Arithmetic Complexity

LOU VAN DEN DRIES
University of Illinois, Urbana-Champaign

and

YIANNIS N. MOSCHOVAKIS
University of California, Los Angeles and Graduate Program in Logic and Algorithms, University of Athens, Greece

We obtain *lower bounds* on the cost of computing various arithmetic functions and deciding various arithmetic relations from specified primitives. This includes lower bounds for computing the greatest common divisor and deciding coprimeness of two integers, from primitives like addition, subtraction, division with remainder and multiplication. Some of our results are in terms of recursive programs, but they generalize directly to most (plausibly all) algorithms from the specified primitives. Our methods involve some elementary number theory as well as the development of some basic notions and facts about recursive algorithms.

Categories and Subject Descriptors: F.1.3 [**Complexity measures and classes**]: Machine-independent complexity; F.2.1. [**Numerical algorithms and problems**]: Number-theoretic computations

General Terms: Lower bounds for arithmetical problems

Additional Key Words and Phrases: Recursive programs, greatest common divisor, coprimeness

1. INTRODUCTION

Motivating question

To what extent is the *euclidean algorithm* optimal with respect to the number of steps? Are there algorithms that compute the greatest common divisor of two integers in many fewer arithmetic steps?

It is well-known that there are real constants $c_1 > c_2 > 0$ such that the euclidean algorithm takes *at most* $c_1 \log a$ steps on *all* integer inputs (a, b) with $a > b > 0$ and *at least* $c_2 \log a$ steps on *infinitely many* integer inputs (a, b) with $a > b > 0$,

Authors' addresses. Lou van den Dries, Department of Mathematics, University of Illinois, 1409 W. Green Street, Urbana, IL 61801, email: vddries@math.uiuc.edu. Yiannis N. Moschovakis, Department of Mathematics, University of California, Los Angeles, CA 90095-1555, email: ynm@math.ucla.edu.
The research for this article was co-funded by the European Social Fund and National Resources - (EPEAEK II) PYTHAGORAS in Greece.

where each step consists of taking a remainder in an integer division.

One typical result in this paper, Theorem 3.8, is roughly as follows:

Any algorithm computing the greatest common divisor and using addition, subtraction, and integer division with remainder as given, will take more than $\frac{1}{4}\log\log a$ *steps on infinitely many integer inputs* (a, b) *with* $a > b > 1$.

"Using ... as given" means that in each *arithmetic* step we add, subtract, divide with remainder, and compare integers obtained at earlier steps. (There are also *logical* steps.)

We prove similar results for the arithmetic complexity of other problems such as deciding coprimeness of two integers, deciding whether an integer is a perfect square, a power of two, or a prime number. In some results we restrict to *Presburger* (piecewise linear) operations as given, or, in the other direction, allow multiplication among the primitives as well. Thus we vary both the function to be computed, and the set of primitives.

With only Presburger operations as given, we obtain single logarithmic lower bounds for a large class of arithmetic functions; when we also allow integer division with remainder, we get double logarithmic ($\log\log$) bounds (as in the statement above), and when we allow multiplication as well, we obtain $\sqrt{\log\log}$-bounds. (All logarithms are with respect to base 2.)

The following features add extra strength to some of our bounds:

(i) explicit inputs (characterized by natural arithmetic conditions) that witness the lower bounds;

(ii) lower bounds on just the number of *arithmetic* steps, not counting *logical steps* (equality tests and boolean operations);

(iii) allowing parallelism in the algorithms;

(iv) local (non-uniform) lower bounds for *term-complexity* with terms using arithmetic and boolean operations as well as equality tests.

We have tried to make this paper easily accessible to anyone interested in computational complexity, including those with no background knowledge in this subject. To achieve this goal we have provided explicit definitions of notions that might be less familiar to some readers, with brief summaries of basic facts as needed.

After a short preliminary section, this paper divides naturally into three parts, with the last two largely independent of the first, except that some of the results of Part 1 are given a more clearly algorithmic interpretation in Parts 2 and 3.

In Part 1 (Sections 3 and 4) we focus on the connection

greatest common divisor – irrationality,

improving results from [van den Dries 2003] and [van den Dries and Moschovakis 2004].

In Part 2 (Sections 5 – 10), we first develop a local (non-uniform) notion of arithmetic term-complexity, which measures for each $N > 0$ the minimum complexity of a term $t(\vec{x})$ that computes a given function $f : \mathbb{N}^n \to \mathbb{N}$ for all n-tuples of numbers $\leq N$. We formulate and prove a whole array of further results, including in Section 7 the optimality of the *binary* gcd algorithm among algorithms that only allow Presburger operations as arithmetic primitives, and in Section 9 a log log lower bound for terms which decide coprimeness on initial segments of ℕ, from Presburger operations and division with remainder—the same primitives used by the euclidean algorithm.

In Part 3 we review briefly the basic theory of *recursive programs*, and establish lower bounds for global (uniform) algorithms corresponding to the non-uniform lower bounds of Part 2. Many of these results have appeared in [van den Dries and Moschovakis 2004], but they are easy to derive here from the results of Part 2.

The proofs in Part 1 depend on minimal and widely accepted assumptions about "algorithms", and so they are completely general. To derive lower bounds for decision problems in Parts 2 and 3 we must make some assumptions about the structure of algorithms, but these too are rather innocuous and we can argue that these lower bounds also hold for all algorithms.

In the last section of the paper we have included a list of upper bounds in the literature for the complexity of the functions and relations we consider, as we know them.

The subject in this paper has deep historical roots, but the lower bounds are recent. To our knowledge, the earliest substantial lower bounds close to ours are about fifteen years old and are due to Mansour, Schieber and Tiwari [1991a; 1991b]. They use *decision-tree complexity*.

In another vein, Moschovakis [2003] considers *primitive recursive* algorithms with piecewise linear functions as given, proves a linear lower bound on the complexity of any such algorithm that computes the greatest common divisor function, and asks if this result persists when the remainder function is included among the primitives. Van den Dries [2003] answers this question affirmatively. However, these results on primitive recursion have a negative character, by showing that all such algorithms are very inefficient compared to known algorithms like the euclidean.

We soon noticed that some ideas in [van den Dries 2003] were relevant beyond primitive recursion, to essentially *all* algorithms that compute the greatest common divisor. We also saw how to improve the bounds in [van den Dries 2003] and extend its methods to other functions and to decision problems, e.g., coprimeness. This led to non-trivial lower bounds on algorithms computing various familiar arithmetic functions and using various combinations of arithmetic operations as given. A communication with an outline of our methods and some of these results was published in [van den Dries and Moschovakis 2004]. The present paper is intended to give a more complete account of these improvements and extensions, and their applications to lower bounds.

Articles [Moschovakis 2003; van den Dries 2003] were written in ignorance of [Mansour et al. 1991a; 1991b] and use different methods. Here we recover most of the lower bounds of [Mansour et al. 1991a; 1991b] in a stronger form by allowing more liberal notions of computation enjoying one or more of the features (i)–(iv) above. But we do not know whether the triple logarithmic lower bound of [Mansour et al. 1991a] on the decision-tree complexity for deciding coprimeness is also valid for our term-complexity (and the same arithmetic primitives); we do obtain in Theorem 4.1 a better lower bound for computing the gcd using these same primitives.

Conventions and notations

Throughout, m and n range over $\mathbb{N} := \{0, 1, 2, \dots\}$. We define the *floor* or *integer part* of $x \in \mathbb{R}$ by

$$\lfloor x \rfloor := \text{ largest integer } \leq x.$$

For positive $x \in \mathbb{R}$, $\log x := \log_2 x$ is the base 2 logarithm of x. For $a, b \in \mathbb{R}$, $b \neq 0$ we write

$$a = qb + r, \quad q = \lfloor \frac{a}{b} \rfloor \in \mathbb{Z}$$

and we view the *integer quotient* q and the *remainder* r as functions of (a, b): $\mathrm{iq}(a, b) := q$ and $\mathrm{rem}(a, b) := r$. It is convenient to extend iq and rem to all of $\mathbb{R}^2$ by setting $\mathrm{iq}(a, 0) := a$ and $\mathrm{rem}(a, 0) := a$, so for all $a, b \in \mathbb{R}$ we have $a = \mathrm{iq}(a, b)b + \mathrm{rem}(a, b)$, with $0 \leq \mathrm{rem}(a, b) < b$ if $b > 0$.

Let $a, b \in \mathbb{Z}$. We say that a *divides* b (notation: $a|b$) if $ax = b$ for some $x \in \mathbb{Z}$. The *greatest common divisor* of a, b is by definition the unique nonnegative integer $c =: \gcd(a, b)$ such that $c\mathbb{Z} = a\mathbb{Z} + b\mathbb{Z}$. (So $\gcd(a, b)$ is indeed the largest integer dividing both a and b, if $a \neq 0$ or $b \neq 0$.) We say that a and b are *coprime* (notation: $a \perp b$) if $\gcd(a, b) = 1$.

2. THE EUCLIDEAN AND THE BINARY ALGORITHMS FOR THE GCD

This paper is about lower bounds for classes of algorithms computing a given function; the closer we can get such a lower bound to an upper bound for some known algorithm in the class, the better. This is why we briefly discuss here upper bounds for some specific algorithms.

The euclidean algorithm

Some historians believe that this algorithm lies behind the discovery of incommensurable magnitudes by the Pythagoreans. Whether or not this belief is correct, the notions of *greatest common divisor* and *incommensurability* fit into a common algorithmic setting, and we find this illuminating. It helps to understand why the number of steps needed to decide coprimeness of two integers goes to infinity as their ratio approximates a fixed irrational number, as we show in several variants in later sections. Here we summarize the euclidean algorithm in a way that highlights this common context. We refer to [Knuth 1973] for more details on this "granddaddy of all algorithms".

Let $a, b \in \mathbb{R}$, $a > b > 0$. We define a strictly decreasing (finite or infinite) sequence of real numbers

$$r_0 > r_1 > r_2 > \dots$$

by $r_0 := a$, $r_1 := b$, $r_{i+1} := \mathrm{rem}(r_{i-1}, r_i)$ for $i > 0$ as long as $r_i \neq 0$. The sequence stops with last term $r_{l+1} = 0$ as soon as this value is reached. We then put $l(a, b) := l =$ the euclidean length of (a, b).

Key properties

(i) The sequence is finite if and only if a and b are commensurable, that is, a/b is rational.

(ii) $r_{i+1} < (1/2)r_{i-1}$ for $i > 0$.

In particular, if $a, b \in \mathbb{Z}$, then the sequence is finite with last term $r_{l+1} = 0$ with $\gcd(a, b) = r_l$ and $l = l(a, b) \leq 2\log\ b$.

This bound is good enough for us. A logarithmic upper bound is indeed realistic in view of an old result by Lamé which says that the "worst case" occurs when the inputs are consecutive Fibonacci numbers; see [Knuth 1973]. A precise formulation is as follows.

Let $F_n := n^{\text{th}}$ Fibonacci number: $F_0 = 0$, $F_1 = 1$ and $F_{n+1} := F_n + F_{n-1}$ for $n > 0$. For $n > 0$, let a, b be integers such that $a > b > 0$, $l(a, b) = n$ and a is minimal with these properties. Then $a = F_{n+1}$, $b = F_n$.

Above we mentioned only the remainders r_i, but in actual implementations of the algorithm the integer quotients $q_i := \mathrm{iq}(r_i, r_{i+1})$ get computed as well, and are also significant: they are the *partial quotients* of the continued fraction expansion of a/b.

The binary (Stein) algorithm

(See Section 4.5.2 of [Knuth 1973]). This algorithm computes (recursively) $\gcd(a, b)$ for positive integers a, b as follows: if a, b are both even, use $\gcd(a, b) = 2\gcd(a/2, b/2)$; if a is even and b is odd, use $\gcd(a, b) = \gcd(a/2, b)$; if a and b are both odd, and $a > b$, use $\gcd(a, b) = \gcd((a - b)/2, b)$; the remaining cases are treated by interchanging a and b, and using $\gcd(a, b) = a$ if $a = b$.

The basic arithmetic relations and operations used here are: parity, comparison, subtraction, multiplication by 2 and division by 2. The binary algorithm is simpler than the euclidean algorithm, in the sense that each step involves only a linear operation. (Another contrast to the euclidean algorithm is that the binary algorithm does not extend naturally to real numbers.) In practice, this algorithm can be competitive with the euclidean algorithm on suitable inputs, especially when these inputs are given in binary notation.

An easy induction shows that it takes at most $2(\log a + \log b)$ steps to compute $\gcd(a, b)$ via the binary algorithm. For a precise statement to this effect we refer to the beginning of Section 7, which contains an optimality result about the binary algorithm.

PART 1. THE GREATEST COMMON DIVISOR

Here[1] we obtain in a purely arithmetic setting lower bounds on the *depth* of greatest common divisor computations. Parts 2 and 3 are needed to deal with *relations* such as coprimeness.

[1]The material in this part improves results of [van den Dries 2003] and is due to van den Dries.

Let $a, b \in \mathbb{Z}$. We define a sequence $(G_n(a,b))$ of finite subsets of $\mathbb{Z}$:

$$\begin{aligned} G_0(a,b) &\subseteq G_1(a,b) \subseteq G_2(a,b) \subseteq \dots \\ G_0(a,b) &:= \{0,1,a,b\} \\ G_{n+1}(a,b) &:= G_n(a,b) \cup \{x+y, x-y, \mathrm{iq}(x,y), \mathrm{rem}(x,y) : x,y \in G_n(a,b)\}. \end{aligned}$$

We set

$$\mathrm{g}(a,b) := \text{ least } n \text{ such that } \gcd(a,b) \in G_n(a,b).$$

By Section 2 we have $\mathrm{g}(a,b) \leq 2\log b$ when $a > b > 0$. In Section 3 we obtain a double logarithmic lower bound for g on an explicit sequence of rational approximations to $\sqrt{2}$.

Next we define an increasing sequence of finite subsets of $\mathbb{Z}$,

$$G_0^\times(a,b) \subseteq G_1^\times(a,b) \subseteq G_2^\times(a,b) \subseteq \dots$$

in the same way as we defined the $G_n(a,b)$, with $G_0^\times(a,b) = \{0,1,a,b\}$, except that $G_{n+1}^\times(a,b)$ contains also all products xy with $x,y \in G_n^\times(a,b)$. We set

$$\mathrm{g}^\times(a,b) := \text{ least } n \text{ such that } \gcd(a,b) \in G_n^\times(a,b).$$

We have a crude upper bound: $\mathrm{g}^\times(a,b) \leq 2\log\log b$ for $a \geq b \geq 4$, see Section 4 of [van den Dries 2003].[2] In Section 4 we prove a $\sqrt{\log\log}$ lower bound for $\mathrm{g}^\times$.

Notice that $\mathrm{g}(a,b)$ counts the number of steps needed to generate $\gcd(a,b)$ from $a, b, 0, 1$ where at each step we apply $+$, $-$, iq and rem to all integers obtained at earlier steps; and so it can be argued that *every algorithm which computes* $\gcd(a,b)$ *from these operations will take at least that many steps to produce its output.* And similarly with $\mathrm{g}^\times(a,b)$ when we add multiplication to the given operations: any algorithm which computes $\gcd(a,b)$ from $+$, $-$, iq, rem and $\times$ will need at least $\mathrm{g}^\times(a,b)$ steps to produce its output.

3. A LOWER BOUND FOR THE GCD FROM $+, -, \div$

Throughout this section a, b and h denote positive integers. We prove here the following lower bound for g at certain inputs:

Theorem 3.1. *If $a^2 - 2b^2 = 1$, $b > 2$, then*

$$\mathrm{g}(a+1,b) > \frac{1}{6}\log\log a.$$

In this connection we recall (see for example [Rose 1994]) that the positive integer solutions of the Pell equation $x^2 - 2y^2 = 1$ grow exponentially: if (a_n, b_n) is the nth positive integer solution of this equation $(n \geq 1)$, then

$$(a_1,b_1) = (3,2), \quad (a_2,b_2) = (17,12),$$

and in general

$$a_n = \frac{(3+2\sqrt{2})^n + (3-2\sqrt{2})^n}{2}, \quad b_n = \frac{(3+2\sqrt{2})^n - (3-2\sqrt{2})^n}{2\sqrt{2}}.$$

[2]Note that $\mathrm{g}^\times$ differs slightly from the function $g^\times$ in [van den Dries 2003].

The proof of the theorem proceeds in several lemmas, most of which do not depend on (a,b) satisfying $a^2-2b^2=1$. The full hypothesis comes in through the following, easy number-theoretic fact, which is even more important in the next section:

LEMMA 3.2. *Suppose that $a^2-2b^2=1$. Then*

(i) *either* $a \equiv 1 \mod 4$ *and* $\gcd(a+1,b)=\sqrt{2(a+1)}$, *or*
(ii) $a \equiv 3 \mod 4$ *and* $\gcd(a+1,b)=\sqrt{a+1}$.

PROOF. Clearly a is odd and b is even. We first consider the case that $a \equiv 1 \mod 4$. Then we have the factorization $b^2=(a-1)\cdot(a+1)/2$, with coprime factors $a-1$ (which is even), and $(a+1)/2$ (which is odd). Thus $a-1=u^2$, $(a+1)/2=v^2$, and $b=uv$, with coprime positive integers u,v. Hence $\gcd(a+1,b)=2v=\sqrt{2(a+1)}$. The case $a \equiv 3 \mod 4$ is treated similarly. □

What we really use from this lemma is the following consequence:

$$a^2-2b^2=1 \implies \sqrt{b} \;\le\; \gcd(a+1,b) \;\le\; 2\sqrt{b}. \tag{1}$$

Another key fact we need is an *effective measure of irrationality* of $\sqrt{2}$ coming from a theorem of Liouville [Rose 1994]: for all a and b:

$$|\frac{a}{b}-\sqrt{2}| > \frac{1}{5b^2}.$$

Note also that if $a^2-2b^2=1$, then

$$0<\frac{a}{b}-\sqrt{2}=\frac{1}{ab+b^2\sqrt{2}}<\frac{1}{b^2}.$$

However, the next four Lemmas 3.3 – 3.6 only use the much weaker

$$|\frac{a}{b}-\sqrt{2}|<\frac{1}{b} \tag{2}$$

and refer to multiples

$$\alpha := \kappa a, \quad \beta := \kappa b,$$

where κ is any positive integer. For the proof of Theorem 3.1 we just need the case $\kappa=1$ of the next four lemmas, but other uses of these lemmas require higher values of κ.

LEMMA 3.3. *Suppose* (2) *holds, $h \ge 2$, $b \ge 10h^2$ and $f=x+y\alpha+z\beta$ with integers x,y,z such that $|x|,|y|,|z| \le h$, and y and z not both zero. Then*

$$f=\beta\big(y\sqrt{2}+z+\epsilon\big), \quad |\epsilon|<\frac{2h}{b}<|y\sqrt{2}+z|.$$

In particular, $f \ne 0$ and f has the same sign as $y\sqrt{2}+z$.

PROOF. We have

$$\begin{aligned}\frac{f}{\beta} &= \frac{x}{\beta}+y\frac{a}{b}+z=\frac{x}{\beta}+y(\frac{a}{b}-\sqrt{2})+y\sqrt{2}+z\\ &= y\sqrt{2}+z+\epsilon, \qquad \epsilon := \frac{x}{\beta}+y(\frac{a}{b}-\sqrt{2}),\end{aligned}$$

so $|\epsilon| < \frac{2h}{b}$. If $y \neq 0$, then $|y\sqrt{2} + z| > \frac{1}{5|y|}$ by Liouville's inequality, which in view of $b \geq 10h^2$ yields

$$|y\sqrt{2} + z| > \frac{1}{5h} \geq \frac{2h}{b} > |\epsilon|,$$

If $y = 0$ and $z \neq 0$, then obviously $|y\sqrt{2} + z| = |z| \geq 1 > \frac{1}{5h} > |\epsilon|$. □

LEMMA 3.4. *Suppose* (2) *holds,* $h \geq 2$, $b \geq 10h^4$, $f = x_1 + y_1\alpha + z_1\beta$, $g = x_2 + y_2\alpha + z_2\beta$, *where* $x_i, y_i, z_i \in \mathbb{Z}$, $|x_i|, |y_1|, |z_i| \leq h$ *for* $i = 1, 2$ *and* y_2 *and* z_2 *are not both* 0 (*so* $g \neq 0$ *by the lemma above*). *Then*

$$\frac{f}{g} = \frac{(y_1z_2 - y_2z_1)\sqrt{2} + (z_1z_2 - 2y_1y_2)}{z_2^2 - 2y_2^2} + \epsilon, \quad |\epsilon| \leq 7.$$

PROOF. Decomposing f and g as in the previous lemma yields

$$f = \beta\big(y_1\sqrt{2} + z_1 + \epsilon_1\big), \quad |\epsilon_1| < \frac{2h}{b}$$
$$g = \beta\big(y_2\sqrt{2} + z_2 + \epsilon_2\big), \quad |\epsilon_2| < \frac{2h}{b}.$$

Hence

$$\begin{aligned}\frac{f}{g} &= \frac{y_1\sqrt{2} + z_1 + \epsilon_1}{y_2\sqrt{2} + z_2 + \epsilon_2} = \frac{(-y_2\sqrt{2} + z_2)(y_1\sqrt{2} + z_1 + \epsilon_1)}{(-y_2\sqrt{2} + z_2)(y_2\sqrt{2} + z_2 + \epsilon_2)}\\ &= \frac{F + \delta_1}{G + \delta_2}, \quad F := (y_1z_2 - y_2z_1)\sqrt{2} + (z_1z_2 - 2y_1y_2), \quad G := z_2^2 - 2y_2^2,\\ \delta_1 &:= (-y_2\sqrt{2} + z_2)\epsilon_1, \quad \delta_2 := (-y_2\sqrt{2} + z_2)\epsilon_2.\end{aligned}$$

With $\delta_3 := \frac{-\delta_2}{G+\delta_2}$ we have $1 + \delta_3 = \frac{G}{G+\delta_2}$, so $\frac{f}{g} = \frac{(F+\delta_1)(1+\delta_3)}{G}$, hence

$$\frac{f}{g} = \frac{F}{G} + \epsilon, \quad \epsilon := \frac{\delta_1(1 + \delta_3) + F\delta_3}{G}.$$

It remains to show that $|\epsilon| \leq 7$. Clearly,

$$|\delta_1| \leq (1 + \sqrt{2})h|\epsilon_1| \leq (1 + \sqrt{2})\frac{2h^2}{b} < \frac{5h^2}{b} \leq \frac{1}{2},$$

and similarly, $|\delta_2| < \frac{5h^2}{b} \leq 1/2$. Since G is a nonzero integer we have $|G| \geq 1$, so $|G + \delta_2| > 1/2$, and thus $|\delta_3| < 2|\delta_2|$. Also, $|F| \leq 2h^2\sqrt{2} + 3h^2 \leq 6h^2$. Hence

$$|\epsilon| \leq 2\delta_1 + |F\delta_3| \leq 1 + 6h^2\frac{10h^2}{b} = 1 + \frac{60h^4}{b} \leq 7. \quad \square$$

With $\alpha = \kappa a$, $\beta = \kappa b$ as above, let

$$A(\kappa; h) := \mathbb{Z} \cap \Big\{\frac{x + y\alpha + z\beta}{w} : x, y, z, w \in \mathbb{Z}, |x|, |y|, |z| \leq h, 0 < w \leq h\Big\}$$

and set $A(h) := A(1; h)$.

