsf{argument-in-app},\\ &\textsf{arg-1-to-}\phi, \ldots, \textsf{arg-}n\textsf{-to-}\phi,\\ &\textsf{cond-1}, \textsf{cond-2}, \textsf{cond-3}, \textsf{fixpoint-term}\},\end{aligned} \tag{4.14}$$

where λ-abstraction-variable, λ-abstraction-term, arg-1-to-ϕ, etc. are functions which return the variable of a λ-abstraction, the term of a λ-abstraction, the first argument to the given ϕ, etc. We assume that these functions are undefined for inappropriate arguments, e.g.,

$$\textsf{λ-abstraction-variable}(\supset^{(\iota\times\iota\times\iota)\to\iota} M_1^\iota M_2^\iota M_3^\iota)\uparrow \tag{4.15}$$

Let

$$\boldsymbol{Sub} = \{\textsf{convert-and-substitute}, \textsf{form-app}\}, \tag{4.16}$$

where

$$\textsf{convert-and-substitute}((\lambda\textsf{v}.M)N) = [N/\textsf{v}]M, \tag{4.17}$$

and the renaming of bound variables in M is done in some straightforward, systematic way which, in particular, does not depend on the parameters occurring in M, and

$$\textsf{form-app}(M, N) = (MN). \tag{4.18}$$

Let $\boldsymbol{Syn}$ be the union of these syntactic functions on terms:

$$\boldsymbol{Syn} = \boldsymbol{Par} \cup \boldsymbol{Lex} \cup \boldsymbol{Sub}. \tag{4.19}$$

Proposition 4.2.1. *Let $\boldsymbol{A}$ be a partial Φ-algebra. If*

$$\boldsymbol{P} = (\{\textit{Closed } \Phi\textit{-PCF terms}\}, 0^{\boldsymbol{A}}, 1^{\boldsymbol{A}}, \{\phi^{\boldsymbol{A}}\}_{\phi\in\Phi}, \boldsymbol{Syn}), \tag{4.20}$$

*is the partial algebra which has as its domain the set of all closed Φ-PCF terms, including elements of A regarded as parameters, and contains the given functions of $\boldsymbol{A}$ together with the syntactic functions from the set **Syn**, then $\boldsymbol{P}$ is a logical extension of $\boldsymbol{A}$.*

Proof. Let $\pi : A \to A$ be a bijection such that $\pi(0^{\mathbf{A}}) = 0^{\mathbf{A}}$ and $\pi(1^{\mathbf{A}}) = 1^{\mathbf{A}}$. Define

$$\rho : \{\text{Closed } \Phi\text{-PCF terms}\} \to \{\text{Closed } \Phi\text{-PCF terms}\} \tag{4.21}$$

so that for any closed Φ-PCF term E, $\rho(E) = \tilde{E}$, where $\tilde{E}$ is gotten from E by replacing all parameters in E with their images under π. (A formal definition of ρ by induction on terms is trivial.)

Clearly, $\pi \sqsubseteq \rho$, and since ρ obviously preserves syntactic structure, we have for all $\psi \in \boldsymbol{Syn}$,

$$\rho(\psi^{\mathbf{B}}(x_1, \ldots, x_n)) = \psi^{\mathbf{B}}(\rho(x_1), \ldots, \rho(x_n)). \tag{4.22}$$

□

We next show that a PCF program on $\mathbf{A}$ can be faithfully represented as a recursive program on a logical extension of $\mathbf{A}$ by $\boldsymbol{Syn}$.

Theorem 4.2.1. *Let $\boldsymbol{A}$ be a partial Φ-algebra, and suppose that M is a Φ-PCF program which computes a partial function $f : A^n \rightharpoonup A$ with reduction sequence length $\mathscr{L}(\vec{x})$. Then there is a small constant $k > 0$, which depends only on Φ, and a recursive program π on*

$$\boldsymbol{P} = (\{\textit{Closed } \Phi\textit{-PCF terms}\}, 0^{\boldsymbol{A}}, 1^{\boldsymbol{A}}, \{\phi^{\boldsymbol{A}}\}_{\phi \in \Phi}, \boldsymbol{Syn}), \tag{4.23}$$

such that π computes f, and for all $\vec{x} \in A^n$,

$$c_\pi(\vec{x}) \leq k\mathscr{L}(\vec{x}). \tag{4.24}$$

Proof. We represent M as the following recursive program π in $\mathbf{P}$, where for readability, calls to the functions of ***Syn*** are not explicit.