LEMMA 3.5. *If* (2) *holds,* $h \geq 2$, $b \geq 10h^8$, *and* $f, g \in A(\kappa; h)$, *then the numbers* $f + g, f - g, \mathrm{iq}(f, g), \mathrm{rem}(f, g)$ *belong to* $A(\kappa; h^8)$.

It follows that for all a, b satisfying (2) *and all $\nu \in \mathbb{N}$,*

$$[h = 2^{2^{3\nu}} \;\&\; b \geq 10h] \implies [G_\nu(a+1, b) \subseteq A(h) \;\&\; G_\nu(\alpha, \beta) \subseteq A(\kappa; h)]. \qquad (3)$$

Proof. Assume (2), $h \geq 2$, $b \geq 10h^8$, and $f, g \in A(\kappa; h)$, $g \neq 0$. Write

$$f = \frac{x_1 + y_1\alpha + z_1\beta}{w_1} \quad \text{and} \quad g = \frac{x_2 + y_2\alpha + z_2\beta}{w_2}$$

where $x_i, y_i, z_i, w_i \in \mathbb{Z}$, $|x_i|, |y_i|, |z_i| \leq h, 0 < w_i \leq h$ for $i = 1, 2$. Adding these fractions in the usual way shows that $f + g, f - g \in A(\kappa; 2h^2) \subseteq A(\kappa; h^8)$. As to $\mathrm{iq}(f, g)$ and $\mathrm{rem}(f, g)$, consider first the case that $y_2 \neq 0$ or $z_2 \neq 0$. Since

$$\frac{f}{g} = \frac{w_2x_1 + w_2y_1\alpha + w_2z_1\beta}{w_1x_2 + w_1y_2\alpha + w_1z_2\beta},$$

the previous lemma with h^2 for h yields

$$\frac{f}{g} = \frac{w_2}{w_1} \, \frac{(y_2z_1 - y_1z_2)\sqrt{2} + (2y_1y_2 - z_1z_2)}{2y_2^2 - z_2^2} + \epsilon, \quad |\epsilon| \leq 7.$$

Hence

$$\begin{aligned}|\mathrm{iq}(f, g)| &\leq |w_2|\big(|y_2z_1 - y_1z_2|\sqrt{2} + |2y_1y_2 - z_1z_2|\big) + 8\\ &\leq h(2\sqrt{2} + 3)h^2 + 8 \leq 7h^3.\end{aligned}$$

In particular, $\mathrm{iq}(f, g) \in A(h^8)$. Also

$$\begin{aligned}\mathrm{rem}(f, g) = f - \mathrm{iq}(f, g)g &= \frac{X + Y\alpha + Z\beta}{w_1w_2}, \text{ with}\\ X &= w_2x_1 - \mathrm{iq}(f, g)w_1x_2\\ Y &= w_2y_1 - \mathrm{iq}(f, g)w_1y_2\\ Z &= w_2z_1 - \mathrm{iq}(f, g)w_1z_2.\end{aligned}$$

Hence $|X|, |Y|, |Z| \leq h^2 + 7h^5 \leq h^8$. Thus $\mathrm{rem}(f, g) \in A(\kappa; h^8)$.

Next, suppose that $y_2 = z_2 = 0$, so $|\mathrm{rem}(f, g)| < |g| \leq h$, in particular $\mathrm{rem}(f, g) \in A(\kappa; h) \subseteq A(\kappa; h^8)$. Also

$$\mathrm{iq}(f, g) = \frac{f - \mathrm{rem}(f, g)}{g} = \frac{(x_1 - \mathrm{rem}(f, g)w_1) + y_1\alpha + z_1\beta}{gw_1},$$

which yields $\mathrm{iq}(f, g) \in A(\kappa; h + h^2) \subseteq A(\kappa; h^8)$.

Now (3) follows by an easy induction on ν. □

Lemma 3.6. *Suppose* (2) *holds, $h \geq 2$, $b \geq 400h^4$, and $f \in A(\kappa; h)$. Then*

$$|f| < \sqrt{b} \quad \text{or} \quad |f| > 2\sqrt{b}.$$

Proof. Write $f = \frac{x+y\alpha+z\beta}{w}$ with $x, y, z, w \in \mathbb{Z}, |x|, |y|, |z| \leq h, 0 < w \leq h$. If $y = z = 0$, then $|f| \leq h < \sqrt{b}$. Suppose $y \neq 0$ or $z \neq 0$. Then $x + y\alpha + z\beta = \beta(y\sqrt{2} + z + \epsilon)$ as in lemma 3.3, so $|y\sqrt{2} + z| > \frac{1}{5h}$ and $|\epsilon| \leq \frac{2h}{b} < \frac{1}{10h}$. Hence

$$|f| = |\frac{\beta(y\sqrt{2} + z + \epsilon)}{w}| > \frac{b}{10h^2} \geq 2\sqrt{b}$$

where the last inequality uses $b \geq 400h^4$. □

Proof of Theorem 3.1. Suppose $a^2 - 2b^2 = 1$, $b > 2$, and for any $\nu \in \mathbb{N}$, let $h := 2^{2^{3\nu}}$. By Lemma 3.5,

$$b \geq 10h \implies G_\nu(a+1, b) \subseteq A(h).$$

With the key inequality (1), Lemma 3.6, and $b > a/2$, this gives

$$a \geq 800h^4 \implies \gcd(a+1, b) \notin G_\nu(a+1, b) \implies \mathrm{g}(a+1, b) > \nu.$$

Thus with $\nu := \mathrm{g}(a+1, b)$ we have $a < 800h^4$. Since $b > 2$ we have $\nu \geq 1$. Taking logarithms, the inequality $a < 800h^4$ yields

$$\log a < \log 800 + 4 \log h = \log 800 + 4 \cdot 2^{3\nu} < 10 + 4 \cdot 2^{3\nu} < 2^{6\nu},$$

hence $\log \log a < 6\nu$, which completes the proof of Theorem 3.1. □

Other inputs.

The next theorem gives a similar lower bound for g on a much bigger set of inputs. The inputs of Theorem 3.1 grow exponentially and are thus very sparse, while those in Theorem 3.8 have subquadratic growth. We also need the inputs of Theorem 3.8 in Section 9 to deal with coprimeness.

Lemma 3.7. *Suppose (2) holds, $b \geq 12h^2$ and $\kappa > h$. Then $\kappa \notin A(\kappa; h)$.*

Proof. Towards a contradiction, assume $\kappa \in A(\kappa; h)$, so $\kappa = \frac{x+y\alpha+z\beta}{w}$ with $w, x, y, z \in \mathbb{Z}$, $w \neq 0$ and $|w|, |x|, |y|, |z| \leq h$. Then $(x - \kappa w) + y\alpha + z\beta = 0$, so, as in the proof of Lemma 3.3,

$$\frac{x}{\kappa b} - \frac{w}{b} + y(\frac{a}{b} - \sqrt{2}) + y\sqrt{2} + z = 0,$$

$$|\frac{x}{\kappa b} - \frac{w}{b} + y(\frac{a}{b} - \sqrt{2})| < \frac{1}{b} + \frac{2h}{b} < \frac{1}{5h}.$$

But $|y\sqrt{2} + z| > \frac{1}{5h}$ if $y \neq 0$ or $z \neq 0$. Therefore $y = z = 0$, hence $x = \kappa w$, so $|x| > h$, a contradiction. □

Theorem 3.8. *Suppose (2) holds with $a \perp b$ and $a \geq 2^{256}$. Let $\nu \in \mathbb{N}$ be maximal with $a \geq 20h^2$ where $h := 2^{2^{3\nu}}$. Then*

$$\mathrm{g}(\alpha, \beta) > \frac{1}{4} \log \log \alpha,$$

where $\kappa := h + 1$, $\alpha := \kappa a$, and $\beta := \kappa b$.

Proof. We have $a \geq 20 \cdot 2^{16}$, so $\nu \geq 1$ and $b \geq 12h^2$. From $\gcd(\alpha, \beta) = \kappa$, and $\kappa \notin A(\kappa; h)$, we conclude $\mathrm{g}(\alpha, \beta) \geq \nu + 1$ by Lemma 3.5. Also,

$$a < 20 \cdot \left(2^{2^{3(\nu+1)}}\right)^2 = 20 \cdot 2^{2^{3\nu+4}},$$

so $\alpha = \kappa a < 20 \cdot \left(2^{2^{3\nu}} + 1\right) \cdot 2^{2^{3\nu+4}}$, so

$$\log \alpha < \log 20 + 2^{3\nu} + 1 + 2^{3\nu+4},$$

hence $\log \log \alpha < 3\nu + 5$, so

$$\mathrm{g}(\alpha, \beta) \geq \nu + 1 > \frac{1}{3} \log \log \alpha - \frac{2}{3} \geq \frac{1}{4} \log \log \alpha$$

where the last inequality uses that $a \geq 2^{256}$. □

We claim that the inputs α in this theorem have subquadratic growth. To see this, let a be any sufficiently large prime number, and then take b such that $|a-b\sqrt{2}|<1$, so $a \perp b$. It remains to observe that

$$\alpha = a\kappa \;\leq\; a(1+\sqrt{a/20}) \;\leq\; a^{3/2},$$

and that the nth prime number is $\leq n^{11/10}$ for large enough n.

We can increase the set of inputs further by allowing a real parameter ρ:

Theorem 3.9. *Let $0<\rho\leq 1$. Then there is a $b(\rho)>0$ such that for all a,b, if $a>b>b(\rho)$, $|\sqrt{2}-a/b|<1/b^{\rho}$, and $a\perp b$, then*

$$\mathrm{g}(\alpha,\beta) > \frac{1}{4}\log\ \log\ \alpha, \quad \textit{where } \alpha=\kappa a,\ \beta=\kappa b,\ \kappa := 1+\lfloor\sqrt{a^{\rho}/20}\rfloor.$$

Sketch of Proof. First extend the lemmas in this section by taking b^{ρ} instead of b in the inequalities that involve b; next, for large enough a,b as in the hypothesis of the theorem, take $\nu\in\mathbb{N}$ maximal with $a^{\rho}\geq 20h^2$ where $h:=2^{2^{3\nu}}$, and then follow the proof of Theorem 3.8.

Open problem

Do there exist sequences (α_n) and (β_n) of positive integers and a real $C>0$ such that $\alpha_n\to\infty$ and $\mathrm{g}(\alpha_n,\beta_n)>C\log\alpha_n$ for all n? A positive solution would answer one version of the motivating question in the introduction.

4. A LOWER BOUND FOR THE GCD FROM $+,-,\div,\times$

In this section we shall prove the following result:

Theorem 4.1. *If a and b are positive integers such that $a^2-2b^2=1$, then*

$$\mathrm{g}^{\times}(a+1,b)\geq\frac{1}{4}\sqrt{\log\log b}.$$

For a polynomial $f(Y)=a_0+a_1Y+\cdots+a_nY^n\in\mathbb{R}[Y]$ (all $a_i\in\mathbb{R}$) we put

$$\|f\| = \max_i |a_i|.$$

Note that then for $f,g\in\mathbb{R}[Y]$ with $\deg f\leq n$ or $\deg g\leq n$ we have

$$\|f+g\|\leq\|f\|+\|g\|, \qquad \|fg\|\leq(n+1)\cdot\|f\|\cdot\|g\|.$$

For the proof of the theorem we introduce distinct indeterminates X and Y as place holders for numbers a and b as above. Consider the integral domain

$$R:=\mathbb{R}[X,Y]/(X^2-2Y^2-1),$$

so R is the coordinate ring over $\mathbb{R}$ of the hyperbola $X^2-2Y^2=1$. Let x and y denote the images of X and Y in R. The canonical map $\mathbb{R}[X,Y]\to R$ maps the subring $\mathbb{R}[Y]$ of $\mathbb{R}[X,Y]$ isomorphically onto the subring $\mathbb{R}[y]$ of R, and

$$R=\mathbb{R}[x,y]=\mathbb{R}[y]+\mathbb{R}[y]x, \quad x^2=2y^2+1.$$

For $F\in R$, take the unique $f,g\in\mathbb{R}[Y]$ with $F=f(y)+g(y)x$, and set

$$\|F\|=\max(\|f\|,\|g\|).$$

We equip the fraction field $\mathbb{R}(x, y)$ of R with the unique ordering that makes it an ordered field with $y > \mathbb{R}$ and $x > 0$. Note that then

$$x = y\sqrt{2}(1 + \epsilon y^{-1}),$$

where $\epsilon \in \mathbb{R}(x, y)$ is an infinitesimal, that is, $|\epsilon| < 1/n$ for all $n > 0$. We are actually going to work inside the subring $\mathbb{Q}[x, y] = \mathbb{Q}[y] + \mathbb{Q}[y]x$ of R, and on this subring the ordering is easy to make explicit: let $F \in \mathbb{Q}[x, y]$, $F \neq 0$, and write $F = f(y) + g(y)x$ with $f, g \in \mathbb{Q}[Y]$. Let $d := \max(\deg f, 1 + \deg g) \in \mathbb{N}$, and let $r, s \in \mathbb{Q}$ be the coefficients of Y^d in f and Y^{d-1} in g, respectively. (So $d = 0$ means $f = r \neq 0$ and $g = s = 0$.) Then $F = (r + s\sqrt{2})y^d(1 + \epsilon y^{-1})$ with infinitesimal $\epsilon \in \mathbb{R}(x, y)$, so

$$F > 0 \iff r + s\sqrt{2} > 0.$$

This suggests defining a degree function on $\mathbb{Q}[x, y]$ by

$$\deg F := \max(\deg f, 1 + \deg g) \in \mathbb{N} \text{ for } F \neq 0, \quad \deg 0 := -\infty.$$

Note that then, for $F, G \in \mathbb{Q}[x, y]$:

$$\deg(FG) = \deg F + \deg G, \quad \deg F \leq 0 \iff F \in \mathbb{Q}.$$

We define the set $P(n, h) \subseteq \mathbb{Q}[x, y]$ as follows:

$$P(n, h) := \Big\{\frac{F}{c} : F \in \mathbb{Z}[x, y], \deg F \leq n, \|F\| \leq h, c \in \mathbb{Z}, 0 < c \leq h\Big\}.$$

Below, h is a positive integer, and a, b are positive integers such that $a^2 - 2b^2 = 1$. Then, writing $F \in R$ as $F = f(y) + g(y)x$ with $f, g \in \mathbb{R}[Y]$, we have the evaluation map $F \mapsto F(a, b) := f(b) + g(b)a : R \to \mathbb{R}$. This map respects the ring operations; it also respects the ordering when restricted to $P(n, h)$ with h small compared to a:

Lemma 4.2. *If $b \geq 20h^2$ and $F \in P(n, h)$, $F > 0$, then $F(a, b) > 0$. If $b \geq 80h^4$, then the evaluation map $F \mapsto F(a, b) : P(n, h) \to \mathbb{Q}$ is injective and preserves order: $F, G \in P(n, h), F < G \implies F(a, b) < G(a, b)$.*

Proof. Assume that $b \geq 20h^2$. We begin with an easy bound:
Suppose $r, s \in \mathbb{Z}$ are not both zero, and $|r|, |s| \leq h$. Then

$$\frac{1}{6h} < |r + s\frac{a}{b}| < 3h, \qquad r + s\sqrt{2} > 0 \implies r + s\frac{a}{b} > 0.$$

To see this, note that $\frac{1}{5b^2} < |\sqrt{2} - \frac{a}{b}| < \frac{1}{2b^2}$, so

$$|r + s\frac{a}{b}| \leq |r + s\sqrt{2}| + |s||\sqrt{2} - \frac{a}{b}| < 3h.$$

Hence $|(r + s\frac{a}{b})(r - s\frac{a}{b}| = |r^2 - s^2\frac{a^2}{b^2}| = |r^2 - 2s^2 - \frac{s^2}{b^2}| > \frac{1}{2}$, since $|r^2 - 2s^2| \geq 1$. Thus $|r + s\frac{a}{b}| > \frac{1}{2|r - sa/b|} > \frac{1}{6h}$. The remaining implication follows from

$$|(r + s\sqrt{2}) - (r + s\frac{a}{b})| = |s||\sqrt{2} - \frac{a}{b}| \leq \frac{h}{2b^2} < \frac{1}{6h} < |r + s\frac{a}{b}|.$$

Next, let $0 < F \in \mathbb{Z}[y] + \mathbb{Z}[y]x$, with $\|F\| \leq h$, and put $d := \deg F$. The case $d = 0$ is trivial, so we assume $d > 0$. Write $F = f(y) + g(y)x$, with

$$f = r_d Y^d + \cdots + r_0 \in \mathbb{Z}[Y], \qquad g = s_{d-1}Y^{d-1} + \cdots + s_0 \in \mathbb{Z}[Y].$$

From $F > 0$ we obtain $r_d + s_{d-1}\sqrt{2} > 0$, and hence $r_d + s_{d-1}\frac{a}{b} > 0$. Setting $s_{-1} := 0$, we have:

$$F(a,b) = \sum_{i=0}^{d} c_i b^i, \quad c_i := r_i + s_{i-1}\frac{a}{b} \text{ for } i = 0, \ldots, d.$$

The polynomial $\sum c_i Y^i \in \mathbb{R}[Y]$ has leading coefficient $c_d = r_d + s_{d-1}\frac{a}{b}$, which is positive, so this polynomial takes only positive values for real arguments $> 1 + \max\{\frac{|c_i|}{|c_d|} : 0 \leq i < d\}$; this last quantity is by the bound above $\leq 1 + 18h^2 < 20h^2$, so $F(a,b) = \sum_{i=0}^{d} c_i b^i > 0$.

The second part of the lemma follows from the first part by noting that $P(n,h) + P(n,h) \subseteq P(n, 2h^2)$, and $20(2h^2)^2 = 80h^4$. $\square$

Lemma 4.3. *Let $h \geq n \geq 1$, $b \geq 2^{10}h^4$, and $G \in P(n,h)$. Then*

$$\deg G \leq 0 \implies |G(a,b)| < \sqrt{b},$$
$$\deg G > 0 \implies |G(a,b)| > 2\sqrt{b}.$$

Proof. If $\deg G \leq 0$, then $G \in \mathbb{Q}$ and $|G| \leq h < \sqrt{b}$. Suppose $\deg G = d > 0$ with $d \leq n$. So $G = \frac{f(y)+g(y)x}{c}$ with $f, g \in \mathbb{Z}[Y]$,

$$\|f\|, \|g\| \leq h, \quad \max(\deg f, 1 + \deg g) = d, \quad c \in \mathbb{Z}, 0 < c \leq h.$$

Put $F := f(y) + g(y)x$ and write

$$f = r_d Y^d + \cdots + r_0 \in \mathbb{Z}[Y], \quad r_0, \ldots, r_d \in \mathbb{Z},$$
$$g = s_{d-1}Y^{d-1} + \cdots + s_0 \in \mathbb{Z}[Y], \quad s_0, \ldots, s_{d-1} \in \mathbb{Z}.$$

Then we have, with the notations in the proof of Lemma 4.2,

$$F(a,b) = \sum_{i=0}^{d} c_i b^i, \quad c_i := r_i + s_{i-1}\frac{a}{b} \ \text{ for } i = 0, \ldots, d,$$

where $s_{-1} = 0$. Hence

$$|\sum_{i=0}^{d-1} c_i b^i| \leq 3h \sum_{i=0}^{d-1} b^i \leq 3nhb^{d-1},$$

while $|c_d b^d| > \frac{b^d}{6h}$, and thus

$$|G(a,b)| \geq \frac{|F(a,b)|}{h} \geq \frac{\big(b^d/6h\big) - 3nhb^{d-1}}{h}$$
$$= b^d\big(\frac{1}{6h^2} - \frac{3n}{b}\big) > 2\sqrt{b}. \quad \square$$

Put $A(n,h) := \mathbb{Z} \cap \{F(a,b) : F \in P(n,h)\}$. It is easy to check that
(i) $A(n,h) + A(n,h) \subseteq A(n, 2h^2)$,
(ii) $A(n,h) \cdot A(n,h) \subseteq A(2n, (4n+1)h^2)$.

LEMMA 4.4. *Let $n \geq 1$, $h \geq 4n+1$, $b \geq (2h)^{50n}$, and $s, t \in A(n,h)$. Then*

$$\operatorname{iq}(s,t), \operatorname{rem}(s,t) \in A\big(3n-1, (2h)^{11n}\big).$$

PROOF. Write $s = \frac{f(b)+g(b)a}{c}$ and $t = \frac{k(b)+l(b)a}{d}$ with $f, g, k, l \in \mathbb{Z}[Y]$,

$$\|f\|, \|g\|, \|k\|, \|l\| \leq h, \quad \deg f, \deg k \leq n, \quad \deg g, \deg l \leq n-1,$$

and $c, d \in \mathbb{Z}$, $0 < c, d \leq h$. The conclusion of the lemma holds trivially if $t = 0$, so we shall assume below that $t \neq 0$.

We first treat the simple case $\deg(k(y) + l(y)x) = 0$. Then $l = 0$ and $k \in \mathbb{Z}$, so $|t| \leq h$, hence $|\operatorname{rem}(s,t)| < h$, and thus

$$\operatorname{iq}(s,t) = \frac{s - \operatorname{rem}(s,t)}{t} = \frac{\big(f(b) - \operatorname{rem}(s,t)c\big) + g(b)a}{ct},$$

from which we get

$$\operatorname{iq}(s,t) \in A(n, h+h^2) \subseteq A\big(3n-1, (2h)^{11n}\big),$$
$$\operatorname{rem}(s,t) \in A(0,h) \subseteq A\big(3n-1, (2h)^{11n}\big).$$

For the rest of the proof we assume that $\deg(k(y) + l(y)x) > 0$. Multiplying s and t by $k(b) - l(b)a$ yields

$$\frac{s}{t} = \frac{d[f(b)k(b) - g(b)l(b)(2b^2+1) + \big(g(b)k(b) - f(b)l(b)\big)a]}{c[k(b)^2 - l(b)^2(2b^2+1)]}.$$

So with $N := 2n$ and $H := (4n+1)h^3$, we have $\frac{s}{t} = \frac{\phi(b)+\psi(b)a}{\theta(b)}$ where $\phi, \psi, \theta \in \mathbb{Z}[Y]$, $\theta(b) \neq 0$, and

$$\deg\phi \leq N, \quad \deg\psi \leq N-1, \quad 0 < \deg\theta = 2\deg(k(y) + l(y)x) \leq N,$$
$$\|\phi\| \leq H, \quad \|\psi\| \leq H, \quad \|\theta\| \leq H.$$

We can also assume that the leading coefficient ℓ of θ is positive.

If $\deg(\phi(y) + \psi(y)x) < \deg\theta$, then $|\phi(b) + \psi(b)a| < \theta(b)$ by Lemma 4.2, so either $\operatorname{iq}(s,t) = 0$ and $\operatorname{rem}(s,t) = s$, or $\operatorname{iq}(s,t) = -1$ and $\operatorname{rem}(s,t) = s + t$, and in either case we have the desired result. For the rest of the proof we can therefore assume that

$$N \geq \max\{\deg\phi, 1 + \deg\psi\} \geq \deg\theta > 0.$$

Pseudo-division of ϕ and ψ by θ—see Chapter 6 in [von zur Gathen and Gerhard 1999]—gives

$$\ell^N\phi = q_1\theta + r_1, \quad \ell^N\psi = q_2\theta + r_2$$

where $q_i, r_i \in \mathbb{Z}[Y]$, $\deg r_i < \deg\theta$ for $i = 1, 2$, and

$$\|q_1\| \leq 2^{N-1}H^N, \quad \|r_1\| \leq 2^N H^{N+1},$$
$$\|q_2\| \leq 2^{N-2}H^N, \quad \|r_2\| \leq 2^{N-1}H^{N+1}.$$

(Details: if $\deg\phi < \deg\theta$, put $q_1 = 0$ and $r_1 = \ell^N\phi$; if $\deg\phi \geq \deg\theta$, then $0 \leq \deg\phi - \deg\theta \leq N-1$, and we can use the bounds of Exercise 6.44 on p. 193 of [von zur Gathen and Gerhard 1999]; if $\deg\psi < \deg\theta$, put $q_2 = 0$ and $r_2 = \ell^N\psi$; if $\deg\psi \geq \deg\theta$, then $0 \leq \deg\psi - \deg\theta \leq N-2$, and we can use again the bounds of that same exercise.) Hence

$$\frac{\ell^N s}{t} = q_1(b) + q_2(b)a + \frac{r_1(b) + r_2(b)a}{\theta(b)}.$$

Let c_2 be the coefficient of $Y^{(\deg\theta)-1}$ in r_2. Then

$$|c_2\sqrt{2}| < 2^{N-1}H^{N+1}\sqrt{2} < 2^N H^{N+1}\ell = \text{leading coefficient of } 2^N H^{N+1}\theta.$$

Thus by Lemma 4.3 and the way $\mathbb{Q}[x,y]$ is ordered, we have

$$|r_1(b) + r_2(b)a| < 2^N H^{N+1}\theta(b),$$

provided $b \geq 80\big(2^N H^{N+2}\big)^4$. This last inequality on b follows from our assumption $b \geq (2h)^{50n}$ because $(2h)^{50n} \geq 80\big(2^N H^{N+2}\big)^4$. (To derive this last inequality, note that it is equivalent to

$$2^{50n}h^{50n} \geq 80 \cdot 2^{8n}(4n+1)^{8n+8}h^{24n+24},$$

that is, to $2^{42n}h^{26n-24} \geq 80 \cdot (4n+1)^{8n+8}$. In view of $h \geq 4n+1$, it is enough that $2^{42n}(4n+1)^{18n-32} \geq 80$, and this is easily verified for $n = 1$, and is obvious for $n > 1$.) From $|r_1(b) + r_2(b)a| < 2^N H^{N+1}\theta(b)$ we get

$$\left|\frac{r_1(b) + r_2(b)a}{\theta(b)}\right| < 2^N H^{N+1}.$$

Therefore, with $\rho := \lfloor \frac{r_1(b)+r_2(b)a}{\theta(b)} \rfloor$:

$$\mathrm{iq}(\ell^N s, t) = (q_1(b) + \rho) + q_2(b)a, \quad |\rho| \leq 2^N H^{N+1}.$$

Now $\mathrm{iq}(s,t) = \frac{\mathrm{iq}(\ell^N s,t) - i}{\ell^N}$ with $i \in \{0, \ldots, \ell^N - 1\}$, and $\ell^N \leq H^N$, so

$$\mathrm{iq}(s,t) = \frac{(q_1(b) + \rho - i) + q_2(b)a}{\ell^N},$$
$$\|q_1 + \rho - i\| \leq 2^{N-1}H^N + 2^N H^{N+1} + H^N.$$

Next, for the remainder we have

$$\begin{aligned}
\mathrm{rem}(s,t) &= s - \mathrm{iq}(s,t)t \\
&= \frac{f(b) + g(b)a}{c} - \frac{[(q_1(b) + \rho - i) + q_2(b)a][k(b) + l(b)a]}{d\ell^N} \\
&= \frac{d\ell^N f(b) + d\ell^N g(b)a}{cd\ell^N} - \frac{(q_1(b) + \rho - i)k(b) + q_2(b)l(b)(2b^2+1)}{d\ell^N} \\
&\qquad - \frac{[(q_1(b) + \rho - i)l(b) + q_2(b)k(b)]a}{d\ell^N} \\
&= \frac{d\ell^N f(b) - c\big(q_1(b) + \rho - i\big)k(b) - cq_2(b)l(b)(2b^2+1)}{cd\ell^N} \\
&\qquad + \frac{[d\ell^N g(b) - c\big(q_1(b) + \rho - i\big)l(b) - cq_2(b)k(b)]a}{cd\ell^N}.
\end{aligned}$$

Straightforward estimates using the last expression and earlier bounds yield

$$\mathrm{rem}(s,t) \in A(3n-1, (n+1)2^{N+1}h^2H^{N+1}).$$

Thus $\mathrm{rem}(s,t) \in A(3n-1, (2h)^{11n})$, provided $(n+1)2^{N+1}h^2H^{N+1} \le (2h)^{11n}$. This last inequality is equivalent to

$$(n+1)2^{2n+1}(4n+1)^{2n+1}h^{6n+5} \le 2^{11n}h^{11n},$$

that is, to $(n+1)(4n+1)^{2n+1} \le 2^{9n-1}h^{5n-5}$, which in view of $h \ge 4n+1$ holds if $n+1 \le 2^{9n-1}(4n+1)^{3n-6}$: this last inequality is easily verified for $n = 1$, and is obvious for $n > 1$. □

Corollary 4.5. *Suppose that* $b \ge 2^{2^{10m^2}}$. *Then*

$$G^{\times}_m(a+1,b) \subseteq A(3^m, 2^{2^{4m^2+4}}),$$

Also, $\gcd(a+1,b) \notin G^{\times}_m(a+1,b)$ *for* $m > 0$.

Proof. The inclusion clearly holds for $m = 0$. Next, we have

$$2^{2^{4(m+1)^2+4}} > \left(2 \cdot 2^{2^{4m^2+4}}\right)^{11\cdot 3^m}$$
$$2^{2^{10(m+1)^2}} > \left(2 \cdot 2^{2^{4m^2+4}}\right)^{50\cdot 3^m},$$

inequalities that are best checked for $m = 0$ by inspection, and are easily proved for $m > 0$. With these inequalities, the inclusion follows by induction on m using the previous lemma.

In view of the inequality right after the proof of Lemma 3.2, the claim about $\gcd(a+1,b)$ follows from Lemma 4.3 (with $n := 3^m$, $h := 2^{2^{4m^2+4}}$, $m > 0$). Thus $\gcd(a+1,b) \notin A(n,h)$. □

Proof of Theorem 4.1. The non-membership result about $\gcd(a+1,b)$ in this corollary means that $\mathrm{g}^{\times}(a+1,b) > m$ whenever $b \ge 2^{2^{10m^2}}$, $m > 0$, that is, whenever $b \ge 2^{2^{10}}$ and $0 < m \le \frac{1}{\sqrt{10}}\sqrt{\log\log b}$. It follows that $\mathrm{g}^{\times}(a+1,b) > \frac{1}{\sqrt{10}}\sqrt{\log\log b}$ if $b \ge 2^{2^{10}}$, which yields the theorem for $b \ge 2^{2^{10}}$; the result holds trivially for $2 \le b < 2^{2^{10}}$. □

PART 2. LOWER BOUNDS FOR (NON-UNIFORM) TERM COMPLEXITY

The basic idea in Part 1 was that the computation of $\gcd(\alpha,\beta)$ cannot take fewer steps than those needed to construct the *value* $\gcd(\alpha,\beta)$ from α and β using the primitives, cf. the remarks at the end of the introduction to Part 1. In order to derive lower bounds for *arithmetic decision problems* which take values in $\{0,1\}$, we require more precise notions about algorithms. We introduce these in Section 5 for simple algorithms given by explicit terms with conditionals and equality tests, and we derive lower bounds for these algorithms over initial segments of $\mathbb{N}$ in Sections 6 – 10. In Part 3 we derive corresponding *uniform* lower bounds for algorithms expressed by *recursive programs.* The idea in both of these parts is to establish lower bounds on complexity using *embeddings*, roughly as automorphisms are used to establish non-definability in a first order language.

In connection with the non-uniform term algorithms of this Part, it is worth noting that, because of the "lookup" algorithm, we cannot expect better than *logarithmic* lower bounds, from any but the most trivial primitives. Lemma 5.2 states a precise version of this well-known fact.

5. COMPUTATION IN PARTIAL ALGEBRAS AND TERM-COMPLEXITY

An adequate notion of *computation* demands that the function computed by an algorithm may be *partial* as opposed to *total*. Moreover, certain basic functions like "division by 2 on $\mathbb{Z}$" are partial by nature. We write $f : X \rightharpoonup Y$ to indicate that f is a *partial function from X to Y*, that is, f is a function with domain $D(f) \subseteq X$ and codomain Y. This includes of course (total) functions $f : X \to Y$. We write $f(x)\!\downarrow$ (read "$f(x)$ converges") to indicate that $x \in D(f)$. In some places, we will also appeal to the natural partial ordering of partial functions,

$$f \sqsubseteq g \iff (\forall x \in X)[f(x)\!\downarrow \implies f(x) = f(y)] \qquad (f, g : X \rightharpoonup Y).$$

Partial algebras

Let Φ be a *signature*, that is, a set of function symbols, each equipped with an arity in $\mathbb{N}$, and let

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

a Φ-*algebra with* $0, 1$; thus $0^{\mathbf{A}}$ and $1^{\mathbf{A}}$ are distinguished elements of the set A with $0^{\mathbf{A}} \neq 1^{\mathbf{A}}$, and $\phi^{\mathbf{A}} : A^n \rightharpoonup A$ is a partial n-ary function on A for $\phi \in \Phi$ of arity n.

In this setting we deal with relations on A via their (total) characteristic functions: the characteristic function $\chi_R : A^n \to \{0, 1\} \subseteq A$ of an n-ary relation $R \subseteq A^n$ is given by $\chi_R(a) = 1$ if $a \in R$ and $\chi_R(a) = 0$ otherwise.

We often omit the superscript $\mathbf{A}$ in $0^{\mathbf{A}}$, $1^{\mathbf{A}}$ and $\phi^{\mathbf{A}}$ when no confusion with the corresponding symbols 0, 1 and ϕ (for $\phi \in \Phi$) is likely. The distinction between these symbols and their interpretations in $\mathbf{A}$, whether or not shown on paper, should of course be kept in mind.

Examples

The two algebras with $0, 1$ encountered so far are

$$\begin{aligned}\mathbb{Z}(1) &:= \big(\mathbb{Z}, 0, 1, +, -, \mathrm{iq}, \mathrm{rem}, <\big) \quad \text{and}\\ \mathbb{Z}(2) &:= \big(\mathbb{Z}, 0, 1, +, -, \cdot, \mathrm{iq}, \mathrm{rem}, <\big).\end{aligned} \tag{4}$$

In referring to these structures as *algebras* we mean that $<$ designates here the characteristic function of the usual (strict) ordering on $\mathbb{Z}$.

For each integer $d > 0$, let $\mathrm{iq}(-, d)$ denote the function $x \mapsto \mathrm{iq}(x, d) : \mathbb{Z} \to \mathbb{Z}$. In the next section we consider for each integer $D > 1$ the following algebra with $0, 1$:

$$\mathbb{Z}(\mathrm{div}_D) := \big(\mathbb{Z}, 0, 1, +, -, \mathrm{iq}(-, 1), \mathrm{iq}(-, 2), \ldots, \mathrm{iq}(-, D), <\big). \tag{5}$$

The algebras $\mathbb{Z}(1)$, $\mathbb{Z}(2)$ and $\mathbb{Z}(\mathrm{div}_D)$ are infinite and total, but their finite *subalgebras* (defined later in this section) are genuinely partial, and play a key role in the rest of the paper.

Objectifying the undefined

For each set A, choose an object $\uparrow \notin A$ to represent "the undefined", and let $A_\uparrow := A \cup \{\uparrow\}$. Each partial function $f : A^n \rightharpoonup A$ extends naturally to a (total) function

$$f_\uparrow : (A_\uparrow)^n \to A_\uparrow$$

by setting $f_\uparrow(a) = \uparrow$ for each $a \in A^n_\uparrow \setminus D(f)$. We call $f_\uparrow$ the *up-arrow extension* of f. Notice that for $a \in A^n$, $f(a)\!\downarrow \iff f_\uparrow(a) \in A$.

Logical symbols

We introduce two symbols Eq and Co. The *equality symbol* Eq is interpreted in $\mathbf{A}$ by the function

$$\mathrm{Eq}^{\mathbf{A}} \;:\; A^2_\uparrow \to \{0, 1, \uparrow\} \subseteq A_\uparrow$$

given by $\mathrm{Eq}^{\mathbf{A}}(a,b) = 1$ if $a, b \in A$ and $a = b$, $\mathrm{Eq}^{\mathbf{A}}(a,b) = 0$ if $a, b \in A$ and $a \neq b$, and $\mathrm{Eq}^{\mathbf{A}}(a,b) = \uparrow$ if $a = \uparrow$ or $b = \uparrow$. The *conditional symbol* Co is thought of as "if ..., then ..., else ..." and it is interpreted in $\mathbf{A}$ by the ternary function $\mathrm{Co}^{\mathbf{A}} : A^3_\uparrow \to A_\uparrow$ given by

$$\mathrm{Co}^{\mathbf{A}}(a,b,c) = b \;\text{ if } a = 1, \quad \mathrm{Co}^{\mathbf{A}}(a,b,c) = c \;\text{ if } a \in A, a \neq 1,$$
$$\mathrm{Co}^{\mathbf{A}}(a,b,c) = \uparrow \;\text{ if } a = \uparrow.$$

Again, we often omit the superscript $\mathbf{A}$ in $\mathrm{Eq}^{\mathbf{A}}$ and $\mathrm{Co}^{\mathbf{A}}$ when no confusion with the symbols Eq and Co is likely.

Notice that $\mathrm{Eq}^{\mathbf{A}}$ is the up-arrow extension of the characteristic function of the equality relation on A, but $\mathrm{Co}^{\mathbf{A}}$ is not the up-arrow extension of any partial function on A, because it takes values in A on some inputs which include the object $\uparrow$, e.g., $\mathrm{Co}(1, 0, \uparrow) = 0$. (In computer science terminology, Co is a *non-strict function* on $A_\uparrow$.) We consider $0, 1, \mathrm{Eq}, \mathrm{Co}$ as four distinct (logical) symbols, not among the symbols in Φ and independent of $\mathbf{A}$. In constructing Φ-terms (see below) we consider $0, 1$ as constant (nullary) function symbols, Eq as a binary function symbol, and Co as a ternary function symbol.

In addition we fix once and for all infinitely many *variables*: symbols of arity 0, not in the set $\{0, 1, \mathrm{Eq}, \mathrm{Co}\} \cup \Phi$.

Terms with equality tests and conditionals

A *Φ-term with equality tests and conditionals* is a word on the alphabet

$$\{\text{variables}\} \cup \{0, 1, \mathrm{Eq}, \mathrm{Co}\} \cup \Phi$$

formed according to the usual syntactic rules that take into account the arity of the symbols.

For brevity we write "Φ-term" instead of "Φ-term with equality tests and conditionals". The *length* of a Φ-term is its length as a word. So the Φ-terms of length one are $0, 1$ and the words x and ϕ where x is a variable and $\phi \in \Phi$ has arity 0; the Φ-terms of length > 1 are exactly the words $\phi t_1 \dots t_n$ where $\phi \in \{\mathrm{Eq}, \mathrm{Co}\} \cup \Phi$ has arity $n > 0$ and $t_1, \dots, t_n$ are Φ-terms. For better readability we often write such a Φ-term $\phi t_1 \dots t_n$ as $\phi(t_1, \dots, t_n)$.

Term evaluation

Below x, y denote distinct variables, $\vec{x}$ denotes a tuple $(x_1, \dots, x_m)$ of distinct variables, and so does $\vec{y} = (y_1, \dots, y_n)$. In referring to a Φ-term $t(\vec{x})$ we mean a Φ-term t together with a tuple $\vec{x} = (x_1, \dots, x_m)$ such that all variables occurring in t are among $x_1, \dots, x_m$. Such a Φ-term $t(\vec{x})$ defines a function

$$\vec{a} \mapsto t(\vec{a}) : \; A^m_{\uparrow} \to A_{\uparrow}, \qquad \vec{a} = (a_1, \dots, a_m)$$

as follows: if t is x_i, then $t(\vec{a}) := a_i$; if t is $0, 1$, then $t(\vec{a}) := 0, 1$, respectively; if t is $\mathrm{Eq}(t_1, t_2)$, then $t(\vec{a}) := \mathrm{Eq}(t_1(\vec{a}), t_2(\vec{a}))$; if $t = \mathrm{Co}(t_1, t_2, t_3)$, then $t(\vec{a}) := \mathrm{Co}(t_1(\vec{a}), t_2(\vec{a}), t_3(\vec{a}))$; if $t = \phi(t_1, \dots, t_n)$ with n-ary $\phi \in \Phi$, then $t(\vec{a}) := \phi_{\uparrow}(t_1(\vec{a}), \dots, t_n(\vec{a}))$.

We only care about the value of $t(\vec{a})$ when all components a_i lie in A, but have to allow $a_i = \uparrow$ because of the inductive nature of the definition. If it is necessary to indicate the Φ-algebra $\mathbf{A}$ in which $t(\vec{a})$ is computed, we write $t^{\mathbf{A}}(\vec{a})$ instead of $t(\vec{a})$.

Given a Φ-algebra $\mathbf{A}$, we also refer to a Φ-term $t(\vec{x})$ as an $\mathbf{A}$-term.

Boolean operations

We can express *boolean* operations by Φ-terms as follows. Let $\neg(x)$, $\vee(x, y)$ and $\wedge(x, y)$ be the Φ-terms defined by

$$\neg(x) := \mathrm{Co}(x, 0, 1), \;\; \vee(x, y) := \mathrm{Co}(x, 1, y), \;\; \wedge(x, y) := \mathrm{Co}(x, y, 0).$$

Then we have, with $0, 1 \in A$:

$$\begin{aligned} \neg(0) = 1, &\quad \neg(1) = 0, \\ \vee(0,0) = 0, &\quad \vee(0,1) = \vee(1,0) = \vee(1,1) = 1, \\ \wedge(0,0) = \wedge(0,1) = \wedge(1,0) = 0, &\quad \wedge(1,1) = 1. \end{aligned}$$

Terms as algorithms

Let $t(\vec{x})$ be an $\mathbf{A}$-term, $g : A^m_{\uparrow} \to A_{\uparrow}$ a function on the up-arrow extension of A, and $S \subseteq A^m_{\uparrow}$. We say that $t(\vec{x})$ *computes* g *on* S if $t(\vec{a}) = g(\vec{a})$ for all $\vec{a} \in S$.

We are primarily interested in the case when $S \subseteq A^m$ and $g = f_{\uparrow}$ is the up-arrow extension of a partial function $f : A^m \rightharpoonup A$, in which case we will say simply that "$t(\vec{x})$ computes f on S". (But it is useful to have the more general notion.) If $S \subseteq A^m$ and f is the characteristic function of an m-ary relation $R \subseteq A^m$, we also say that "$t(\vec{x})$ decides" (rather than "computes") R on S.

We think of terms as expressing algorithms of a particularly explicit nature. In Part 3 we will consider McCarthy programs which express more general (recursive) algorithms; by Proposition 11.4 terms define exactly the recursive algorithms of bounded complexity.

Depth and arithmetic depth

The *depth* of a Φ-term t is the natural number defined inductively as follows:

$$\begin{aligned}
&\mathrm{depth}(t) := 0 \ \text{ if } t \text{ is } 0,\ 1 \text{ or a variable},\\
&\mathrm{depth}(t) := 1 + \max\{\mathrm{depth}(t_1),\dots,\mathrm{depth}(t_n)\} \ \text{ if } t = \phi(t_1,\dots,t_n)\\
&\qquad\qquad\quad \text{where } \phi \in \{\mathrm{Eq},\mathrm{Co}\} \cup \Phi \text{ has arity } n.
\end{aligned}$$

Here and below we set $\max\emptyset := 0$, so $\mathrm{depth}(t) = 1$ if t is a constant symbol in Φ.

We modify this depth to arithmetic depth by taking into account only function symbols in Φ. The *arithmetic depth* of a Φ-term t is the natural number defined inductively as follows:

$$\begin{aligned}
&\mathrm{depth}_a(t) := 0 \ \text{ if } t \text{ is a variable or } 0 \text{ or } 1,\\
&\mathrm{depth}_a(t) := \max\{\mathrm{depth}_a(t_1),\dots,\mathrm{depth}_a(t_n)\}\\
&\qquad\qquad\quad \text{if } t = \phi(t_1,\dots,t_n) \text{ where } \phi \in \{\mathrm{Eq},\mathrm{Co}\} \text{ has arity } n,\\
&\mathrm{depth}_a(t) := 1 + \max\{\mathrm{depth}_a(t_1),\dots,\mathrm{depth}_a(t_n)\}\\
&\qquad\qquad\quad \text{if } t = \phi(t_1,\dots,t_n) \text{ where } \phi \in \Phi \text{ has arity } n.
\end{aligned}$$

In particular, $\mathrm{depth}_a(t) = 1$ if t is a constant symbol in Φ.