$\zeta_0(x_1, \ldots, x_n) = \zeta_1(M x_1 \cdots x_n)$.

$$\zeta_1(E) = \begin{cases} E & \text{if } E \equiv a, \text{ a parameter} \\ \mathsf{if}\ \zeta_1(N)\ \mathsf{then}\ \zeta_1(P)\ \mathsf{else}\ \zeta_1(Q) & \text{if } E \equiv\supset NPQ, \\ \zeta_1(N(\mathsf{fix}\, N)) & \text{if } E \equiv \mathsf{fix}\, N, \\ \zeta_1([P/\mathsf{v}]N) & \text{if } E \equiv (\lambda\mathsf{v}.N)P, \\ \phi(\zeta_1(E_1), E_2, \ldots, E_n) & \text{if } E \equiv \phi(E_1, E_2, \ldots, E_n) \text{ and} \\ & \quad E_1 \text{ is not a parameter,} \\ \phi(a_1, \zeta_1(E_2), \ldots, E_n) & \text{if } E \equiv \phi(a_1, E_2, \ldots, E_n) \text{ and} \\ & \quad a_1 \text{ is a parameter,} \\ \qquad\vdots & \\ \phi(a_1, a_2, \ldots, \zeta_1(E_n)) & \text{if } E \equiv \phi(a_1, a_2, \ldots, E_n) \text{ and} \\ & \quad a_1, \ldots, a_{n-1} \text{ are parameters,} \\ \phi(a_1, a_2, \ldots, a_n) & \text{if } E \equiv \phi(a_1, a_2, \ldots, a_n) \text{ and} \\ & \quad a_1, \ldots, a_n \text{ are parameters,} \\ \zeta_1(\zeta_1(N)P) & \text{if } E \equiv NP, \end{cases}$$

The constant k is the maximum, taken over the syntactic categories of Φ-PCF terms, of the complexity of deciding which ***Syn*** operations to perform, and then performing them, in the term-rewriting procedure expressed by the above program.

It is easy to see both that $\overline{\pi} = f$ and that the complexity of π is as claimed, by induction on the length of the PCF reduction sequence for M, taking cases on the syntactic form of M. □

Since PCF programs can be represented on logical extensions, all of the lower bounds of this Dissertation hold for PCF. For example, we have the following.

Corollary 4.2.1. *There is a rational constant $r > 0$, and an infinite sequence $\{(a_n, p_n)\}_{n \in \mathbb{N}}$, such that if M is a $\boldsymbol{Lin}_0$-PCF program that computes the Legendre symbol with reduction sequence length $\mathscr{L}(a, p)$, then for all $n \in \mathbb{N}$,*

$$\mathscr{L}(a_n, p_n) > r \log_2 \max(a_n, p_n). \tag{4.25}$$

CHAPTER 5

Known algorithms and related bounds

5.1 Other bounds for quadratic residuosity and pseudoprimality

There are few results in the literature which give lower bounds on the complexity of number-theoretic computations relative to a fixed set of given arithmetic operations. Other work in this area includes [Mei91], where the techniques of [MST91a] and [MST91b] are used to derive lower bounds on the depth of decision trees and the time complexity of random access machines (RAMs) which compute the Legendre and Jacobi symbols, pseudoprimality, and modular exponentiation. These computations are relative to a set of operations which is richer than $\boldsymbol{Lin}_0$, and consequently the lower bounds established there—for input values $\leq n$,

$$\Omega(\log\log\log n) \tag{5.1}$$

for the Jacobi symbol, and

$$\Omega(\sqrt{\log\log n}) \tag{5.2}$$

for pseudoprimality and modular exponentiation—are lower than the ones of the present work.

For the other number-theoretic functions that we consider, the following upper bounds are known. There is a recursive program μ on $\mathbf{Z}$ which expresses the binary method for modular exponentiation from [BS96], such that

$$\overline{\mu}^{\mathbf{Z}}(b, c, m) = b^c \pmod{m} \tag{5.3}$$

with

$$c_\mu(b, c, m) = O(\log^2 m). \tag{5.4}$$

Since deciding whether m is pseudoprime to the base b is reducible in constant time to modular exponentiation, the $O(\log^2 m)$ upper bound also holds for pseudoprimality. The lower bounds we obtain in Section 2.4 for this function and relation are both $\Omega_\infty(\log m)$ where, as above, both the constant and the infinitely many values absorbed by the Ω_∞-notation are independent of the recursive program.