Embeddings

Let $\mathbf{B} = (B, 0, 1, \dots)$ be a second Φ-algebra.

A map $\imath : A \to B$ is said to be an *embedding of* $\mathbf{A}$ *into* $\mathbf{B}$ if $\imath(0) = 0$, $\imath(1) = 1$ and for each $\phi \in \Phi$ of arity n and $\vec{a} \in D(\phi^{\mathbf{A}})$ we have $\imath(\vec{a}) := \big(\imath(a_1),\dots,\imath(a_n)\big) \in D(\phi^{\mathbf{B}})$ and $\imath\big(\phi(\vec{a})\big) = \phi\big(\imath(\vec{a})\big)$. We write $\imath : \mathbf{A} \rightarrowtail \mathbf{B}$ to indicate that $\imath$ is an embedding of $\mathbf{A}$ into $\mathbf{B}$. One easily verifies that for such an embedding $\imath$, a Φ-term $t(\vec{x})$, and $\vec{a} \in A^m$, we have

$$t^{\mathbf{A}}(\vec{a})\downarrow \implies \imath\big(t^{\mathbf{A}}(\vec{a})\big) = t^{\mathbf{B}}\big(\imath(\vec{a})\big).$$

In case $A \subseteq B$ and the inclusion map $A \hookrightarrow B$ is an embedding $\mathbf{A} \rightarrowtail \mathbf{B}$ we call $\mathbf{A}$ a *subalgebra of* $\mathbf{B}$. For each set $U \subseteq A$ with $0, 1 \in U$ we obtain a subalgebra $\mathbf{A} \restriction U$ of $\mathbf{A}$ by taking U as its underlying set, and requiring $D(\phi^{\mathbf{A}\restriction U}) = \{\vec{a} \in U^n : \vec{a} \in D(\phi^{\mathbf{A}}),\ \phi^{\mathbf{A}}(\vec{a}) \in U\}$ for each $\phi \in \Phi$ of arity n. (Notice that $\phi^{\mathbf{A}\restriction U}$ may not be a total function, even if $\phi^{\mathbf{A}}$ is total.)

All proofs of lower bounds in later sections use embeddings of algebras $\mathbf{A} \restriction U$ into $\mathbf{A}$ which are not given by the inclusion $U \hookrightarrow A$.

Composition of terms

We compose terms as follows: given a Φ-term $t(\vec{x})$ and Φ-terms $\tau_1(\vec{y}),\dots,\tau_m(\vec{y})$, we let $t(\tau_1(\vec{y}),\dots,\tau_m(\vec{y}))$ be the Φ-term $t^*(\vec{y})$ obtained from t by replacing each occurrence of a variable x_i in t by τ_i. One verifies easily by induction on $\mathrm{depth}(t)$ that then

$$\begin{gathered}
t^*(\vec{b}) = t(\tau_1(\vec{b}),\dots,\tau_m(\vec{b})) \text{ for } \vec{b} \in A^n_{\uparrow}\\
\mathrm{depth}(t(\tau_1(\vec{y}),\dots,\tau_m(\vec{y})) \le \mathrm{depth}(t) + \max\{\mathrm{depth}(\tau_1),\dots,\mathrm{depth}(\tau_m)\}\\
\mathrm{depth}_a(t(\tau_1(\vec{y}),\dots,\tau_m(\vec{y})) \le \mathrm{depth}_a(t) + \max\{\mathrm{depth}_a(\tau_1),\dots,\mathrm{depth}_a(\tau_m)\}
\end{gathered}$$

Arithmetic depth and generation

For $X \subseteq A$ and $k \in \mathbb{N}$ we define the set $G_k^{\mathbf{A}}(X) \subseteq A$ generated by X in k steps in the obvious way:

$$G_0^{\mathbf{A}}(X) :=\{0,1\} \cup X$$
$$G_{k+1}^{\mathbf{A}}(X) :=G_k^{\mathbf{A}}(X) \cup \{\phi(\vec{b}) : \phi \in \Phi \text{ has arity } n,\ \vec{b} \in D(\phi) \cap G_k^{\mathbf{A}}(X)^n\}.$$

Note that $G_0^{\mathbf{A}}(X) \subseteq G_1^{\mathbf{A}}(X) \subseteq G_2^{\mathbf{A}}(X) \subseteq \dots$, and $G_k^{\mathbf{A}}(G_\ell^{\mathbf{A}}(X)) = G_{k+\ell}^{\mathbf{A}}(X)$.

Let $\vec{a} = (a_1, \dots, a_m) \in A^m$ and put $G_k^{\mathbf{A}}(\vec{a}) := G_k^{\mathbf{A}}(\{a_1, \dots, a_m\})$. An easy induction on k yields

Lemma 5.1. *$G_k^{\mathbf{A}}(\vec{a}) = A \cap \{t(\vec{a}) : t(\vec{x})$ is a $\mathbf{A}$-term with $\mathrm{depth}_a(t) \leq k\}$. If $G_k^{\mathbf{A}}(\vec{a}) \subseteq U \subseteq A$, then for each Φ-term $t(\vec{x})$ with $\mathrm{depth}_a(t) \leq k$ we have $t^{\mathbf{A}\restriction U}(\vec{a}) = t^{\mathbf{A}}(\vec{a})$.*

The second part of this lemma states a useful absoluteness property. Note that with the notations from Sections 3 and 4 we have

$$G_k^{\mathbb{Z}(1)}(a,b) = G_k(a,b), \quad G_k^{\mathbb{Z}(2)}(a,b) = G_k^{\times}(a,b), \qquad a,b \in \mathbb{Z}.$$

Examples

(i) Let $\mathbf{A} = (\mathbb{N}, 0, 1, +)$ and $a \in \mathbb{N}^{>0}$. We claim that there is an $\mathbf{A}$-term $t(x)$ such that $t(a) = a^2$ and $\mathrm{depth}_a(t) \leq 2 \log a$.

To see this, write $a = 2^{e(0)} + \dots + 2^{e(k)}$ with integers $0 \leq e(0) < \dots < e(k)$, so $k \leq e(k) \leq \log a < e(k) + 1$. Iterated doubling yields $2^e a \in G_e^{\mathbf{A}}(a)$ for all $e \in \mathbb{N}$, in particular, $2^{e(0)}a, \dots, 2^{e(k)}a \in G_{e(k)}^{\mathbf{A}}(a)$. Adding these numbers yields $a^2 \in G_{e(k)+k}^{\mathbf{A}}(a)$. Now use that $e(k) + k \leq 2 \log a$.

This upper bound is matched by a similar lower bound: $\mathrm{depth}_a(t) \geq \log a$ for each $\mathbf{A}$-term $t(x)$ such that $t(a) = a^2$. (To see this, note that by induction on $e \in \mathbb{N}$ we have $\max G_e^{\mathbf{A}}(a) = 2^e a$.) This lower bound remains valid for $\mathbf{A} = (\mathbb{Z}, 0, 1, +, -, \mathrm{iq}, \mathrm{rem}, \mathrm{gcd}, <)$ and $a \in \mathbb{N}^{>0}$ because addition and subtraction are the only primitives of $\mathbf{A}$ that can output numbers larger in absolute value than the inputs.

(ii) In connection with $\mathbf{A} = \mathbb{Z}(1)$, we can reformulate Theorem 3.1 as saying that whenever a, b are integers > 2 such that $a^2 - 2b^2 = 1$, then any $\mathbb{Z}(1)$-term $t(x,y)$ such that $t(a+1,b) = \gcd(a+1,b)$ must have arithmetic depth greater than $\frac{1}{6} \log \log a$. Theorem 4.1 can be reformulated in a similar way with lower bounds on the arithmetic depth of $\mathbb{Z}(2)$-terms computing the gcd of certain integers.

Lower bound arguments as in Example (i) are too crude to deal with arithmetic functions like $\lfloor \sqrt{n} \rfloor$. More refined results on growth functions as in Example (ii) still cannot handle decision problems where 0 and 1 are the only values to be determined. For such problems we need a more subtle approach.

Term-complexity

We define two non-uniform measures of complexity, one based on *depth*, the other on *arithmetic depth*.

Let $f : A^m \rightharpoonup A$ be a partial function and $S \subseteq A^m$. (In applications S will be finite.) If there is an $\mathbf{A}$-term $t(\vec{x})$ that computes f on S, then we let $c^{\mathbf{A}}(f, S)$ be the minimal depth of such a term, and call it the *term-complexity of computing f on S in* $\mathbf{A}$. If there is no such term we put $c^{\mathbf{A}}(f, S) = \infty$. If f is the characteristic function of an m-ary relation $R \subseteq A^m$, we also write $c^{\mathbf{A}}(R, S)$ for $c^{\mathbf{A}}(f, S)$ and call it the *term-complexity of deciding R on S*.

When in this definition we replace "depth" by "arithmetic depth", then *term-complexity* turns into *arithmetic term-complexity*, and the notations $c^{\mathbf{A}}(f, S)$ and $c^{\mathbf{A}}(R, S)$ must be replaced by $c_a^{\mathbf{A}}(f, S)$ and $c_a^{\mathbf{A}}(R, S)$. Note that $c_a^{\mathbf{A}}(f, S) \leq c^{\mathbf{A}}(f, S)$.

Usually we have $\mathbb{N} \subseteq A$ and $0^{\mathbf{A}}$ and $1^{\mathbf{A}}$ are the elements 0 and 1 of $\mathbb{N}$. In that case we write $c^{\mathbf{A}}(f, N)$ instead of $c^{\mathbf{A}}(f, S)$ for $S = \{1, \dots, N\}^m$, $N \in \mathbb{N}^{>0}$, and we use $c^{\mathbf{A}}(R, N)$, $c_a^{\mathbf{A}}(f, N)$ and $c_a^{\mathbf{A}}(R, N)$ in a similar way.

We have an easy logarithmic upper bound when addition is a primitive:

Lemma 5.2 The lookup term algorithm. *Given any relation $R \subseteq \mathbb{N}^m$ and any integer $N > 0$ we have $c_a^{\mathbf{A}}(R, N) \leq 2 \log N$, where $\mathbf{A} = (\mathbb{N}, 0, 1, +)$.*

Proof. For the simplest case $m = 1$, argue as in Example (i) above: For $n \geq 1$ we have $n = 2^{e(0)} + \cdots + 2^{e(k)}$, with $e(0) < e(1) < \cdots < e(k)$ and $e(k) + k \leq 2 \log n$, so n is the value in $\mathbf{A}$ of a closed term of arithmetic depth $\leq 2 \log n$, and hence the unary relation $x = n$ on $\mathbb{N}$ is decided by a term $t_n(x)$ of arithmetic depth $\leq 2 \log n$. It follows that R is decided on $\{1, \dots, N\}$ by the "disjunction" of the terms $t_n(x)$ for which $1 \leq n \leq N$ and $R(n)$ holds. The case $m > 1$ is handled in the same way. $\square$

In the next two sections we deal with the following situation: We are given an m-ary relation $R \subseteq A^m$ and a finite set $S \subseteq A^m$, and are interested in a lower bound on $c_a^{\mathbf{A}}(R, S)$ or $c^{\mathbf{A}}(R, S)$. In this connection the following is relevant.

Lemma 5.3. *Let $\vec{a} \in S$ be such that $R(\vec{a})$ holds. Suppose $G_k^{\mathbf{A}}(\vec{a}) \subseteq U \subseteq A$ ($k \in \mathbb{N}$), and $\imath : \mathbf{A}{\restriction}U \rightarrowtail \mathbf{A}$ is an embedding such that $\imath(\vec{a}) \in S$ and $\neg R(\imath(\vec{a}))$. Then $c_a^{\mathbf{A}}(R, S) > k$.*

Proof. Let $t(\vec{x})$ be an $\mathbf{A}$-term such that $t^{\mathbf{A}}(\vec{a}) = 1$ and $\mathrm{depth}_a(t) \leq k$. Then $t^{\mathbf{A}{\restriction}U}(\vec{a}) = 1$ by the absoluteness property of Lemma 5.1, hence $t^{\mathbf{A}}(\imath(\vec{a})) = 1$. But $\neg R(\imath(\vec{a}))$, so t cannot decide R on S. $\square$

This suggests how to go about obtaining lower bounds: construct an embedding $\imath$ as above with k as high as possible.

Robustness of term-complexity

Let $\mathbf{B} = \big(B, 0^{\mathbf{B}}, 1^{\mathbf{B}}, (\psi^{\mathbf{B}})_{\psi \in \Psi}\big)$ be a second partial algebra with $0, 1$ such that $A = B$, $0^{\mathbf{A}} = 0^{\mathbf{B}}$, $1^{\mathbf{A}} = 1^{\mathbf{B}}$. Then $\mathbf{A}$ is said to be *term-reducible to* $\mathbf{B}$ if each n-ary $\phi^{\mathbf{A}}_{\uparrow}$ with $\phi \in \Phi$ is computable by a $\mathbf{B}$-term on $A^n_{\uparrow}$. Note that the relation of term-reducibility is reflexive and transitive. The corresponding equivalence relation is called *term-equivalence*, that is, $\mathbf{A}$ and $\mathbf{B}$ are term-equivalent if and only if $\mathbf{A}$ is term-reducible to $\mathbf{B}$ and $\mathbf{B}$ is term-reducible to $\mathbf{A}$. In that case $\mathbf{A}$ and $\mathbf{B}$ are essentially equivalent from the term-complexity viewpoint. More precisely, we have the following robustness or invariance feature of term-complexity.

LEMMA 5.4. *Suppose Φ is finite and $\mathbf{A}$ is term-reducible to $\mathbf{B}$. Then there are positive integers $c(\mathbf{A},\mathbf{B}), c_a(\mathbf{A},\mathbf{B})$ such that for all $f : A^n \rightharpoonup A$ and $S \subseteq A^n$ we have*

$$c^{\mathbf{B}}(f,S) \le c(\mathbf{A},\mathbf{B}) \cdot c^{\mathbf{A}}(f,S) \quad \textit{and} \quad c_a^{\mathbf{B}}(f,S) \le c_a(\mathbf{A},\mathbf{B}) \cdot c_a^{\mathbf{A}}(f,S).$$

PROOF. For each m-ary $\phi \in \Phi$, let $t_\phi(\vec{x})$ be a $\mathbf{B}$-term that computes ϕ on $A^m_\uparrow$. Take positive integers $c(\mathbf{A},\mathbf{B})$ and $c_a(\mathbf{A},\mathbf{B})$ such that $c(\mathbf{A},\mathbf{B}) \ge \operatorname{depth}(t_\phi)$ and $c_a(\mathbf{A},\mathbf{B}) \ge \operatorname{depth}_a(t_\phi)$ for all $\phi \in \Phi$.

We define inductively for each $\mathbf{A}$-term $t(\vec{y})$ a $\mathbf{B}$-term ${}^{\mathbf{B}}t(\vec{y})$ as follows: if $t(\vec{y})$ is the variable y_j, then ${}^{\mathbf{B}}t(\vec{y})$ is y_j; if $t(\vec{y})$ is $0, 1$, then ${}^{\mathbf{B}}t(\vec{y})$ is $0, 1$, respectively; if $t(\vec{y}) = \phi\big(t_1(\vec{y}), \dots, t_k(\vec{y})\big)$ with k-ary $\phi \in \{\mathrm{Eq}, \mathrm{Co}\}$, then ${}^{\mathbf{B}}t(\vec{y}) := \phi\big({}^{\mathbf{B}}t_1(\vec{y}), \dots, {}^{\mathbf{B}}t_k(\vec{y})\big)$; if $t(\vec{y}) = \phi\big(t_1(\vec{y}), \dots, t_m(\vec{y})\big)$ with m-ary $\phi \in \Phi$, then ${}^{\mathbf{B}}t(\vec{y}) := t_\phi\big({}^{\mathbf{B}}t_1(\vec{y}), \dots, {}^{\mathbf{B}}t_m(\vec{y})\big)$.

An obvious induction using the facts on composition of terms then yields $t(\vec{a}) = {}^{\mathbf{B}}t(\vec{a})$ for all $\vec{a} \in A^n_\uparrow$ and

$$\operatorname{depth}({}^{\mathbf{B}}t) \le c(\mathbf{A},\mathbf{B}) \cdot \operatorname{depth}(t) \quad \text{and} \quad \operatorname{depth}_a({}^{\mathbf{B}}t) \le c_a(\mathbf{A},\mathbf{B}) \cdot \operatorname{depth}_a(t).$$

The lemma now follows easily. □

Circuits

One could also introduce *circuits*, directed acyclic graphs whose nodes are labeled by either a variable (input gate), or a logical symbol (logic gate), or an element of Φ (arithmetic gate). Circuits do the same job as terms, but can be much smaller in size as measured by the *number of edges of a circuit* versus the *length of a term*; see [Poizat 1995]. So far our methods only yield lower bounds on computational *depth*, and circuits perform no better than terms when depth is the relevant parameter. Therefore we do not consider circuits in this paper.

6. LOWER BOUNDS WITH LINEAR OPERATIONS AS PRIMITIVES

Throughout this section and the rest of Part 2 we let a, b, c, d, e, sometimes with subscripts, denote integers, while κ, λ, p, N range over *positive* integers.

Presburger functions

Presburger arithmetic refers to the first order theory of the structure $(\mathbb{Z}, 0, 1, +, -, <)$, the ordered group of integers with 1 as distinguished element. A classical result due to Presburger (and Skolem, see [Smoryński 1991]) is that this structure has elimination of quantifiers when we admit as extra primitives the unary relations $2\mathbb{Z}, 3\mathbb{Z}, 4\mathbb{Z}, 5\mathbb{Z}, \dots$.

This means the following. A *basic Presburger set in* $\mathbb{Z}^n$ is by definition an intersection of finitely many sets of the form

$$\{\vec{a} \in \mathbb{Z}^n : q(\vec{a}) = 0\}, \ \{\vec{a} \in \mathbb{Z}^n : q(\vec{a}) > 0\}, \ \text{and} \ \{\vec{a} \in \mathbb{Z}^n : q(\vec{a}) \in d\mathbb{Z}\},$$

where $q(X_1, \dots, X_n) \in \mathbb{Z}[X_1, \dots, X_n]$ is a polynomial of degree at most 1, and $d > 0$. A *Presburger set in* $\mathbb{Z}^n$ is by definition a finite union of basic Presburger sets in $\mathbb{Z}^n$.

Then a subset of $\mathbb{Z}^n$ is (first-order) definable in $(\mathbb{Z}, 0, 1, +, -, <)$ iff it is a Presburger set in $\mathbb{Z}^n$. Also, a partial function $f : \mathbb{Z}^n \rightharpoonup \mathbb{Z}$ is definable in $(\mathbb{Z}, 0, 1, +, -, <)$

iff its domain $D(f)$ is a disjoint union of of basic Presburger sets $B_1, \dots, B_k$ in $\mathbb{Z}^n$ and there are polynomials $q_1(X), \dots, q_k(X) \in \mathbb{Q}[X]$, $X = (X_1, \dots, X_n)$, of degree at most 1 such that $f(\vec{a}) = q_j(\vec{a})$ for all $\vec{a} \in B_j$, $j = 1, \dots, k$.

Let **Lin** denote the collection of all partial functions $\mathbb{Z}^n \rightharpoonup \mathbb{Z}$ (for all n) that are definable in $(\mathbb{Z}, 0, 1, +, -, <)$. (We choose this notation to reflect the piecewise *linear* nature of these functions.)

Let Φ be a *finite* subset of **Lin**, and let $\mathbf{A} := \big(\mathbb{Z}, 0, 1, (\phi)_{\phi \in \Phi}\big)$ be the corresponding partial algebra with $0, 1$.

Theorem 6.1. *Let $R \subseteq \mathbb{Z}$ be a unary relation, and let $F : \mathbb{N}^{>0} \to \mathbb{N}^{>0}$ be such that for all λ and p:*

(i) *$R(p) \implies \neg R(\kappa p)$ with $\kappa = 1 + \lambda F(\lambda)$,*

(ii) *$F(\lambda) \le d\lambda^e$, with $d, e > 0$ independent of λ.*

Then there is a $p_0 \in \mathbb{N}$ and a real constant $r > 0$ such that if $p > p_0$ and $R(p)$ holds, and $N := dp^{e+2}$, then

$$c_a^{\mathbf{A}}(R, N) > r \log N. \tag{6}$$

In particular, if $R(p)$ holds for infinitely many p, then (6) holds for infinitely many N.

The proof will show that p_0 and r can be chosen to depend only on $\mathbf{A}$ and on $(\mathbf{A}, d, e)$, respectively, and not on R.

Corollary 6.2. *The relations $R_i \subseteq \mathbb{Z}$ and functions F_i, $i = 1, \dots, 4$, defined below satisfy the hypothesis of the theorem, and so its conclusion:*

$$\begin{aligned}
R_1(x) &:\iff x > 0 \text{ and } x \text{ is prime}, & F_1(\lambda) &= 1\\
R_2(x) &:\iff x > 0 \text{ and } x \text{ is a power of } 2, & F_2(\lambda) &= 2\\
R_3(x) &:\iff x > 0 \text{ and } x \text{ is a perfect square}, & F_3(\lambda) &= \lambda\\
R_4(x) &:\iff x > 0 \text{ and } x \text{ is square-free}, & F_4(\lambda) &= 2 + \lambda.
\end{aligned}$$

Note that $(\mathbb{Z}, 0, 1, (\phi)_{\phi \in \Phi})$ is term-reducible to $\mathbb{Z}(\mathrm{div}_D)$ as defined in (5), where D is an integer > 1 depending on Φ. Thus by Lemma 5.4, it suffices to prove the Theorem for $\mathbf{A} = \mathbb{Z}(\mathrm{div}_D)$, and this is what we do below.

Lower bounds for $\mathbb{Z}(\mathrm{div}_D)$

Let D be an integer > 1, $E := D!$, and $\mathbf{A} := \mathbb{Z}(\mathrm{div}_D)$. Let $\vec{a} = (a_1, \dots, a_n)$, and, for $k \in \mathbb{N}$, put

$$B_k(\vec{a}) := \mathbb{Z} \cap \Big\{ \frac{c_0 + c_1 a_1 + \cdots + c_n a_n}{E^k} : |c_0|, \dots, |c_n| \le (2E)^k \Big\}.$$

Lemma 6.3. *We have the following inclusions for $k \in \mathbb{N}$:*

(i) *$B_k(\vec{a}) \subseteq B_{k+1}(\vec{a})$;*

(ii) *$B_k(\vec{a}) + B_k(\vec{a}) \subseteq B_{k+1}(\vec{a})$, $B_k(\vec{a}) - B_k(\vec{a}) \subseteq B_{k+1}(\vec{a})$;*

(iii) *$\mathrm{iq}\big(B_k(\vec{a}), d\big) \subseteq B_{k+1}(\vec{a})$ for $1 \le d \le D$;*

(iv) *$G_k^{\mathbf{A}}(\vec{a}) \subseteq B_k(\vec{a})$.*

Proof. The last inclusion follows from the earlier ones by induction and from $G_0^{\mathbf{A}}(\vec{a}) = \{0, 1, a_1, \dots, a_n\} \subseteq B_0(\vec{a})$.

Let $b \in B_k(\vec{a})$, and write

$$b = \frac{c_0 + c_1 a_1 + \cdots + c_n a_n}{E^k}, \qquad |c_0|, \dots, |c_n| \leq (2E)^k.$$

Then

$$b = \frac{Ec_0 + Ec_1 a_1 + \cdots + Ec_n a_n}{E^{k+1}}, \qquad |Ec_0|, \dots, |Ec_n| \leq (2E)^{k+1},$$

so $b \in B_{k+1}(\vec{a})$. Let $1 \leq d \leq D$ and $\mathrm{iq}(b, d) = (b - i)/d$, $0 \leq i < d$. Then

$$\mathrm{iq}(b, d) = \frac{(E/d)(c_0 - iE^k) + (E/d)c_1 a_1 + \cdots + (E/d)c_n a_n}{E^{k+1}}$$
$$\text{with } |(E/d)(c_0 - iE^k)|,\ |E/d)c_1|, \dots, |(E/d)c_n| \leq (2E)^{k+1},$$

so $\mathrm{iq}(b, d) \in B_{k+1}(\vec{a})$.

Let $b, b' \in B_k(\vec{a})$, write b as above and

$$b' = \frac{c_0' + c_1' a_1 + \cdots + c_n' a_n}{E^k}, \qquad |c_0'|, \dots, |c_n'| \leq (2E)^k.$$

Then

$$b + b' = \frac{E(c_0 + c_0') + E(c_1 + c_1')a_1 + \cdots + E(c_n + c_n')a_n}{E^{k+1}},$$
$$\text{with } |E(c_0 + c_0')|, \dots, |E(c_n + c_n')| \leq (2E)^{k+1},$$

so $b + b' \in B_{k+1}(\vec{a})$, and similarly with $b - b'$. □

Lemma 6.4. *Suppose* $|c_0|, |c_1| < a$. *Then*

(i) $c_0 + c_1 a = 0 \iff c_0 = c_1 = 0$;

(ii) $c_0 + c_1 a > 0 \iff$ *either* $c_1 > 0$, *or* ($c_1 = 0$ *and* $c_0 > 0$).

Suppose $|c_0|, |c_1|, |d_0|, |d_1| < \frac{a}{2}$. *Then*

(iii) $c_0 + c_1 a = d_0 + d_1 a \iff c_0 = d_0$ *and* $c_1 = d_1$;

(iv) $c_0 + c_1 a < d_0 + d_1 a \iff$ *either* $c_1 < d_1$, *or* ($c_1 = d_1$ *and* $c_0 < d_0$).

We leave (i) and (ii) as an easy exercise; note that (iii) and (iv) follow.

Lemma 6.5. *Let* $a > (D + 1)(2E)^m$, $\kappa \equiv 1 \mod E^m$, *and* $U := B_m(a)$. *Then we have an embedding* $\imath : \mathbf{A} \restriction U \rightarrowtail \mathbf{A}$ *given by*

$$\imath\Big(\frac{c_0 + c_1 a}{E^m}\Big) = \frac{c_0 + c_1 \kappa a}{E^m}, \quad |c_0|, |c_1| \leq (2E)^m. \tag{7}$$

Proof. By Lemma 6.4, (7) defines a mapping $\imath$, and it is quite easy to prove that it is an embedding $\imath : \mathbf{A} \restriction U \rightarrowtail \mathbf{A}$, using Lemma 6.4. We verify here only one item to explain why the factor $D + 1$ is included in the inequality on a. Let $c \in U \cap d\mathbb{Z}$ and $1 \leq d \leq D$ be such that $c/d \in U$; we need to check that then $\imath(c/d) = \imath(c)/d$. Write $c = (c_0 + c_1 a)/E^m$ and $c/d = (d_0 + d_1 a)/E^m$ with $|c_i|, |d_i| \leq (2E)^m$ for $i = 0, 1$. Then $(c_0 - d_0 d) + (c_1 - d_1 d)a = 0$, so $c_0 = d_0 d$ and $c_1 = d_1 d$ by the inequality on a and part (i) of Lemma 6.4. Hence $\imath(c/d) = \imath(c)/d$. □

The following result is a more precise version of Theorem 6.1 for the algebra $\mathbf{A} = \mathbb{Z}(\mathrm{div}_D)$.

Theorem 6.6. *Suppose $R \subseteq \mathbb{Z}$ is a unary relation, $F : \mathbb{N}^{>0} \to \mathbb{N}^{>0}$ is a function and $d, e > 0$ are such that for all λ and p:*

(i) *$R(p) \implies \neg R(\kappa p)$ with $\kappa = 1 + \lambda F(\lambda)$,*

(ii) *$F(\lambda) \le d\lambda^e$.*

Then for all $p > D + 1$, if $R(p)$ holds and $N := dp^{e+2}$, then

$$c_a^{\mathbf{A}}(R, N) > r \log N, \ \textit{where } r = \frac{1}{(2e + 4 + \log d)\log 2E}. \tag{8}$$

In particular, if $R(p)$ is true for infinitely many p, then (8) holds for infinitely many N.

The specific value of r is of no consequence, of course, but it witnesses the fact that r is independent of the specific relation R and it exhibits the form of its dependence on D, e and d.

Proof. Suppose $p > D + 1$ and $R(p)$. Take m such that

$$(D + 1)(2E)^m < p \le (D + 1)(2E)^{m+1}.$$

Put $\kappa := 1 + E^m F(E^m)$, so $\neg R(\kappa p)$. By Lemma 6.5 we have an embedding $\imath : \mathbf{A} \restriction B_m(p) \rightarrowtail \mathbf{A}$ such that $\imath(p) = \kappa p$. Also

$$\kappa p = (1 + E^m F(E^m))p \le (1 + dE^{m(e+1)})p \le dp^{e+2} = N,$$

where the last inequality uses $(2E)^m < p$. Hence $c_a^{\mathbf{A}}(R, N) \ge m + 1$ by Lemma 5.3, and it is enough to show that $m + 1 > r \log N$ with the r above.

Using the assumed hypotheses and setting $\delta = \log d$:

$$\begin{aligned} N = dp^{e+2} &\le d(D + 1)^{e+2}(2E)^{(e+2)(m+1)} \\ &\le d(2E)^{(e+2)(m+2)} \\ &< (2E)^{(e+2)(m+2)+\delta} \\ &\le (2E)^{(e+2)(m+1)+(e+2)+\delta} \\ &\le (2E)^{(m+1)(e+2+\frac{e+2+\delta}{m+1})} \\ &\le (2E)^{(m+1)(2e+4+\delta)}. \end{aligned}$$

The desired inequality follows from this by taking the logarithms of both sides. □

Remark

The proof shows that the logarithmic lower bound is forced by just *two* inputs: if $p > D + 1$ and $R(p)$, then there is a number κ such that $\kappa p \le N = dp^{e+2}$ and any $\mathbf{A}$-term $t(x)$ that decides R on $\{p, \kappa p\}$ has arithmetic depth $> r \log N$.

7. OPTIMALITY OF THE BINARY GCD-ALGORITHM

A careful look at the binary algorithm shows that for $\mathbf{A} = \mathbb{Z}(\mathrm{div}_2)$ and $a, b > 0$ we have $\gcd(a, b) \in G_m^{\mathbf{A}}(a, b)$ with $m \le 2(\log a + \log b)$. A little more work yields for each positive integer N an $\mathbf{A}$-term $t(x,y)$ of arithmetic depth $\leq 4\log N$ that computes gcd on $\{1,\dots,N\}^2$. (One obtains $t(x,y)$ by imitating what the binary algorithm does on inputs in $\{1,\dots,N\}^2$.) Hence there is also for each positive integer N an $\mathbf{A}$-term $t_\perp(x,y)$ of arithmetic depth $\leq 4\log N$ that decides the coprimeness relation $\perp$ on $\{1,\dots,N\}^2$.

Theorem 7.1 below shows that the binary algorithm is in a sense optimal for algorithms deciding $\perp$ using just linear operations, since the upper bound is matched by a proportional lower bound. (More precisely, Theorem 7.1 is the non-uniform version; the uniform version is Theorem 12.2.)

Theorem 7.1. *If $\mathbf{A} := \mathbb{Z}(\mathrm{div}_D)$ with $D > 1$, then there is a real constant $r > 0$ depending only on D such that for all $a > (D+1)^2$,*

$$c_a^{\mathbf{A}}(\perp, N) > r\log N \text{ with } N = a^3.$$

The lemmas and proofs involved are very similar to those in the previous section.

Lemma 7.2. *Suppose $|c_0|,|c_1|,|c_2| < a$ and $b > a^2$. Then*

(i) $c_0 + c_1\kappa a + c_2\kappa b = 0 \iff c_0 = c_1 = c_2 = 0$;

(ii) $c_0 + c_1\kappa a + c_2\kappa b > 0 \iff$ *either* $c_2 > 0$, *or* ($c_2 = 0$ *and* $c_1 > 0$), *or* ($c_2 = c_1 = 0$ *and* $c_0 > 0$).

Suppose $|c_0|,|c_1|,|d_0|,|d_1| < a/2$ and $b > a^2$. Then

(iii) $c_0 + c_1\kappa a + c_2\kappa b = d_0 + d_1\kappa a + d_2\kappa b \iff c_0 = d_0, c_1 = d_1, c_2 = d_2$;

(iv) $c_0 + c_1\kappa a + c_2\kappa b < d_0 + d_1\kappa a + d_2\kappa b \iff$ *either* $c_2 < d_2$, *or*
($c_2 = d_2$ *and* $c_1 < d_1$), *or*
($c_2 = d_2, c_1 = d_1, c_0 < d_0$).

Proof. Note that $|c_0 + c_1\kappa a| \leq a - 1 + \kappa(a^2 - a) < \kappa a^2 < \kappa b$, so Lemma 6.4 with b in the role of a yields (i) and (ii). Items (iii) and (iv) are obvious consequences of (i) and (ii). $\square$

The proof of the next lemma is like that of Lemma 6.5.

Lemma 7.3. *Suppose $D > 1$, $E = D!$, $a > (D+1)(2E)^m$, $U := B_m(a,b)$, $b > a^2$, and $\kappa \equiv 1 \mod E^m$. Then there exists an embedding $\imath : \mathbf{A}\restriction U \rightarrowtail \mathbf{A}$ given by*

$$\imath\Big(\frac{c_0 + c_1 a + c_2 b}{E^m}\Big) = \frac{c_0 + c_1\kappa a + c_2\kappa b}{E^m}, \quad |c_0|,|c_1|,|c_2| \leq (2E)^m.$$

Proof of Theorem of 7.1. Suppose $a > (D+1)^2 > (D+1)$, and choose m such that

$$(D+1)(2E)^m < a \leq (D+1)(2E)^{m+1}.$$

Put $b := a^2 + 1$ and $\kappa := 1 + E^m$, so $a \perp b$, but $\kappa a \not\perp \kappa b$. By Lemma 7.3 we have an embedding $\imath : \mathbf{A}\restriction B_m(a,b) \rightarrowtail \mathbf{A}$ such that $\imath(a,b) = (\kappa a, \kappa b)$. Also $\kappa b = (1 + E^m)(a^2 + 1) < a^3 = N$. Hence $c_a^{\mathbf{A}}(\perp, N) \geq m + 1$ by Lemma 5.3. From $(D+1)(2E)^{m+1} \geq a$ we obtain $m + 1 \geq \big(\log a - \log(D+1)\big)/\log 2E$. Using $\log N = 3\log a$ this yields

$$c_a^{\mathbf{A}}(\perp, N) \geq \frac{1}{3\log 2E}(\log N - 3\log(D+1)).$$

Now the hypothesis $a > (D+1)^2$ together with $\log N = 3\log a$ implies that $\log N > 6\log(D+1)$, so that $\log N - 3\log(D+1) > \frac{1}{2}\log N$, and when we insert this in the inequality just derived we obtain the desired

$$c_a^{\mathbf{A}}(\perp, N) \geq \frac{1}{6\log 2E}\log N. \quad \square$$

Notice that if only division by 2 is assumed, then $E = 2! = 2$ and so the constant in the lower bound is $\frac{1}{12}$—compared to 4 in the upper bound for the binary algorithm which only uses division by 2.

Remark

As in the previous section, the lower bound is forced by just two inputs: if $a > (D+1)^2$ then there is $\kappa > 1$ such that $\kappa(a^2+1) < N := a^3$ and any $\mathbf{A}$-term that decides $\perp$ on the two inputs (a, a^2+1) and $(\kappa a, \kappa(a^2+1))$ has arithmetic depth $> (\log N)/6\log 2E$.

8. LOWER BOUNDS FOR UNARY RELATIONS FROM $+, -, \div$

For each finite $\Phi \subseteq \mathbf{Lin}$ the partial algebra $(\mathbb{Z}, 0, 1, +, -, (\phi)_{\phi\in\Phi}, <)$ is term-reducible to $\mathbb{Z}(1) = (\mathbb{Z}, 0, 1, +, -, \mathrm{iq}, \mathrm{rem}, <)$. But iq and rem do not belong to $\mathbf{Lin}$, so $\mathbb{Z}(1)$ is not term-reducible to $(\mathbb{Z}, 0, 1, +, -, (\phi)_{\phi\in\Phi}, <)$ for any finite $\Phi \subseteq \mathbf{Lin}$. In contrast to the single logarithmic bounds of Sections 6 and 7, the lower bounds we will derive relative to $\mathbb{Z}(1)$ in this section are *double logarithmic*, and we will establish them for *term complexity* rather than the (smaller) arithmetic term complexity.

For unary relations we have an analogue of Theorem 6.1, but for term-complexity.

THEOREM 8.1. *Suppose $R \subseteq \mathbb{Z}$ is a unary relation, $F : \mathbb{N}^{>0} \to \mathbb{N}^{>0}$ is a function, and $d, e > 0$ are such that for all λ and p:*

(i) $R(p) \implies \neg R(\kappa p)$ *with* $\kappa = 1 + \lambda F(\lambda)$,

(ii) $F(\lambda) \leq d\lambda^e$.

Then for all sufficiently large p such that $R(p)$, if $N := p^{(e+1)\lfloor\frac{\log p}{5}\rfloor}$, then

$$c^{\mathbb{Z}(1)}(R, N) > \frac{1}{7}\log\log N. \tag{9}$$

In particular, if $R(p)$ is true for infinitely many p, then (9) *holds for infinitely many N.*

As with Theorem 6.1 this result applies to the unary relations of being prime, being a power of 2, being a perfect square, and being square-free. We do not know if a double logarithmic bound as in this theorem holds for the arithmetic term-complexity $c_a^{\mathbb{Z}(1)}$.

LEMMA 8.2. *Suppose $a > 0$, $|c_i|, |d_i| < \sqrt{\frac{a}{2}}$ for $i = 0, 1, 2$ and $c_2, d_2 \neq 0$. Then*

(i) $\frac{c_0+c_1a}{c_2} = \frac{d_0+d_1a}{d_2} \iff \frac{c_0}{c_2} = \frac{d_0}{d_2}$ *and* $\frac{c_1}{c_2} = \frac{d_1}{d_2}$;

(ii) $\frac{c_0+c_1a}{c_2} < \frac{d_0+d_1a}{d_2} \iff$ *either* $\frac{c_1}{c_2} < \frac{d_1}{d_2}$, *or* $(\frac{c_1}{c_2} = \frac{d_1}{d_2}$ *and* $\frac{c_0}{c_2} < \frac{d_0}{d_2})$.

This follows easily from Lemma 6.4 by clearing denominators.

Let h be an integer > 1 and put

$$A(a,h) := \mathbb{Z} \cap \Big\{ \frac{c_0 + c_1 a}{c_2} : c_2 \neq 0, |c_i| \leq h \text{ for } i = 0,1,2 \Big\}.$$

LEMMA 8.3. *Let $a > 2h^4$, and $c, d \in A(a,h)$. Then $c+d$, $c-d$, $\mathrm{iq}(c,d)$, and $\mathrm{rem}(c,d)$ lie in $A(a, 2h^4)$.*

PROOF. Write

$$c = \frac{c_0 + c_1 a}{c_2}, \qquad d = \frac{d_0 + d_1 a}{d_2}$$

with $c_2, d_2 \neq 0$ and $|c_i|, |d_i| \leq h$ for $i = 0,1,2$. Then

$$c + d = \frac{(c_0 d_2 + c_2 d_0) + (c_1 d_2 + c_2 d_1) a}{c_2 d_2},$$

hence $c + d \in A(a, 2h^2) \subseteq A(a, 2h^4)$, and similarly for $c - d$. To deal with $\mathrm{iq}(c,d)$ and $\mathrm{rem}(c,d)$ we first consider the case $d_1 \neq 0$. Solving for a in $d = (d_0 + d_1 a)/d_2$ and substituting in the expression for c, we get

$$c = \frac{c_1 d_2}{c_2 d_1} d + \frac{c_0 d_1 - c_1 d_0}{c_2 d_1}, \quad \text{with } \left| \frac{c_0 d_1 - c_1 d_0}{c_2 d_1} \right| \leq 2h^2.$$

The assumption $a > 2h^4$ easily yields $|d| > 2h^2$, so

$$\frac{c}{d} = \frac{c_1 d_2}{c_2 d_1} + \epsilon, \quad \text{with } |\epsilon| < 1.$$

Hence $|\mathrm{iq}(c,d)| \leq h^2 + 2$, in particular $\mathrm{iq}(c,d) \in A(a, 2h^4)$, and

$$\begin{aligned}\mathrm{rem}(c,d) = c - \mathrm{iq}(c,d) d &= \frac{c_0 + c_1 a}{c_2} - \frac{\mathrm{iq}(c,d) d_0 + \mathrm{iq}(c,d) d_1 a}{d_2} \\ &= \frac{(c_0 d_2 - \mathrm{iq}(c,d) c_2 d_0) + (c_1 d_2 - \mathrm{iq}(c,d) c_2 d_1) a}{c_2 d_2}.\end{aligned}$$

Hence $\mathrm{rem}(c,d) \in A(a, h^2 + (h^2+2)h^2) \subseteq A(a, 2h^4)$.

Next, suppose $d_1 = 0$. We can assume $d \neq 0$, so

$$|\mathrm{rem}(c,d)| < |d| \leq |d_0| \leq h,$$

in particular $\mathrm{rem}(c,d) \in A(a, 2h^4)$, and

$$\mathrm{iq}(c,d) = \frac{c - \mathrm{rem}(c,d)}{d} = \frac{c_0 d_0 - \mathrm{rem}(c,d) c_2 d_2 + c_1 d_0 a}{c_2 d_0},$$

which yields $\mathrm{iq}(c,d) \in A(a, 2h^3) \subseteq A(a, 2h^4)$. $\square$

Suppose that $a > 2h^2$ and $u \in A(a,h)$. Then by part (i) of Lemma 8.2 there is a unique triple (c_0, c_1, c_2) such that $|c_i| \leq h$ for $i = 0,1,2$, $c_2 > 0$, $\gcd(c_0, c_1, c_2) = 1$, and $u = \frac{c_0 + c_1 a}{c_2}$. Denote the component c_2 of this unique triple by $c_2(u)$. With these notations we have:

Lemma 8.4. *Suppose $a > 2h^4$, $0, 1 \in U \subseteq A(a, h)$, and $\kappa \equiv 1 \mod c_2(u)$ for each $u \in U$. Then we have an embedding $\imath : \mathbb{Z}(1) \restriction U \rightarrowtail \mathbb{Z}(1)$ such that*

$$\imath\Big(\frac{c_0 + c_1 a}{c_2}\Big) = \frac{c_0 + c_1 \kappa a}{c_2}$$

whenever $|c_i| \le h$ for $i = 0, 1, 2$, $c_2 \neq 0$ and $\frac{c_0 + c_1 a}{c_2} \in U$.

This follows easily from Lemma 8.2, and from Lemma 8.3 and its proof.

Lemma 8.5. *Suppose $a > 2^{5^m}$, and put $A_k(a) := A(a, 2^{5^k})$ for $0 \le k \le m$. We have the following inclusions for $k < m$:*

(i) *$A_k(a) \subseteq A_{k+1}(a)$;*

(ii) *$A_k(a) + A_k(a)$, $A_k(a) - A_k(a)$, $\mathrm{iq}(A_k(a), A_k(a))$ and $\mathrm{rem}(A_k(a), A_k(a))$ are subsets of $A_{k+1}(a)$;*

(iii) *$G_k^{\mathbb{Z}(1)}(a) \subseteq A_k(a)$ and $G_m^{\mathbb{Z}(1)}(a) \subseteq A_m(a)$.*

Proof. The last inclusion follows from (ii) by induction and from $G_0^{\mathbf{A}}(a) = \{0, 1, a\} \subseteq A(a, 2) = A_0(a)$. The inclusions (ii) follow from 8.3 using the fact that $2^{5^{k+1}} \ge 2\big(2^{5^k}\big)^4$. $\square$

If we were to proceed now as in the proof of Theorem 6.6, we would take $\lambda := 2^{5^m}!$ (for suitable m depending on p), put $\kappa := 1 + \lambda F(\lambda)$, and obtain an embedding of a certain algebra $\mathbb{Z}(1) \restriction U$ into $\mathbb{Z}(1)$ that sends p to κp. Unfortunately, κ as a function of m grows so fast that it would result in a lower bound on $c_a^{\mathbb{Z}(1)}(R, N)$ (for R as in Theorem 8.1) that is triple logarithmic in N. To achieve a double logarithmic lower bound we shall work with ordinary term-complexity instead of arithmetic term-complexity, and introduce the notion of *computation space.*

Computation space

This subsection is of a general nature, and $\mathbf{A}$ is any Φ-algebra. Given a Φ-term $t(\vec{x})$ and $\vec{a} \in A^m$ such that $t(\vec{a})\downarrow$, the *computation space* $S\big(t(\vec{x}), \vec{a}\big)$ is the finite subset of A defined inductively as follows:

(S1) if $t(\vec{x})$ is $0, 1, x_i$, respectively, then $S\big(t(\vec{x}), \vec{a}\big) = \{0\}, \{1\}, \{a_i\}$, respectively;

(S2) if $t(\vec{x})$ is $\mathrm{Eq}\big(t_1(\vec{x}), t_2(\vec{x})\big)$, then $S\big(t(\vec{x}), \vec{a}\big) = S\big(t_1(\vec{x}), \vec{a}\big) \cup S\big(t_2(\vec{x}), \vec{a}\big)$;
and if $t(\vec{x})$ is $\mathrm{Co}\big(t_1(\vec{x}), t_2(\vec{x}), t_3(\vec{x})\big)$, then

$$S\big(t(\vec{x}), \vec{a}\big) = \begin{cases} S\big(t_1(\vec{x}), \vec{a}\big) \cup S\big(t_2(\vec{x}), \vec{a}\big) & \text{if } t_1(\vec{a}) = 1, \\ S\big(t_1(\vec{x}), \vec{a}\big) \cup S\big(t_3(\vec{x}), \vec{a}\big) & \text{otherwise;} \end{cases}$$

(S3) if $t(\vec{x})$ is $\phi\big(t_1(\vec{x}), \dots, t_k(\vec{x})\big)$ where $\phi \in \Phi$ has arity k, then $S\big(t(\vec{x}), \vec{a}\big) = S\big(t_1(\vec{x}), \vec{a}\big) \cup \cdots \cup S\big(t_k(\vec{x}), \vec{a}\big) \cup \{t(\vec{a})\}$.