During the review of [Bus07], one referee quite correctly observed that the proof of Theorem 2.4.1 also establishes an $\Omega_\infty(\log a)$-lower bound for deciding the predicate "b divides a". There is a known algorithm for deciding this predicate from $\boldsymbol{Lin}_0$ with basic complexity $O(\log a)$, and so this furnishes a new proof of the previously known result [Mos04] that this algorithm is optimal.

5.2 The binary algorithm for the Jacobi symbol

The binary algorithm for the Jacobi symbol from [SS93] can be expressed as a recursive program β on $\mathbf{Z}$ which computes $\left(\frac{a}{n}\right)$ with

$$c_\beta(a, n) = O(\log \max(a, n)). \tag{5.5}$$

This algorithm is described as "probably the most efficient in practice" in [MS98]. We show in Section 2.3 that it is optimal: there is a rational constant $C > 0$ and an infinite set $\{(a_k, n_k)\}_{k \in \mathbb{N}}$ such that if α is any recursive program on $\mathbf{Z}$ which computes the Jacobi symbol, then for all $k \in \mathbb{N}$,

$$c_\alpha(a_k, n_k) > C \log_2 \max(a_k, n_k). \tag{5.6}$$

In particular, we have that

$$c_\alpha(a, n) = \Omega_\infty(\log \max(a, n)), \tag{5.7}$$

but note that the result is stronger than this, since both the constant C and the input values $\{(a_k, n_k)\}_{k\in\mathbb{N}}$ are independent of α.

Proposition 5.2.1 (Shallit and Sorenson [SS93])**.** *The recursive program β below computes the Jacobi symbol $\left(\frac{a}{n}\right)$ on $\boldsymbol{Z}$ with*

$$c_\beta(a, n) = O(\log \max(a, n)), \tag{5.8}$$

for $a, n > 0$ and odd n.

$$\zeta_\beta(a,n) = \begin{cases} 0 & \text{if } a = 0 \text{ and } n \neq 1, \\ 1 & \text{if } a = 0 \text{ and } n = 1, \\ \zeta_\beta(a/2, n) & \text{if } a \text{ is even, and} \\ & \qquad n \equiv 1 \ (mod\ 8) \text{ or } n \equiv 7 \ (mod\ 8), \\ -\zeta_\beta(a/2, n) & \text{if } a \text{ is even, and} \\ & \qquad n \equiv 3 \ (mod\ 8) \text{ or } n \equiv 5 \ (mod\ 8), \\ \zeta_\beta(a - n, n) & \text{if } a > n, \\ \zeta_\beta(n - a, a) & \text{if } a \not\equiv 3 \ (mod\ 4) \text{ or } n \not\equiv 3 \ (mod\ 4), \\ -\zeta_\beta(n - a, a) & \text{if } a \equiv 3 \ (mod\ 4) \text{ and } n \equiv 3 \ (mod\ 4). \end{cases} \tag{5.9}$$

Proof. The correctness of the algorithm is shown in [SS93], and follows from basic properties of the Jacobi symbol which may be found, e.g., in [HW79] and [Lan58].

Observe that the remainder in division by 4, the remainder in division by 8, and multiplication on $\{-1, 0, 1\}$ are all definable by $\boldsymbol{Lin}_0$-terms of fixed, finite depth, and therefore have constant complexity. In particular, β is a program on $\mathbf{Z}$. To determine its complexity, notice that in computing $\left(\frac{a}{n}\right)$ from (5.9), at worst, one of the arguments a, n is halved at every other step. Each such step involves no more than some fixed number of calls to the functions in $\boldsymbol{Lin}_0$, and so

$$c_\beta(a, n) = O(\log a + \log n) = O(\log \max(a, n)). \tag{5.10}$$

□

Corollary 5.2.1. *Let β be the recursive program defined by* (5.9)*. There is a rational number $r > 0$ and an infinite sequence $\{(a_k, p_k)\}_{k\in\mathbb{N}}$ such that if α is any recursive program on $\boldsymbol{Z}$ which computes the Jacobi symbol, then for every $k \in \mathbb{N}$,*

$$c_\alpha(a_k, p_k) > r c_\beta(a_k, p_k). \tag{5.11}$$

Proof. Let $\{(a_k, p_k)\}_{k\in\mathbb{N}}$ be as in Theorem 2.3.1, and by Proposition 1, let $M \in \mathbb{N}$ be such that for all $a, n > 0$ with odd n,

$$c_\beta(a, n) \leq M \log_2 \max(a, n). \tag{5.12}$$

For pairs (a, n) with prime n, the Jacobi symbol agrees with the Legendre symbol, and so for all $k \in \mathbb{N}$,

$$c_\alpha(a_k, p_k) > \frac{1}{20} \log_2 \max(a_k, p_k) \geq \frac{1}{20M} c_\beta(a_k, p_k). \tag{5.13}$$

□

5.3 Binary-like gcd algorithms in imaginary norm-Euclidean quadratic integer rings

The complexity lower bound for gcd computation that appears in the statement of Corollary 3.5.1 applies to the large class of gcd algorithms which can be expressed as recursive programs in $\mathbf{O}_d^M$. We show in this Section that known algorithms belong to this class, and have complexity that is quadratic in our lower bound.

We first consider the binary-like gcd algorithm from [Wei00], which computes a gcd in the Gaussian integers $\mathbb{Z}[i] = \mathcal{O}_{-1}$. It is easiest to understand in the following terms, where we write $i = \sqrt{-1}$. If one of the arguments is 0, the algorithm outputs the other argument. If both arguments are non-zero, the algorithm extracts factors of the prime $(1 + i)$ from both arguments until they are each relatively prime to $(1 + i)$. When this stage is reached, the argument that is larger in norm is replaced with a certain difference, which is smaller in norm and divisible by $(1+i)$. The algorithm continues recursively, and terminates after finitely many steps because the norms of the arguments decrease after each stage. The strong similarity of this algorithm with the binary gcd algorithm for $\mathbb{Z}$ is evident.

This description is an oversimplification because the algorithm is optimized not to compute norms exactly, but only to compute them approximately. Specifically, when both arguments are relatively prime to $(1 + i)$, the algorithm does not replace the argument that is larger in norm, but instead, replaces the argument which is approximately larger in norm. Suitable norm approximations suffice for the algorithm's correctness, and lead to a reduction in its complexity. The approximations are determined from the lengths of the arguments and the values of their most-significant bits, and these are complex values in our model.

Consequently, we express the algorithm below in its non-optimized form, in which norms are computed exactly and compared in logtime. Norm computations and comparisons are complex; the authors of each of the binary-like gcd algorithms considered in this Section use clever approximations, and the embeddings defined in Lemmas 3.4.3 and 3.5.3 do not respect norm comparisons.

Proposition 5.3.1 ([Wei00])**.** *There is a recursive program ρ in $\boldsymbol{O}^5_{-1}$ such that for all $x, z \in \mathcal{O}_{-1}$, $\overline{\rho}(x, z)$ is a gcd of x, z, and*

$$c_\rho(x, z) = O(\log^2 \max\{N(x), N(z)\}). \tag{5.14}$$

Proof. Let ρ be the following recursive program in $\mathbf{O}^5_{-1}$.

$$\zeta(x, z) = \begin{cases} z & \text{if } x = 0, \\ x & \text{if } z = 0, \\ (1+i)\zeta(\mathrm{quo}_{1+i}(x), \mathrm{quo}_{1+i}(z)) & \text{if } \mathrm{rem}_{1+i}(x) = 0, \text{ and} \\ & \qquad \mathrm{rem}_{1+i}(z) = 0, \\ \zeta(\mathrm{quo}_{1+i}(x), z) & \text{if } \mathrm{rem}_{1+i}(x) = 0, \text{ and} \\ & \qquad \mathrm{rem}_{1+i}(z) \neq 0, \\ \zeta(x, \mathrm{quo}_{1+i}(z)) & \text{if } \mathrm{rem}_{1+i}(x) \neq 0, \text{ and} \\ & \qquad \mathrm{rem}_{1+i}(z) = 0, \\ \zeta(x - \varepsilon z, z) & \text{if } N(z) \leq N(x) \text{ and } \varepsilon \in \mathcal{O}^\times_{-1} \\ & \qquad \text{minimizes } N(x - \varepsilon z), \\ \zeta(x, z - \varepsilon x) & \text{if } N(x) < N(z) \text{ and } \varepsilon \in \mathcal{O}^\times_{-1} \\ & \qquad \text{minimizes } N(z - \varepsilon x). \end{cases} \tag{5.15}$$