Intuitively, $S\big(t(\vec{x}), \vec{a}\big)$ is the part of A used in computing $t(\vec{a})$ via $t(\vec{x})$. The next lemma follows by an easy induction on $\mathrm{depth}(t)$.

Lemma 8.6. *Let $t(\vec{x})$ be a Φ-term and $\vec{a} \in A^m$ with $t(\vec{a}) \in A$. Suppose that $\mathrm{depth}(t) \le k \in \mathbb{N}$ and each symbol in Φ has arity $\le \ell$ where $1 \le \ell \in \mathbb{N}$. Then*

(i) *$t(\vec{a}) \in S\big(t(\vec{x}), \vec{a}\big) \subseteq G_k^{\mathbf{A}}(\vec{a})$;*

(ii) $|S\big(t(\vec{x}),\vec{a}\big)| \le (\ell+1)^k$.

The advantage of $S\big(t(\vec{x}),\vec{a}\big)$ over $G_k^{\mathbf{A}}(\vec{a})$ is that $|S\big(t(\vec{x}),\vec{a}\big)|$ grows at most exponentially in k (by part (2) of the above lemma), while $|G_k^{\mathbf{A}}(\vec{a})|$ can grow *doubly* exponentially in k.

The next absoluteness lemma also follows by an easy induction on t.

Lemma 8.7. *Let $t(\vec{x})$ be a Φ-term, $\vec{a} \in A^m$ and $t(\vec{a}) \in A$. Let $U \subseteq A$ be such that $0, 1, a_1, \dots, a_m \in U$ and $S\big(t(\vec{x}),\vec{a}\big) \subseteq U$. Then $t^{\mathbf{A}\restriction U}(\vec{a}) = t^{\mathbf{A}}(\vec{a})$.*

Proof of Theorem 8.1. Suppose $p \ge 2^{25(e+1)}$, $p > d$, and $R(p)$ holds. Take m maximal such that $p \ge 2^{5^{m+1}}$, so $m \ge 1$, and notice that with $M := 2^{5^m}$, we have

$$p \ge M^5 > 2M^4.$$

Suppose $t(x)$ is a $\mathbb{Z}(1)$-term of depth $\le m$ such that $t(p) = 1$.

Claim. There exists λ such that for $\kappa = 1 + \lambda F(\lambda)$ we have $t(\kappa p) = 1$ and $\kappa p \le N := p^{(e+1)\lfloor \frac{\log p}{5} \rfloor}$.

To prove this claim, set $U := S(t(x), p) \cup \{0, 1, p\}$, and note that by part (i) of Lemma 8.6 and part (iii) of Lemma 8.5 we have

$$U \subseteq G_m^{\mathbb{Z}(1)}(p) \subseteq A_m(p).$$

By part (ii) of Lemma 8.6 we have $|U \setminus \{0,1\}| \le K := 3^m + 1$. We set

$$\lambda := \prod_{u \in U\setminus\{0,1\}} c_2(u),$$

with $c_2(u)$ as defined just before Lemma 8.4, with p and M in the role of a and h. Note that $\lambda \equiv 0 \mod c_2(u)$ for all $u \in U$. Put $\kappa := 1+\lambda F(\lambda)$, so $\kappa \equiv 1 \mod c_2(u)$ for all $u \in U$. Hence by Lemma 8.4 we have an embedding $\imath : \mathbb{Z}(1)\restriction U \rightarrowtail \mathbb{Z}(1)$ such that $\imath(p) = \kappa p$, and thus $t(\kappa p) = 1$ by part (1) of Lemma 8.6, using also Lemma 8.7. To finish the proof of the claim it remains to show that $\kappa p \le N$. Using the assumption $p > d$, we have

$$\kappa \le 1 + d\lambda^{e+1} \le 1 + dM^{(e+1)K} \le pM^{(e+1)K},$$

hence $\kappa p \le p^2 M^{(e+1)K} \le p^{(e+1)K}$, where the last inequality uses the assumption $M \le p^{\frac{1}{5}}$ and $K \ge 3$. Finally

$$K = 3^m + 1 < 5^m = \frac{5^{m+1}}{5} \le \frac{\log p}{5},$$

and hence $K \le \lfloor \frac{\log p}{5} \rfloor$, so that $p^{(e+1)K} \le p^{(e+1)\lfloor \frac{\log p}{5} \rfloor} = N$.

Because $t(x)$ was an arbitrary $\mathbb{Z}(1)$-term of depth $\le m$ with $t(p) = 1$, it follows from the claim that

$$c^{\mathbb{Z}(1)}(R, N) \ge m + 1,$$

and so to complete the proof, it suffices to show that $\log\log N < 7(m+1)$. Using the definition of N, we compute:

$$\begin{aligned} \log N &= (e+1)\lfloor \frac{\log p}{5} \rfloor \log p \\ &\leq (e+1)\frac{(\log p)^2}{5} \\ &\leq \frac{(\log p)^3}{5^3} \qquad \text{(by the hypothesis } 2^{5^2(e+1)} \leq p), \end{aligned}$$

and so $\log\log N \leq 3\log\log p - 3\log 5$. On the other hand, by the choice of m we have $p < 2^{5^{m+2}}$, which yields $\log\log p < (\log 5)(m+2)$; thus

$$\log\log N < (3\log 5)(m+2) - 3\log 5 = (3\log 5)(m+1)$$

which yields the required $7(m+1) > (3\log 5)(m+1) > \log\log N$. $\square$

9. A LOWER BOUND FOR COPRIMENESS FROM $+, -, \div$

Here we derive a $\log\log$ lower bound for deciding coprimeness in

$$\mathbb{Z}(1) = (\mathbb{Z}, 0, 1, +, -, \mathrm{iq}, \mathrm{rem}, <),$$

the analog for $\mathbb{Z}(1)$ of Theorem 7.1.

Theorem 9.1. *For infinitely many positive integers N,*

$$c^{\mathbb{Z}(1)}(\perp, N) \geq \frac{1}{7}\log\log N. \tag{10}$$

In fact, for all coprime positive a, b such that $|\frac{a}{b} - \sqrt{2}| < \frac{1}{b}$ and $a \geq 2^{128}$, if m is largest with $a \geq 2^{2^{3m+4}}$, then (10) holds with $N := 2^{2^{7(m+1)}}$.

The proof is very much like that of Theorem 7.1, with almost all the required estimates already derived in Sections 3 and 7. In the next two lemmas a and b are positive with $|\frac{a}{b} - \sqrt{2}| < \frac{1}{b}$. (These lemmas do not assume coprimeness of a and b.) Note that then $G_m(a,b) = G_m^{\mathbb{Z}(1)}(a,b)$, with $G_m(a,b)$ as defined in the beginning of Part 1. We also use the sets $A(h)$ defined just before lemma 3.5 in terms of a, b and an integer $h \geq 2$.

Lemma 9.2. *Let $m \geq 1$, $h := 2^{2^{3m}}$, $a \geq h^{16}$, $b > 0$, $|\frac{a}{b} - \sqrt{2}| < \frac{1}{b}$. Then $G_m(a,b) \subseteq A(h)$, and for each $u \in A(h)$ there is a unique tuple (c_0, c_1, c_2, c_3) such that*

$$\begin{aligned} &|c_i| \leq h \text{ for } i = 0,1,2,3, && c_3 > 0, \\ &u = \frac{c_0 + c_1 a + c_2 b}{c_3}, && \gcd(c_0, c_1, c_2, c_3) = 1. \end{aligned}$$

Proof. From $h \geq 2^8$, $a \geq h^{16}$ and $|\frac{a}{b} - \sqrt{2}| < \frac{1}{b}$ we obtain the following lower bounds on b, to be used in this proof and in the proof of the next lemma:

$$b > \frac{1}{2}a \geq \frac{1}{2}h^{16} > 400h^4 > 10h.$$

Then Lemma 3.5 yields $G_m(a,b) \subseteq A(h)$. Note that each $u \in A(h)$ has the required form. The uniqueness of the representation follows from Lemma 3.3, which for its application to the equality of two numbers of the required form assumes that $2h^2 \geq 2$ and $10(2h^2)^2 = 40h^4 \leq b$; these bounds follow from the above. $\square$

Let $c_3(u)$ denote the component c_3 of the unique tuple (c_0, c_1, c_2, c_3) that the lemma above associates to an element $u \in A(h)$.

LEMMA 9.3. *With the same assumptions as the previous lemma, let $0, 1 \in U \subseteq A(h)$ and $\kappa \equiv 1 \mod c_3(u)|v|$ for all $u, v \in U$ with $0 < |v| \leq h$. Then we have an embedding $\imath : \mathbb{Z}(1) \restriction U \rightarrowtail \mathbb{Z}(1)$ such that*

$$\imath\Big(\frac{c_0 + c_1 a + c_2 b}{c_3}\Big) = \frac{c_0 + c_1 \kappa a + c_2 \kappa b}{c_3}$$

whenever $|c_i| \leq h$ for $i = 0, 1, 2, 3$, $c_3 \neq 0$ and $\frac{c_0 + c_1 a + c_2 b}{c_3} \in U$.

PROOF. The previous lemma yields a function $\imath : U \to \mathbb{Z}$ satisfying the indicated identity. We have to show that $\imath$ is an embedding $\mathbb{Z}(1) \restriction U \rightarrowtail \mathbb{Z}(1)$. Consider elements $f, g, u \in U$,

$$f = \frac{c_0 + c_1 a + c_2 b}{c_3}, \quad g = \frac{d_0 + d_1 a + d_2 b}{d_3}, \quad u = \frac{e_0 + e_1 a + e_2 b}{e_3},$$

with $|c_i|, |d_i|, |e_i| \leq h$ for $i = 0, 1, 2, 3$, and $c_3, d_3, e_3 > 0$. Among the things to check is that if $f < g$, then $\imath(f) < \imath(g)$. We have

$$f - g = \frac{C_0 + C_1 a + C_2 b}{C_3}, \quad \text{where } C_0 := c_0 d_3 - c_3 d_0,$$
$$C_1 := c_1 d_3 - c_3 d_1, \quad C_2 := c_2 d_3 - c_3 d_2, \quad C_3 := c_3 d_3, \text{ and}$$
$$\imath(f) - \imath(g) = \frac{C_0 + C_1 \kappa a + C_2 \kappa b}{C_3}.$$

Now $|C_i| \leq 2h^2$ for $i = 0, 1, 2$ and $b \geq 10(2h^2)^2$, so by Lemma 3.3, if $f < g$ and C_1 and C_2 are not both zero, then $C_1\sqrt{2} + C_2 < 0$, and thus $\imath(f) < \imath(g)$, again by Lemma 3.3. If $f < g$ and C_1 and C_2 are both zero, then $f - g = \imath(f) - \imath(g)$, so $\imath(f) < \imath(g)$ as well.

Next assume that $f - g = u$; we have to check that then $\imath(f) - \imath(g) = \imath(u)$. The equality $f - g = u$ yields

$$(C_0 e_3 - C_3 e_0) + (C_1 e_3 - C_3 e_1)a + (C_2 e_3 - C_3 e_2)b = 0.$$

Since $|C_i e_3 - C_3 e_i| \leq 4h^3$ for $i = 0, 1, 2$, and $b \geq 10(4h^3)^2$, Lemma 3.3 yields $C_i e_3 - C_3 e_i = 0$ for $i = 0, 1, 2$, which gives $\imath(f) - \imath(g) = \imath(u)$. Likewise it follows that if $f + g = u$, then $\imath(f) + \imath(g) = \imath(u)$.

It remains to show that if $\mathrm{iq}(f, g) = u$, then $\mathrm{iq}(\imath(f), \imath(g)) = \imath(u)$, and that if $\mathrm{rem}(f, g) = u$, then $\mathrm{rem}(\imath(f), \imath(g)) = \imath(u)$. If $g = 0$, then these implications are trivially valid. In the rest of the proof we assume $g > 0$; the case $g < 0$ can be treated in the same way. First assume that also $d_1 = d_2 = 0$. Then $0 \leq \mathrm{rem}(f, g) < g \leq h$.

Suppose that $\mathrm{rem}(f, g) = u$; then $f \equiv u \mod g$. Without loss we have $c_3(f) = c_3$, so $\kappa \equiv 1 \mod c_3 g$ and thus $\imath(f) \equiv u \mod g$. In view of $\imath(g) = g$ and $\imath(u) = u$, this yields the desired $\mathrm{rem}(\imath(f), \imath(g)) = \imath(u)$.

Next, suppose that $\mathrm{iq}(f,g) = u$. Put $r := \mathrm{rem}(f,g)$. From the identity $f - gu - r = 0$ we obtain by substitution for f and u:

$$(c_0e_3 - c_3e_0g - rc_3e_3) + (c_1e_3 - c_3e_1g)a + (c_2e_3 - c_3e_2g)b = 0.$$

Since $b \geq 10(3h^3)^2$, this yields by Lemma 3.3:

$$c_0e_3 - c_3e_0g - rc_3e_3 = c_1e_3 - c_3e_1g = c_2e_3 - c_3e_2g = 0,$$

which in turn yields $\imath(f) - g\imath(u) - r = 0$; in view of $\imath(g) = g$ and $0 \leq r < g$, this gives $\mathrm{iq}(\imath(f), \imath(g)) = \imath(u)$, as required.

For the rest of the proof we assume that d_1 and d_2 are not both zero. From the proof of Lemma 3.5 we recall that

$$0 \leq \mathrm{rem}(f,g) = f - \mathrm{iq}(f,g)g = \frac{X + Ya + Zb}{C_3} < g, \quad \text{with}$$
$$X = c_0d_3 - \mathrm{iq}(f,g)c_3d_0, \quad Y = c_1d_3 - \mathrm{iq}(f,g)c_3d_1, \quad Z = c_2d_3 - \mathrm{iq}(f,g)c_3d_2,$$

Likewise, $\imath(f) - \mathrm{iq}(f,g)\imath(g) = \frac{X+Y\kappa a+Z\kappa b}{C_3}$. Note also that by the proof of Lemma 3.5 we have $|\mathrm{iq}(f,g)| \leq 7h^3$.

Suppose $\mathrm{iq}(f,g) = u$; we have to show $\mathrm{iq}(\imath(f), \imath(g)) = \imath(u)$. The proof of Lemma 3.6 with u instead of f shows that if $e_1 \neq 0$ or $e_2 \neq 0$, then $|u| > 2\sqrt{b} > 7h^3$, which contradicts $|u| = |\mathrm{iq}(f,g)| \leq 7h^3$. So $e_1 = e_2 = 0$, hence $|u| \leq h$, and thus $\imath(u) = u$ and $|X|, |Y|, |Z| \leq h^2 + h^3$. Also

$$\mathrm{rem}(f,g) = \frac{X + Ya + Zb}{C_3} < g = \frac{d_0 + d_1a + d_2b}{d_3},$$

so

$$(X - C_3d_0) + (Y - C_3d_1)a + (Z - C_3d_2)b < 0.$$

Since $b \geq h^7 \geq 10(h^2 + h^3 + h^3)^2$, Lemma 3.3 yields

$$(Y - C_3d_1)\sqrt{2} + (Z - C_3d_2) < 0$$

if $Y - C_3d_1$ and $Z - C_3d_2$ are not both zero. Again by Lemma 3.3, and trivially so when $Y - C_3d_1 = Z - C_3d_2 = 0$, this yields

$$(X - C_3d_0) + (Y - C_3d_1)\kappa a + (Z - C_3d_2)\kappa b < 0, \quad \text{so}$$
$$\imath(f) - u \cdot \imath(g) = \frac{X + Y\kappa a + Z\kappa b}{C_3} < \imath(g).$$

From $\frac{X+Ya+Zb}{C_3} \geq 0$ we obtain likewise that $\frac{X+Y\kappa a+Z\kappa b}{C_3} \geq 0$. Hence

$$\mathrm{rem}(\imath(f), \imath(g)) = \imath(f) - u \cdot \imath(g) = \frac{X + Y\kappa a + Z\kappa b}{C_3},$$

in particular, $\mathrm{iq}(\imath(f), \imath(g)) = u = \imath(u)$.

Finally, let $\mathrm{rem}(f,g) = u$; to derive $\mathrm{rem}(\imath(f), \imath(g)) = \imath(u)$, note that

$$\mathrm{rem}(f,g) = \frac{X + Ya + Zb}{C_3} = \frac{e_0 + e_1a + e_2b}{e_3} = u,$$

from which we obtain

$$(e_0C_3 - e_3X) + (e_1C_3 - e_3Y)a + (e_2C_3 - e_3Z)b = 0.$$

Since $|X|, |Y|, |Z| \leq h^2 + 7h^3h^2 \leq 8h^5$, the coefficients of $1, a, b$ in the above expression are all of absolute value $\leq h^7$. Since $b \geq 10(h^7)^2$, Lemma 3.3 yields

$$e_0C_3 - e_3X = e_1C_3 - e_3Y = e_2C_3 - e_3Z = 0,$$

so

$$\frac{X + Y\kappa a + Z\kappa b}{C_3} = \frac{e_0 + e_1\kappa a + e_2\kappa b}{e_3},$$

that is, $\imath(f) - \mathrm{iq}(f, g)\imath(g) = \imath(u)$, and thus $\mathrm{rem}(\imath(f), \imath(g)) = \imath(u)$. $\square$

Proof of Theorem 9.1. Let a, b be positive and coprime, such that

$$|\frac{a}{b} - \sqrt{2}| < \frac{1}{b}, \quad a \geq 2^{128}.$$

Take m be maximal with $a \geq 2^{2^{3m+4}}$, so $m \geq 1$, and put $h := 2^{2^{3m}}$ and $N := 2^{2^{7(m+1)}}$. Then $h^{16} \leq a \leq N$ and $\frac{1}{7}\log\log N = m + 1$; assume towards a contradiction that $t(x, y)$ is a $\mathbb{Z}(1)$-term of depth $\leq m$ deciding coprimeness for all pairs c, d with $1 \leq c, d \leq N$, in particular, $t(a, b) = 1$. Let $U := S(t(x, y), a, b) \cup \{0, 1, a, b\}$ (the computation space for t at input (a, b) augmented with $0, 1, a, b$). Then $U \subseteq G_m(a, b) \subseteq A(h)$ by Lemmas 8.6 and 9.2. Put

$$\kappa \;=\; 1 \;+\; \prod_{u \in U} c_3(u) \;\cdot\; \prod_{u \in U,\ 0 < |u| \leq h} |u|.$$

Then $\kappa \equiv 1 \mod c_3(u)|v|$ for all $u, v \in U$ with $0 < |v| \leq h$. Let

$$\imath : \ \mathbb{Z}(1) \restriction U \rightarrowtail \mathbb{Z}(1)$$

be the embedding of Lemma 9.3. Since $\iota(a) = \kappa a$ and $\iota(b) = \kappa b$ and $t(a, b) = 1$ we have $t(\kappa a, \kappa b) = 1$. Thus to reach a contradiction, it is enough to show that $\kappa a \leq N$, since κa and κb are not coprime and hence cannot be within the interval $[1, N]$ in which $t(x, y)$ is assumed to decide coprimeness. Recall that $|U| \leq 3^m + 4$ by Lemma 8.6, and note that if $u \in U, 0 < |u| \leq h$, then $c_3(u) = 1$. Hence

$$\kappa \leq 1 + \left(2^{2^{3m}}\right)^{3^m+4} = 1 + 2^{2^{3m}(3^m+4)} = 1 + 2^{(24)^m + 4.8^m} < 2^{(64)^m} = 2^{2^{6m}},$$

and then, by the selection of m,

$$\kappa a < 2^{2^{6m}} 2^{2^{3(m+1)+4}} = 2^{2^{6m}(1+2^{7-3m})} \leq 2^{2^{6m}(1+2^4)} < 2^{2^{7(m+1)}} = N,$$

which is the desired contradiction. $\square$

10. LOWER BOUNDS FOR UNARY RELATIONS FROM $+, -, \div, \times$

We now include multiplication among the primitives by considering

$$\mathbb{Z}(2) = \big(\mathbb{Z}, 0, 1, +, -, \cdot, \mathrm{iq}, \mathrm{rem}, <\big).$$

We have the following analogue of Theorem 8.1. Instead of a $\log\log$ bound we get a $\sqrt{\log\log}$ bound.

Theorem 10.1. *Let $R \subseteq \mathbb{Z}$ be a unary relation, $F : \mathbb{N}^{>0} \to \mathbb{N}^{>0}$ a function, and $d, e > 0$ such that for all λ and p:*

(i) $R(p) \implies \neg R(\kappa p)$ *with* $\kappa = 1 + \lambda F(\lambda)$,
(ii) $F(\lambda) \leq d\lambda^e$.

Then for all sufficiently large p, if $R(p)$ and $N := p^{e+3}$, then

$$c^{\mathbb{Z}(2)}(R, N) > \frac{1}{2}\sqrt{\log\ \log N}. \tag{11}$$

In particular, if $R(p)$ is true for infinitely many p, then (11) *holds for infinitely many N.*

As with Theorem 8.1 this result applies to the unary relations of being prime, being a power of 2, being a perfect square, and being square-free.