Here, $M = 5$ is chosen to allow multiplication and division by $(1+i)$ and 2. As discussed above, multiplication and division by $(1+i)$ is central to the algorithm. Operations involving 2 are used to multiply algebraic integers—specifically z and $\overline{z}$ to obtain $N(z)$—in logtime.

The correctness of ρ is established in [Wei00], and uses the fact that if

$$(1+i) \nmid x, z, \tag{5.16}$$

then

$$(1+i) \mid (x - \varepsilon z) \tag{5.17}$$

for every $\varepsilon \in \mathcal{O}_{-1}^{\times}$. To estimate the running time of the algorithm, notice that the norm of an argument is reduced by a factor of $N(1+i) = 2$ at least as often as every other step. For the other steps, suppose that $N(z) \leq N(x)$ and that $\varepsilon \in \mathcal{O}_{-1}^{\times}$ minimizes $N(x - \varepsilon z)$. A geometric argument in [Wei00] shows that $N(x - \varepsilon z) < N(x)$. Thus the number of steps is

$$O(\log \max\{N(x), N(z)\}). \tag{5.18}$$

Each step involves either removing a factor of $(1+i)$ from one or both arguments, or computing and comparing norms. The former can be done in constant time. For the latter, observe that norms can be computed in logtime by the familiar technique of multiplication by halving, and two natural numbers can be compared in logtime.[1] Hence

$$c_\rho(x, z) = O(\log^2 \max\{N(x), N(z)\}). \tag{5.19}$$

[1] In fact, if $\chi_{\mathbb{N}_<}(x, z)$ is the characteristic function of $<$ on $\mathbb{N}$, which is undefined for arguments $\notin \mathbb{N}$, then we can take $\chi_{\mathbb{N}_<}$ as a primitive operation and work in the expansion $(\mathbf{O}_d^M, \chi_{\mathbb{N}_<})$. It can be shown that all of the results about $\mathbf{O}_d^M$ obtained above also go through for this expansion, and so we get the same lower bounds when natural numbers can be compared in one step. This does nothing to improve the asymptotic complexity of the known binary-like gcd algorithms, and increases the conceptual complexity of the model of computation by introducing the truly *partial* algebra $(\mathbf{O}_d^M, \chi_{\mathbb{N}_<})$, which contains a function that diverges on some input values.

$\square$

The gcd algorithms of [AF04] and [DF05] compute gcd's in $\mathbf{O}_d^M$ for small M and $d \in \{-2, -3, -7, -11\}$. They are similar in spirit to Weilert's algorithm above, and have the same asymptotic complexity.

It is unknown whether an $\Omega_\infty(\log^2 \max\{N(x), N(z)\})$-lower bound holds for gcd computation in $\mathbf{O}_d^M$. The techniques used in this dissertation do not seem to yield superlogarithmic lower bounds.

There do not appear to be any published algorithms which decide whether a given algebraic integer is prime or squarefree in an imaginary norm-Euclidean quadratic integer ring, without resorting to factoring.

5.4 Comparison with Boolean circuit complexity

There is an extensive literature on size lower bounds for Boolean circuits which compute number-theoretic functions and predicates, e.g., [ASS01, BDS01, Gat87, GS91, Woo04]. Among these results, those which give lower bounds on circuit breadth are apparently difficult to relate to lower bounds on depth like Corollary 2.5.1, cf. Introduction of [ASS01]. The results which do address depth are also difficult to compare to ours, because the circuit model has different primitive operations and also assumes a different representation of the input.

References

[AF04] Saurabh Agarwal and Gudmund Skovbjerg Frandsen. “Binary GCD like algorithms for some complex quadratic rings.” In *Algorithmic number theory*, volume 3076 of *Lecture Notes in Comput. Sci.*, pp. 57–71. Springer, Berlin, 2004.

[ASS01] Eric Allender, Michael Saks, and Igor Shparlinski. “A lower bound for primality.” *J. Comput. System Sci.*, **62**(2):356–366, 2001.