Let $h \in \mathbb{N}$, $h > 0$, let X be an indeterminate, and put

$$\begin{aligned}\mathbb{Q}[X; n, h] : &= \{\frac{F}{c} : F \in \mathbb{Z}[X],\ \deg F \leq n,\ \|F\|_\infty \leq h,\ 0 < c \leq h\} \\ &= \{\frac{c_0 + c_1 X + \cdots + c_n X^n}{c} : |c_0|, \ldots, |c_n| \leq h,\ 0 < c \leq h\}.\end{aligned}$$

We make $\mathbb{Q}[X]$ into an ordered ring by setting, for non-zero $f(X) \in \mathbb{Q}[X]$: $f > 0$ iff the leading coefficient of f is > 0.

Lemma 10.2. *If $a > h$ and $f \in \mathbb{Q}[X; n, h]$, $f > 0$, then $f(a) > 0$. If $a > 2h^2$, then the evaluation map*

$$f \mapsto f(a) :\ \mathbb{Q}[X; n, h] \to \mathbb{Q}$$

is injective, and preserves order: $f, g \in \mathbb{Q}[X; n, h]$, $f < g \Longrightarrow f(a) < g(a)$.

The proof is a routine exercise left to the reader.

The set of integer values of polynomials in $\mathbb{Q}[X; n, h]$ at a is given by

$$\mathbb{Z}[a; n, h] := \mathbb{Z} \cap \{f(a) : f(T) \in \mathbb{Q}[X; n, h]\}.$$

Notice that

(i) $\mathbb{Z}[a; n, h] + \mathbb{Z}[a; n, h] \subseteq \mathbb{Z}[a; n, 2h^2]$,
(ii) $\mathbb{Z}[a; n, h] \cdot \mathbb{Z}[a; n, h] \subseteq \mathbb{Z}[a; 2n, (n + 1)h^2]$.

Lemma 10.3. *Suppose $a > 2^{n+1}h^{n+2}$, and $s, t \in \mathbb{Z}[a; n, h]$. Then*

$$\mathrm{iq}(s, t), \mathrm{rem}(s, t) \in \mathbb{Z}[a; n, 2^{n+1}h^{n+2}].$$

Proof. By symmetry we can assume $t > 0$. Write $s = f(a)$ and $t = g(a)$ with $f, g \in \mathbb{Z}[X; n, h]$ and $g \neq 0$. If $\deg f < \deg g$, then $|s| < t$, so either $\mathrm{iq}(s, t) = 0$ and $\mathrm{rem}(s, t) = s$, or $\mathrm{iq}(s, t) = -1$ and $\mathrm{rem}(s, t) = s + t$, and in either case the desired result clearly holds. If $\deg g = 0$, then $t \leq h$, so $|\mathrm{rem}(s, t)| \leq h$, and hence

$$\mathrm{iq}(s, t) = \frac{f(a) - \mathrm{rem}(s, t)}{t} \in \mathbb{Z}[a; n, h^2 + h] \subseteq \mathbb{Z}[a; n, 2^{n+1}h^{2n+3}].$$

as desired. Therefore we can assume $n \geq \deg f \geq \deg g > 0$ in the rest of the proof. Take $0 < c, d \leq h$ such that $F := cf$ and $G := dg$ belong to $\mathbb{Z}[X]$ and

$\|F\|_\infty, \|G\|_\infty \le h$. Let $e > 0$ be the leading coefficient of G. Then pseudo-division of F by G (cf. [von zur Gathen and Gerhard 1999]) yields, with $k = \deg f - \deg g < n$:

$$e^{k+1}F = QG + R, \quad Q, R \in \mathbb{Z}[X], \quad \deg R < \deg G = \deg g$$

with $\|Q\|_\infty \le 2^{n-1}h^n$ and $\|R\|_\infty \le 2^n h^{n+1}$, for example by Exercise 6.44 in [von zur Gathen and Gerhard 1999]. Put $b := e^{k+1}c$, so $0 < |b| \le h^{n+1}$. Division by b and substitution of a for X in the equality above yields

$$s = qt + r, \text{ with } q = \frac{dQ(a)}{b} \text{ and } r = \frac{R(a)}{b}.$$

From $\deg R < \deg G$, the inequality assumed about a, and the first part of Lemma 10.2 it follows easily that $|r| < t$. Writing $dQ(a) = q'b + r'$ where $q', r' \in \mathbb{Z}$ and $0 \le |r'| < |b|$, we have

$$s = \frac{q'b + r'}{b}t + r = q't + (\frac{r'}{b}t + r), \qquad |\frac{r'}{b}t + r| < 2t.$$

We now distinguish four cases.

Case 1. $0 \le \frac{r'}{b}t + r < t$. Then $\mathrm{iq}(s,t) = q'$ and $\mathrm{rem}(s,t) = \frac{r'}{b}t + r$. We have $q' = \frac{dQ(a)-r'}{b}$, hence by the bounds on $\|Q\|_\infty$, $\|R\|_\infty$:

$$\begin{aligned}\mathrm{iq}(s,t) &\in \mathbb{Z}[a; n, 2^{n-1}h^{n+1} + h^{n+1}] \subseteq \mathbb{Z}[a; n, 2^{n+1}h^{n+2}],\\ \mathrm{rem}(s,t) &= \frac{r'}{b}t + r = \frac{r'G(a)}{bd} + \frac{R(a)}{b} = \frac{r'G(a) + dR(a)}{bd}\\ &\in \mathbb{Z}[a; n, h^{n+2} + 2^n h^{n+2}] \subseteq \mathbb{Z}[a; n, 2^{n+1}h^{n+2}].\end{aligned}$$

Case 2. $-t \le \frac{r'}{b}t + r < 0$. Then $s = (q'-1)t + \frac{r'+b}{b}t + r$ with $0 \le \frac{r'+b}{b}t + r < t$, so $\mathrm{iq}(s,t) = q' - 1$ and $\mathrm{rem}(s,t) = \frac{r'+b}{b}t + r$. Now $q' - 1 = \frac{dQ(a)-r'-b}{b}$, so

$$\begin{aligned}\mathrm{iq}(s,t) &\in \mathbb{Z}[a; n, 2^{n-1}h^{n+1} + 2h^{n+1}] \subseteq \mathbb{Z}[a; n, 2^{n+1}h^{n+2}],\\ \mathrm{rem}(s,t) &= \frac{r'+b}{b}t + r = \frac{(r'+b)G(a) + dR(a)}{bd}\\ &\in \mathbb{Z}[a; n, 2h^{n+2} + 2^n h^{n+2}] \subseteq \mathbb{Z}[a; n, 2^{n+1}h^{n+2}].\end{aligned}$$

Case 3. $t \le \frac{r'}{b}t + r < 2t$. Then $\frac{r'}{b} > 0$; after replacing r' by $r' - b$ (and q' by $q' + 1$) we are back in case 1.

Case 4. $-2t < \frac{r'}{b}t + r < -t$. Then $\frac{r'}{b} < 0$; after replacing r' by $r' + b$ (and q' by $q' - 1$) we are back in case 2. $\square$

Suppose that $a > 2h^2$ and $u \in \mathbb{Z}[a; n, h]$. Then by Lemma 10.2 there is a unique tuple $(c, c_0, \dots, c_n)$ such that $0 < c \le h$, $|c_i| \le h$ for $i = 0, \dots, n$, $\gcd(c, c_0, \dots, c_n) = 1$, and $u = \frac{c_0 + c_1 a + \cdots + c_n a^n}{c}$. Denote the component c of this unique tuple by $c(u)$. With these notations we have

Lemma 10.4. *Let $a > 2^{n+1}h^{n+2}$, let $0, 1 \in U \subseteq \mathbb{Z}[a; n, h]$, and suppose $\kappa \equiv 1$ mod $c(u)$ for each $u \in U$. Then we have an embedding $\imath : \mathbb{Z}(2)\restriction U \rightarrowtail \mathbb{Z}(2)$ such that*

$$\imath\Big(\frac{c_0 + c_1 a + \cdots + c_n a^n}{c}\Big) = \frac{c_0 + c_1\kappa a + \cdots + c_n\kappa^n a^n}{c}$$

whenever $0 < |c| \leq h$, $|c_i| \leq h$ *for* $i = 0, 1, \ldots, n$ *and* $\frac{c_0 + c_1 a + \cdots + c_n a^n}{c} \in U$.

This follows by closely inspecting the proofs of Lemma 10.3 and of items (i) and (ii) listed just ahead of that lemma.

LEMMA 10.5. *Suppose* $a > 2^{2^{3m^2}}$, *and put*

$$C_k(a) := \mathbb{Z}[a; 2^k, 2^{2^{3k^2}}], \quad 0 \leq k \leq m.$$

We have the following inclusions for $0 \leq k < m$*:*

(i) $C_k(a) \subseteq C_{k+1}(a)$*;*

(ii) $C_k(a) + C_k(a)$, $C_k(a) - C_k(a)$, $\mathrm{iq}(C_k(a), C_k(a))$, $\mathrm{rem}(C_k(a), C_k(a))$ *and* $C_k(a) \cdot C_k(a)$ *are subsets of* $C_{k+1}(a)$*;*

(iii) $G_k^{\mathbb{Z}(2)}(a) \subseteq C_k(a)$ *and* $G_m^{\mathbb{Z}(2)}(a) \subseteq C_m(a)$.

PROOF. Inclusions (iii) follow from (ii) by induction and $G_0^{\mathbb{Z}(2)}(a) = \{0, 1, a\} \subseteq \mathbb{Z}[a; 1, 2] = C_0(a)$. Inclusions (ii) follow from Lemma 10.3. □

PROOF OF THEOREM 10.1. Let $p \geq 2^{2^{12}}$, $p > d$, such that $R(p)$ holds. Take m maximal such that $p \geq 2^{2^{3(m+1)^2}}$, so $m \geq 1$. Put $M := 2^{2^{3m^2}}$. Suppose $t(x)$ is a $\mathbb{Z}(2)$-term of depth $\leq m$ such that $t(p) = 1$.

Claim. There exists λ such that for $\kappa = 1 + \lambda F(\lambda)$ we have $t(\kappa p) = 1$ and $\kappa p \leq N := p^{e+3}$.

To prove this claim we set $U := S(t(x), p) \cup \{0, 1, p\}$, and we note that by part (i) of Lemma 8.6 and part (iii) of Lemma 10.5 we have

$$U \subseteq G_m^{\mathbb{Z}(2)}(p) \subseteq C_m(p).$$

By part (ii) of Lemma 8.6 we have $|U| \leq K := 3^m + 3$. Put

$$\lambda := \prod_{u \in U} c(u)$$

where we use the notations of Lemma 10.4 with p, 2^m and M in the role of a, n and h. Then $\lambda \equiv 0 \mod c(u)$ for all $u \in U$. Put $\kappa := 1 + \lambda F(\lambda)$, so $\kappa \equiv 1 \mod c(u)$ for all $u \in U$. Hence by Lemma 10.4 we have an embedding $\imath : \mathbb{Z}(2) \restriction U \rightarrowtail \mathbb{Z}(2)$ such that $\imath(p) = \kappa p$, and thus $t(\kappa p) = 1$ by part (i) of Lemma 8.6, using also Lemma 8.7. To finish the proof of the claim it remains to show that $\kappa p \leq N$. We have $\kappa \leq 1 + d\lambda^{e+1} \leq 1 + dM^{(e+1)K} \leq pM^{(e+1)K} \leq p^{e+2}$, since

$$p^{e+1} = 2^{(e+1)2^{3(m+1)^2}} \geq M^{(e+1)K} = 2^{(e+1)(3^m+3)2^{3m^2}}.$$

Hence $\kappa p \leq p^{e+3} = N$, which establishes the claim.

Because $t(x)$ was an arbitrary $\mathbb{Z}(2)$-term of depth $\leq m$ with $t(p) = 1$, it follows from the claim that $c^{\mathbb{Z}(2)}(R, N) \geq m + 1$. In view of

$$\log\log N = \log(e+3) + 3(m+1)^2$$

it follows that $c^{\mathbb{Z}(2)}(R, N) > \frac{1}{2}\sqrt{\log\log N}$ provided $p > P_0$ where the constant $P_0 > 0$ depends only on e.

The arguments above show that for $P = \max\{d, 2^{2^{12}}, P_0\}$ we have

$$p > P \text{ and } R(p) \quad \Longrightarrow \quad c^{\mathbb{Z}(2)}(R, N) > \frac{1}{2}\sqrt{\log\log N}. \quad \square$$

Decision-tree complexity

With essentially the same primitives, [Mansour et al. 1991b] has a $\sqrt{\log\log}$ lower bound for "being a perfect square", and [Meidânis 1991] has such a bound for primality. As usual in computer science these bounds are expressed as functions of the length of the binary representation of the input, and therefore show up as $\sqrt{\log}$ bounds. A more substantial difference is that the lower bounds in [Mansour et al. 1991a], [Mansour et al. 1991b] and [Meidânis 1991] are for a *sequential* model of computation, namely *decision-trees*, in contrast to our term-complexity.

PART 3. LOWER BOUNDS FOR RECURSIVE PROGRAMS

There is no single term $t(x, y)$ which computes $\gcd(x, y)$ for all $x, y \in \mathbb{N}^{>0}$, in any of the algebras we have considered, and similarly for coprimeness and the other examples we have studied. Thus the lower bounds of Part 2 apply to (non-uniform) term-algorithms, which compute the relevant function for inputs $\leq N$ using some term $t_N(\vec{x})$. In this Part we will derive lower bounds for uniform algorithms which apply to all inputs and are expressed by *recursive programs.* These uniform results have an especially simple, "pointwise" form which identifies directly the inputs that account for the lower bounds. Moreover, their precise formulation sheds some light on the expected "lower efficiency" of uniform (relative to non-uniform) algorithms, and we will discuss this phenomenon briefly after the statements of Theorems 12.1 and 12.2.

Weaker versions of some of the results in this part have already appeared in [van den Dries and Moschovakis 2004], whose last Section 6 also explains how to extend them to classes of algorithms much wider than those expressed by recursive programs.

11. RECURSIVE (MCCARTHY) PROGRAMS

We follow here the notation and conventions of Section 5, and we let k and K range over $\mathbb{N}$.

Program syntax

An n-ary *recursive* or *McCarthy* Φ*-program* is a $(K + 1)$-tuple

$$\alpha = (\alpha_0(\vec{v}, \vec{f}), \alpha_1(\vec{v}, \vec{f}), \ldots, \alpha_K(\vec{v}, \vec{f}))$$

of terms (with equality tests and conditionals), in the expanded signature

$$(\Phi, \vec{f}) = \Phi \cup \{f_1, \ldots, f_K\},$$

where, as usual, $\vec{v} = (v_1, \ldots, v_n)$, is a sequence of (distinct) variables which includes all the variables that occur free in the terms α_i. Here $f_1, \ldots, f_K$ are fresh, n-ary function symbols, the *recursion variables* of α. The *body* of α is the K-tuple of terms $\alpha_i(\vec{v}, \vec{f})$, which encodes the system of identities

$$\begin{aligned} f_1(\vec{v}) &= \alpha_1(\vec{v}, \vec{f}) \\ &\vdots \\ f_K(\vec{v}) &= \alpha_K(\vec{v}, \vec{f}) \end{aligned} \tag{12}$$

The first term $\alpha_0(\vec{v}, \vec{f})$ is the *head* of α.

We allow $K = 0$, in which case $\vec{f}$ is the empty sequence and α is identified with its head—so that each Φ-term can be viewed as a program.

Fixed point semantics

Terms in the signature $(\Phi, \vec{f})$ of a recursive program α are naturally interpreted in expansions

$$(\mathbf{A}, p_1, \dots, p_K) = (A, 0, 1, \{\phi^{\mathbf{A}}\}_{\phi \in \Phi}, p_1, \dots, p_K)$$

of a (partial) Φ-algebra $\mathbf{A}$. To interpret α on $\mathbf{A}$, we view the recursion variables $f_1, \dots, f_K$ as the *unknowns* in the system of identities determined by the body of α. It is well known that each such system has a $\sqsubseteq$-*least* tuple of partial functions

$$\overline{f}_1, \dots, \overline{f}_K : A^n \rightharpoonup A$$

which satisfies it, essentially because the operations on partial functions defined by terms are *monotone* and *continuous*; in fact

$$\overline{f}_i = \bigcup\nolimits_s \overline{f}_i^{(s)}, \text{ where } \overline{f}_i^{(0)} \sqsubseteq \overline{f}_i^{(1)} \sqsubseteq \cdots \quad (i = 1, \dots, K),$$

so that the *iterates* $\overline{f}_i^{(s)}$ provide approximations to the tuple $(\overline{f}_1, \dots, \overline{f}_K)$ of *mutual fixed points* of the body of α, e.g., see Section 1 of [van den Dries and Moschovakis 2004]. This tuple of partial functions is the first semantic value of α on $\mathbf{A}$, and it determines the associated expansion

$$(\mathbf{A}, \alpha) = (\mathbf{A}, \overline{f}_1, \dots, \overline{f}_K) = (A, 0, 1, \{\phi^{\mathbf{A}}\}_{\phi \in \Phi}, \overline{f}_1, \dots, \overline{f}_K).$$

It depends only on the body of α. The second is the n-ary partial function $\overline{\alpha} : A^n \rightharpoonup A$ *computed by* α,

$$\overline{\alpha}(\vec{a}) := \alpha_0(\vec{a}, \overline{f}_1, \dots, \overline{f}_K).$$

If $K = 0$, then $(\mathbf{A}, \alpha) = \mathbf{A}$ and $\overline{\alpha}$ is just the partial function explicitly defined by the head $\alpha_0(\vec{v})$. Given a relation $R \subseteq A^n$ and a set $S \subseteq A^n$ we say that α *decides* R on the set $S \subseteq A^n$ if for all $\vec{a} \in S$,

$$\overline{\alpha}(\vec{a}) = \begin{cases} 1 & \text{if } \vec{a} \in R, \\ 0 & \text{otherwise.} \end{cases}$$

Notation

We have exhibited the recursion variables $f_1, \dots, f_K$ in the parts of a recursive program, to emphasize that they can be used to define operations on partial functions. It will also be convenient to do this in the sequel sometimes, i.e., name $(\Phi, \vec{f})$-terms by expressions like $t(\vec{x}, \vec{f})$. Most of the times, however, when it is not needed, we will denote $(\Phi, \vec{f})$-terms as usual, $s(\vec{x}), u_1(\vec{x}), \ldots$, and for each Φ-algebra $\mathbf{A}$ we will interpret them in the associated expansion $(\mathbf{A}, \alpha)$. Thus each $(\Phi, \vec{f})$-term $s(\vec{x})$ defines on each Φ-algebra a partial function $\vec{a} \mapsto t(\vec{a})$ by the standard definitions of Section 5. If none of the recursion variables f_i occurs in $t(\vec{x})$, then (obviously) $t(\vec{a})$ has the same value in $\mathbf{A}$ and in $(\mathbf{A}, \alpha)$.

Examples

(1) Let $\mathbf{A} = \big(\mathbb{N}, 0, 1, \mathrm{S}, \mathrm{Pd}\big)$ where S is the successor function and Pd the predecessor (with $\mathrm{Pd}(0) = 0$). Then a partial function $f : \mathbb{N}^n \rightharpoonup \mathbb{N}$ is $\mathbf{A}$-computable exactly when it is computable by a Turing machine. This is the main result of [McCarthy 1963], where these programs were introduced.

(2) Recursive algorithms are sometimes most naturally defined by a single recursive equation. For example, the euclidean algorithm,

$$\gcd(x, y) = \begin{cases} y, & \text{if } \mathrm{rem}(x, y) = 0, \\ \gcd(y, \mathrm{rem}(x, y)), & \text{otherwise} \end{cases}$$

computes the greatest common divisor of two positive integers. To agree with the definition of programs given here, we add a trivial head which just "calls" the single function defined, leading in this case to the program

$$\begin{aligned} \varepsilon_0(x, y) &= \gcd(x, y), \\ \gcd(x, y) &= \mathrm{Co}(\mathrm{Eq}(\mathrm{rem}(x, y), 0), y, \gcd(y, \mathrm{rem}(x, y)) \end{aligned}$$

in the algebra $(\mathbb{Z}, 0, 1, \mathrm{rem})$. Formally, this is the pair of terms

$$\varepsilon = \big(\gcd(x, y),\ \mathrm{Co}(\mathrm{Eq}(\mathrm{rem}(x, y), 0), y, \gcd(y, \mathrm{rem}(x, y)))\big).$$

Complexity functions

For a Φ-algebra $\mathbf{A}$ and a recursive program α of signature $(\Phi, \vec{f})$, let $\Sigma = \Sigma^{\mathbf{A}}_{\alpha}$ be the set of *convergent* $(\Phi, \vec{f})$*-terms with parameters*, i.e., the set of all pairs $\big(t(\vec{x}), \vec{a}\big)$ such that $t(\vec{x}) = t(\vec{x}, \vec{f})$ is a $(\Phi, \vec{f})$-term, $\vec{a} \in A^m$, and $t(\vec{a})\downarrow$.

PROPOSITION 11.1. *There are two functions*

$$C = C^{\mathbf{A}}_{\alpha} : \Sigma \to \mathbb{N}, \quad D = D^{\mathbf{A}}_{\alpha} : \Sigma \to \mathbb{N}$$

which satisfy the following conditions.