[BDS01] Anna Bernasconi, Carsten Damm, and Igor Shparlinski. “Circuit and decision tree complexity of some number theoretic problems.” *Information and Computation*, **168**(2):113–124, 2001.

[BS96] Eric Bach and Jeffrey Shallit. *Algorithmic number theory. Vol. 1.* Foundations of Computing Series. MIT Press, Cambridge, MA, 1996.

[Bus07] J. Busch. “On the optimality of the binary algorithm for the Jacobi symbol.” *Fund. Inform.*, **76**(1-2):1–11, 2007.

[DF05] Ivan Bjerre Damgård and Gudmund Skovbjerg Frandsen. “Efficient algorithms for the gcd and cubic residuosity in the ring of Eisenstein integers.” *J. Symbolic Comput.*, **39**(6):643–652, 2005.

[DM04] Lou van den Dries and Yiannis N. Moschovakis. “Is the Euclidean algorithm optimal among its peers?” *Bulletin of Symbolic Logic*, **10**(3):390–418, 2004.

[DM07] Lou van den Dries and Yiannis N. Moschovakis. “Arithmetic Complexity.” *To appear in ACM Transactions on Computational Logic*, 2007.

[Gat87] Joachim von zur Gathen. “Computing powers in parallel.” *SIAM J. Comput.*, **16**(5):930–945, 1987.

[GS91] Joachim von zur Gathen and Gadiel Seroussi. “Boolean circuits versus arithmetic circuits.” *Information and Computation*, **91**(1):142–154, 1991.

[HW79] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers.* The Clarendon Press Oxford University Press, New York, fifth edition, 1979.

[Lan58] Edmund Landau. *Elementary number theory.* Chelsea Publishing Co., New York, N.Y., 1958. Translated by J. E. Goodman.

[McC63] John McCarthy. "A basis for a mathematical theory of computation." In *Computer programming and formal systems*, pp. 33–70. North-Holland, Amsterdam, 1963.

[Mei91] João Meidânis. "Lower bounds for arithmetic problems." *Information Processing Letters*, **38**:83–87, 1991.

[Mos04] Yiannis N. Moschovakis. "Arithmetic complexity." unpublished notes, 2004.

[Mos05] Yiannis N. Moschovakis. "Recursion and complexity." In S. Barry Cooper, Benedikt Löwe, and Leen Torenvliet, editors, *New Computational Paradigms: First Conference on Computability in Europe, CiE 2005, Amsterdam, The Netherlands, June 8-12, 2005. Proceedings*, volume 3526/2005, pp. 350–357. Springer Lecture Notes in Computer Science, 2005.

[MS98] Shawna Meyer Eikenberry and Jonathan P. Sorenson. "Efficient algorithms for computing the Jacobi symbol." *J. Symbolic Comput.*, **26**(4):509–523, 1998.

[MST91a] Yishay Mansoor, Baruch Schieber, and Prasoon Tiwari. "A lower bound for integer greatest common divisor computations." *Journal of the Association for Computing Machinery*, **38**:453–471, 1991.

[MST91b] Yishay Mansoor, Baruch Schieber, and Prasoon Tiwari. "Lower bounds for computations with the floor operation." *SIAM Journal on Computing*, **20**:315–327, 1991.

[Plo78] G. D. Plotkin. "LCF considered as a programming language." *Theoret. Comput. Sci.*, **5**(3):223–255, 1977/78.

[SS93] J. O. Shallit and J. P. Sorenson. "A binary algorithm for the Jacobi symbol." *ACM SIGSAM Bulletin*, **27**:4–11, 1993.

[ST87] Ian Stewart and David Tall. *Algebraic number theory.* Chapman and Hall Mathematics Series. Chapman & Hall, London, second edition, 1987.

[Ste67] J. Stein. "Computational Problems Associated with Racah Algebra." *Journal of Computational Physics*, **1**:397–405, 1967.

[Wei00] André Weilert. "$(1+i)$-ary GCD computation in $\mathbf{Z}[i]$ is an analogue to the binary GCD algorithm." *J. Symbolic Comput.*, **30**(5):605–617, 2000.

[Woo04] Alan R. Woods. “Subset sum “cubes” and the complexity of primality testing.” *Theoret. Comput. Sci.*, **322**(1):203–219, 2004.