(C0) $C(x_i, \vec{a}) = C(0, \vec{a}) = C(1, \vec{a}) = 0$

$D(x_i, \vec{a}) = D(0, \vec{a}) = D(1, \vec{a}) = 0$

(C1) $C(\mathrm{Eq}(t_1(\vec{x}), t_1(\vec{x})), \vec{a}) = \max\{C(t_1(\vec{x}), \vec{a}), C(t_2(\vec{x}), \vec{a})\}$

$D(\mathrm{Eq}(t_1(\vec{x}), t_1(\vec{x})), \vec{a}) = \max\{D(t_1(\vec{x}), \vec{a}), D(t_2(\vec{x}), \vec{a})\} + 1$

(C2) $$C(\mathrm{Co}(t_1(\vec{x}), t_2(\vec{x}), t_3(\vec{x})), \vec{a}) = \begin{cases} \max\{C(t_1(\vec{x}), \vec{a}), C(t_2(\vec{x}), \vec{a})\}, \\ \qquad \textit{if } t_1(\vec{a}) = 1, \\ \max\{C(t_1(\vec{x}), \vec{a}), C(t_3(\vec{x}), \vec{a})\}, \\ \qquad \textit{if } t_1(\vec{a}) \neq 1 \end{cases}$$

$$D(\mathrm{Co}(t_1(\vec{x}), t_2(\vec{x}), t_3(\vec{x})), \vec{a}) = \begin{cases} \max\{D(t_1(\vec{x}), \vec{a}), D(t_2(\vec{x}), \vec{a})\} + 1, \\ \qquad \textit{if } t_1(\vec{a}) = 1, \\ \max\{D(t_1(\vec{x}), \vec{a}), D(t_3(\vec{x}), \vec{a})\} + 1, \\ \qquad \textit{if } t_1(\vec{a}) \neq 1 \end{cases}$$

(C3) $C(\phi(t_1(\vec{x}), \dots, t_k(\vec{x})), \vec{a}) = \max\{C(t_1(\vec{x}), \vec{a}), \dots, C(t_k(\vec{x}), \vec{a})\} + 1,$

$$D(\phi(t_1(\vec{x}), \dots, t_k(\vec{x})), \vec{a}) = \max\{D(t_1(\vec{x}), \vec{a}), \dots, D(t_k(\vec{x}), \vec{a})\} + 1,$$

if $\phi \in \Phi$

(C4) $C(f_i(t_1(\vec{x}), \dots, t_m(\vec{x})), \vec{a}) = \max\{C(t_1(\vec{x}), \vec{a}), \dots, C(t_m(\vec{x}), \vec{a})\}$
$$+ C(\alpha_i(\vec{x}), t_1(\vec{a}), \dots, t_m(\vec{a}))$$

$$D(f_i(t_1(\vec{x}), \dots, t_m(\vec{x})), \vec{a}) \\ = \max\{D(t_1(\vec{x}), \vec{a}), \dots, D(t_m(\vec{x}), \vec{a}), D(\alpha_i(\vec{x}), t_1(\vec{a}), \dots, t_m(\vec{a}))\} + 1$$

PROOF. This follows by an application of the same *Fixed Point Lemma* in Section 1 of [van den Dries and Moschovakis 2004] which justifies the definitions of $\overline{f}_i$ and $\overline{\alpha}$ above. □

For $(t(\vec{x}), \vec{a}) \in \Sigma$, the *tree-depth* complexity $D(t(\vec{x}), \vec{a})$ is the depth of a *computation tree* which is naturally associated with the computation of $t(\vec{a})$ using the system of recursive equations (12). It is a natural (semantic) extension to programs of depth(t). We will not have much to say about it, but it provides a useful method of proving inductively properties of convergent, recursive terms.

The (strict) *complexity* $C(t(\vec{x}), \vec{a})$ measures (roughly) the *maximal depth of nested calls to the primitives* in the computation of $t(\vec{a})$. It is a natural extension of $\text{depth}_a(t)$ to recursive programs, and it is the main object of our study here.

LEMMA 11.2. *Let* $\mathbf{A}$ *be a* Φ*-algebra and* $t(\vec{x})$ *a* Φ*-term. For each* $\vec{a}$ *such that* $t(\vec{a})\downarrow$*,*

$$C(t(\vec{x}), \vec{a}) \leq \text{depth}_a(t(\vec{x})).$$

This is proved by a straightforward induction on depth(t). The Lemma is not true with equality (rather than $\leq$) because of the conditional, where the syntactic measure $\text{depth}_a(t(\vec{x}))$ counts the formal (putative) calls to the primitives in both branches, while the semantic measure $C(t(\vec{x}), \vec{a})$ has access to the parameters $\vec{a}$ and disregards the branch which is not used.

The *complexity function of a program* α is the complexity of its head term,

$$C_\alpha(\vec{a}) = C^{\mathbf{A}}_\alpha(\vec{a}) = C(\alpha_0(\vec{x}), \vec{a}).$$

It is defined when $\overline{\alpha}(\vec{a})\downarrow$.

The next proposition is the key tool for the lower bound results in Section 12. It is the version of Lemmas 5.1 and 5.3 appropriate for recursive programs.

PROPOSITION 11.3. *Let* $\mathbf{A}$ *be a* Φ*-algebra and* α *an* n*-ary* Φ*-program with recursion variables* $\vec{f}$*, and let* $t(\vec{x})$ *be a* $(\Phi, \vec{f})$*-term,* $\vec{x} = (x_1, \dots, x_m)$*.*

(i) *If $\imath : \mathbf{A} \rightarrowtail \mathbf{B}$ is an embedding of $\mathbf{A}$ into a Φ-algebra $\mathbf{B}$, then for every $\vec{a} \in A^m$ such that $t^{\mathbf{A}}(\vec{a})\downarrow$,*

$$\imath(t^{\mathbf{A}}(\vec{a})) = t^{\mathbf{B}}(\imath(\vec{a})) \text{ and } C^{\mathbf{A}}(t(\vec{x}), \vec{a}) = C^{\mathbf{B}}(t(\vec{x}), \iota(\vec{a})).$$

(ii) *If $\vec{a} \in A^m$, $t^{\mathbf{A}}(\vec{a})\downarrow$ and $C^{\mathbf{A}}(t(\vec{x}), \vec{a}) \le k$, then*

$$t^{\mathbf{A}}(\vec{a}) \in G_k^{\mathbf{A}}(\vec{a}) \text{ and } t^{\mathbf{A}}(\vec{a}) = t^{\mathbf{A} \upharpoonright G_k^{\mathbf{A}}(\vec{a})}(\vec{a}).$$

(iii) *If $\vec{a} \in A^n$ and k are such that $\overline{\alpha}(\vec{a}) = 1$ and $\overline{\alpha}(\imath(\vec{a})) \ne 1$ for some embedding $\imath : \mathbf{A} \upharpoonright G_k^{\mathbf{A}}(\vec{a}) \rightarrowtail \mathbf{A}$, then $C_\alpha(\vec{a}) > k$.*

Proof. (i) and (ii) are proved by routine inductions on $D(t(\vec{x}), \vec{a})$, and (iii) follows directly. $\square$

One easy corollary of it is the identification of explicit definability with recursive definability with bounded complexity:

Proposition 11.4. *Let $\mathbf{A}$ be a Φ-algebra. Given $f : A^n \rightharpoonup A$ and $S \subseteq D(f)$, the following are equivalent:*

(i) *there is a Φ-term $\tau(\vec{x})$ that computes f on S;*

(ii) *there is an n-ary Φ-program α that computes f on S such that the set $\{C_\alpha(\vec{a}) : \vec{a} \in S\}$ is bounded.*

Proof (Outline). (i) $\Longrightarrow$ (ii) by Lemma 11.2. For the converse, it is convenient first to add one more equation

$$f_0(\vec{x}) = \alpha_0(\vec{x}, f_1, \dots, f_K)$$

to the given program, so that $\overline{\alpha}(\vec{x}) = \overline{f}_0(\vec{x})$ and we can treat the head of α as part of its body. We first check that *each of the iterates $\overline{f}_i^{(s)}$ of α in a Φ-algebra $\mathbf{A}$ is defined by an explicit $(\Phi, \uparrow)$-term $t_{s,i}(\vec{x}, \uparrow)$* which does not depend on $\mathbf{A}$, where $\uparrow$ is a new symbol denoting the undefined, nullary function; this is done by a simple (if somewhat messy) argument that relies on replacement properties for partial function variables in $(\Phi, \vec{f}, \uparrow)$-terms. It follows that for each m, there is some $s = s(m)$ such that

$$\overline{f}_i(\vec{a}) = t_{s,i}^{\mathbf{A}}(\vec{a}, \uparrow) \quad (i = 0, \dots, K)$$

for every finite algebra $\mathbf{A}$ of size $\le m$, because of the monotonicity of the iterates; and then by (ii) of Proposition 11.3, if $C_\alpha(\vec{a}) \le k$ for all $\vec{a} \in S$, m is an upper bound on the size of $G_k^{\mathbf{A}}(\vec{a})$ (which depends only on the arities of the symbols in Φ, the arity n of α and k) and $s = s(m)$, then

$$\overline{f}_i(\vec{a}) = t_{s,i}^{\mathbf{A} \upharpoonright G_k^{\mathbf{A}}}(\vec{a}, \uparrow) \quad (\vec{a} \in S).$$

We finish the proof by setting $\tau(\vec{x}) = t_{s,0}(\vec{x}, 0)$, which (easily) has the same value as $t_{s,0}(\vec{x}, \uparrow)$ on S because we have assumed that $\overline{f}_0(\vec{a}) = t_{s,0}^{\mathbf{A} \upharpoonright G_k^{\mathbf{A}}}(\vec{a}, \uparrow)\downarrow$ when $\vec{a} \in S$. $\square$

Simulating Turing Machines with **Lin**-programs

We end this section with a simple simulation result which helps relate some of the lower bounds in the next section to classical complexity estimates.

PROPOSITION 11.5. *If a function $f : \mathbb{N} \to \mathbb{N}$ is computable by a Turing machine M in time $T(\log n)$ for $n > 0$, then there is a natural number k and a* **Lin***-program α which computes f with $c_\alpha(n) \le kT(\log n)$ for $n > 0$.*

In particular, if f is computable in polynomial time, then it is computable by a **Lin***-program α with $c_\alpha(n) \le k \log^d n$, for some natural numbers k, d and all $n > 1$.*

We are assuming here that the input n is entered on a tape of M in binary notation (which is why we express the complexity as a function of $\log n$), but other than that, the result holds in full generality: M may or may not have separate input and output tapes, it may have one or many, semi-infinite or infinite work tapes, etc. An analogous result holds also for functions of several variables.

PROOF (OUTLINE). Consider the simplest case, where M has a two-way infinite tape and only one symbol in its alphabet, 1. We use 0 to denote the blank symbol, so that the "full configuration" of a machine at a stage in a computation is a triple (q, τ, i), where q is a state, $\tau : \mathbb{Z} \to \{0, 1\}$, $\tau(j) = 0$ for all but finitely many j's, and $i \in \mathbb{Z}$ is the location of the scanned cell. If we write τ as a pair of sequences emanating from the scanned cell y_0

$$\cdots x_3 x_2 x_1 x_0 \underset{\uparrow}{y_0} y_1 y_2 y_3 \cdots$$

one "growing" to the left and the other to the right, we can then code (τ, i) by the pair of numbers

$$(x, y) = (\textstyle\sum_j x_j 2^j, \sum_j y_j 2^j).$$

Notice that $y_0 = \mathrm{rem}(y, 2)$ and $x_0 = \mathrm{rem}(x, 2)$, so that the scanned symbol and the symbol immediately to its left are computed from x and y by **Lin**-operations. The input configuration for the number z is coded by the triple $(q_0, 0, z)$, with q_0 the starting state, and all machine operations correspond to simple **Lin**-functions on these codes. For example:

$$\text{move to the right} : x \mapsto 2x + y_0, \quad y \mapsto \mathrm{iq}(y, 2),$$
$$\text{move to the left} : x \mapsto \mathrm{iq}(x, 2), \quad y \mapsto 2y + x_0.$$

It is now not difficult to write a **Lin**-program using these functions which simulates M.

If M has $K > 1$ symbols, we can either give essentially the same proof starting with a coding of tapes using numbers in K-adic expansions, or appeal to some of the standard reductions of K-symbol Turing machines to 1-symbol machines. The codings are somewhat messier when M has semi-infinite tapes. □

12. LOWER BOUNDS IN THE UNIFORM SETTING

In this section we obtain lower bounds on C_α for programs that decide relations like those considered earlier in the non-uniform setting of term-complexity. In fact, things are a bit simpler than before, and results take on a rather attractive form.

We collect in two theorems the uniform versions of Theorems 6.6, 8.1, 10.1, 7.1 and 9.1. Throughout this section we let a, b, p and λ denote *positive* integers.

THEOREM 12.1. *Suppose $R \subseteq \mathbb{Z}$ is a unary relation such that for all p, λ and some function $F : \mathbb{N}^{>0} \to \mathbb{N}^{>0}$,*

$$R(p) \implies \neg R(1 + \lambda F(\lambda)).$$

(i) *Let $\mathbf{A} := \mathbb{Z}(\mathrm{div}_D)$ where $D > 1$ is an integer. Then for each unary $\mathbf{A}$-program α that decides R on $\mathbb{N}^{>0}$,*

$$p > (D+1)^2 \text{ and } R(p) \implies C_\alpha(p) > \frac{\log p}{2 \log 2E} \quad (E := D!).$$

(ii) *Let $\mathbf{A} := \mathbb{Z}(1)$. Then for each unary $\mathbf{A}$-program α that decides R on $\mathbb{N}^{>0}$,*

$$p > 2^{25} \text{ and } R(p) \implies C_\alpha(p) > \frac{\log \log p}{2}.$$

(iii) *Let $\mathbf{A} := \mathbb{Z}(2)$. Then there is a constant $P > 0$* (independent of the relation R)*, such that for each unary $\mathbf{A}$-program α that decides R on $\mathbb{N}^{>0}$,*

$$p > P \text{ and } R(p) \implies C_\alpha(p) > \frac{\sqrt{\log\ \log\ p}}{2}.$$

Note that the hypothesis on F is weaker than in Theorems 6.1, 8.1 and 10.1, since no polynomial growth is assumed.

PROOF. All the real work for these results has been done in Sections 6, 8, 10 in constructing certain embeddings $\mathbf{A} \restriction U \rightarrowtail \mathbf{A}$. We only show (iii), since the other proofs are similar.

Let $\mathbf{A} = \mathbb{Z}(2)$ and let α be a unary $\mathbf{A}$-program that decides R on $\mathbb{N}^{>0}$. As in the proof of Theorem 10.1, let $p \geq 2^{2^{12}}$ be such that $R(p)$ holds; take m maximal such that $p \geq 2^{2^{3(m+1)^2}}$, so $m \geq 1$ and $p < 2^{2^{3(m+2)^2}}$, so $\log \log p < 3(m+2)^2$, hence $\frac{\sqrt{\log \log p}}{\sqrt{3}} < m+2$. Since $\frac{1}{2} < \frac{1}{\sqrt{3}}$, this gives a natural number $P \geq 2^{2^{12}}$ (independent of the relation R) such that $\frac{\sqrt{\log \log p}}{2} < m+1$ if $p > P$. Also, $G_m^{\mathbf{A}}(p) \subseteq C_m(p)$, using notations from Section 10. Put $U := G_m^{\mathbf{A}}(p)$, and take $\lambda := \prod_{u \in U} c(u)$, using the notations of Lemma 10.4. That lemma provides an embedding $\imath : \mathbf{A} \restriction U \rightarrowtail \mathbf{A}$ such that $\imath(p) = \kappa p$ where $\kappa = 1 + \lambda F(\lambda)$, so by (iii) of Proposition 11.3, we have $C_\alpha(p) \geq m + 1 > \frac{\sqrt{\log \log p}}{2}$ if $p > P$. □

THEOREM 12.2. (i) *If α is a binary program which decides coprimeness on $(\mathbb{N}^{>0})^2$ in $\mathbb{Z}(\mathrm{div}_D)$ (with $D > 1$), then*

$$p > (D+1)^2 \implies C_\alpha(p, p^2+1) > \frac{\log p}{2 \log 2E} \quad (E := D!).$$

(ii) *If α is a binary program which decides coprimeness on $(\mathbb{N}^{>0})^2$ in $\mathbb{Z}(1)$ and a, b are coprime such that $|\frac{a}{b} - \sqrt{2}| < \frac{1}{b}$, then*

$$a \geq 2^{2^{16}} \implies C_\alpha(a, b) > \frac{\log \log a}{4}.$$

PROOF. We only show (ii), since the proof of (i) is similar. Assume the hypotheses in (ii) on α, a and b, with $a \geq 2^{2^{16}}$, and take m maximal with $a \geq 2^{2^{3m+4}}$, so $m \geq 1$ and $a < 2^{2^{3(m+1)+4}}$, hence $\log\log a < 3(m+1)+4$, and thus

$$m+1 > \frac{\log\log a}{3} - \frac{4}{3} \geq \frac{\log\log a}{4}.$$

Setting $h := 2^{2^m}$ and $\kappa := 1 + (h^2)!$, Lemmas 9.2 and 9.3 yield an embedding $\imath : \mathbb{Z}(1) \restriction G_m(a,b) \rightarrowtail \mathbb{Z}(1)$ such that $\imath(a) = \kappa a$ and $\imath(b) = \kappa b$. Then (iii) of Proposition 11.3 gives $C_\alpha(a,b) \geq m+1 > \frac{\log\log a}{4}$, as promised. □

Uniform vs. non-uniform lower bounds

It is plausible that uniform algorithms are in general less efficient on inputs $\leq N$ than some non-uniform term algorithm $t_N(\vec{x})$, so we would generally expect higher (better) lower bounds for uniform algorithms. In the case of $\mathbb{Z}(\mathrm{div}_D)$, however, with Presburger primitives, we derived in Theorems 12.1 (i) and 12.2 (i) logarithmic lower bounds for recursive programs which are the same as the non-uniform lower bounds of Theorems 6.6 and 7.1, up to a multiplicative constant. For $\mathbb{Z}(1)$ and $\mathbb{Z}(2)$, we also derived lower bounds of the same form as in Part 2, but for the complexity function $C_\alpha(\vec{x})$, which corresponds to *arithmetic complexity* for terms and is generally lower than the corresponding uniform version of *term complexity*—for which better lower bounds may well be derivable. The cost in increased complexity for uniformity is an important problem, but this indication of its form (and what causes it) is about the best we can do now, in the absence of proofs of optimality for the known uniform algorithms, like the euclidean.

A refinement

We are now going to extend part (ii) of Theorem 12.2 in two directions. First, the embedding lemma 9.3 on which it depends does not assume that a and b are coprime, and we shall take advantage of that. Secondly, we increase the set of inputs for which the lower bound holds by considering $a > b$ with $|\sqrt{2} - \frac{a}{b}| < \frac{1}{b^\rho}$ where ρ is a real parameter, $0 < \rho \leq 1$. The price to pay is that we have to restrict to inputs with $b > b(\rho)$ where $b(\rho) \to \infty$ as $\rho \to 0$. Here is a precise statement.

THEOREM 12.3. *Let $0 < \rho \leq 1$. Then there exists $b(\rho) > 0$ such that if $a > b > b(\rho)$ and $|\sqrt{2} - \frac{a}{b}| < \frac{1}{b^\rho}$, then for some λ and every binary $\mathbb{Z}(1)$-program α, if $\overline{\alpha}(a,b) = 1$ and $\overline{\alpha}\big((n\lambda+1)a, (n\lambda+1)b\big) \neq 1$ for some $n > 0$, then $c_\alpha(a,b) > \frac{\log\log a}{4}$.*

PROOF. In connection with Theorem 3.9 it was already mentioned that the lemmas of Section 3 have ρ-versions. This leads to ρ-versions of Lemmas 9.2 and 9.3: in the first of these lemmas the inequality $a \geq h^{16}$ should be replaced by $a^\rho \geq h^{16}$, and the statement of the second lemma can be kept, with "previous lemma" referring now to its ρ-version. The proofs are the same, *mutatis mutandis.*

The rest of the proof follows that of part (ii) of Theorem 12.2 but is included here to show how $b(\rho)$ depends on ρ and λ on ρ, a. Assume that $a > b$ and $|\sqrt{2} - \frac{a}{b}| < \frac{1}{b^\rho}$, with $a^\rho \geq 2^{2^{16}}$, and take m maximal with $a^\rho \geq 2^{2^{3m+4}}$, so $m \geq 1$ and $a^\rho < 2^{2^{3(m+1)+4}}$, hence

$$\log\rho + \log\log a < 3(m+1)+4,$$

and thus

$$m+1 > \frac{\log\log a}{3} + \frac{\log \rho}{3} - \frac{4}{3} \geq \frac{\log\log a}{4}$$

where the last inequality holds provided b (and hence a) is large enough, say $b > b(\rho)$, which we assume in the rest of the proof. Set $h := 2^{2^m}$ and $\lambda := (h^2)!$. Let α be a binary $\mathbb{Z}(1)$-program and let $n > 0$ be such that $\overline{\alpha}(a,b) = 1$ and $\overline{\alpha}\big((n\lambda+1)a, (n\lambda+1)b\big) \neq 1$. With $\kappa := n\lambda + 1$, the ρ-versions of Lemmas 9.2 and 9.3 yield an embedding

$$\imath : \mathbb{Z}(1) \restriction G_m(a,b) \rightarrowtail \mathbb{Z}(1)$$

such that $\imath(a) = \kappa a$ and $\imath(b) = \kappa b$. Then (iii) of Proposition 11.3 gives $C_\alpha(a,b) \geq m+1 > \frac{\log\log a}{4}$, as promised. □

Corollary 12.4. *Let $0 < \rho \leq 1$, and let $\ell : \mathbb{N}^{>1} \to \mathbb{R}^{>1}$ be a function such that $\ell(n) = o(n)$ as $n \to \infty$. Then there exists a $b(\rho) > 0$ (independent of ℓ) such that for each binary $\mathbb{Z}(1)$-program α that decides the binary relation*

$$\{(x,y) \in \mathbb{Z}^2 : x > 0,\ y > 0 \text{ and } \gcd(x,y) < \ell(x+y)\}$$

on $(\mathbb{N}^{>0})^2$ and all a, b, if $a > b > b(\rho)$, $|\sqrt{2} - \frac{a}{b}| < \frac{1}{b^\rho}$, and $\gcd(a,b) < \ell(a+b)$, then $c_\alpha(a,b) > \frac{\log\log\ a}{4}$.

Proof. Take $b(\rho)$ as in Theorem 12.3, let α be a binary $\mathbb{Z}(1)$-program α that decides the relation indicated, and let $a > b > b(\rho)$, $|\sqrt{2} - \frac{a}{b}| < \frac{1}{b^\rho}$, and $\gcd(a,b) < \ell(a+b)$. Note that then $\overline{\alpha}(a,b) = 1$. Take λ with the property of Theorem 12.3. The assumption on ℓ allows us to take an n so large that $n\lambda+1 \geq \ell\big((n\lambda+1)(a+b)\big)$. Then

$$\gcd\big((n\lambda+1)a, (n\lambda+1)b\big) \geq \ell\big((n\lambda+1)a + (n\lambda+1)b\big),$$

so $\overline{\alpha}\big((n\lambda+1)a, (n\lambda+1)b\big) \neq 1$, and thus $c_\alpha(a,b) > \frac{\log\log\ a}{4}$. □

Theorem 12.2(ii) is the case where $\rho = 1$ and $\ell(n) = 2$ for all $n > 1$, but we can also take $\ell(n) := \sqrt{n}$, giving rise to the relation

$$x > 0,\ y > 0 \text{ and } \gcd(x,y) < \sqrt{x+y},$$

and $\ell(n) := n/\log\ n$, with corresponding relation

$$x > 0,\ y > 0 \text{ and } \gcd(x,y) < (x+y)/\log(x+y).$$

Upper bounds

Theorem 12.1 yields lower bounds for programs deciding primeness, being a power of 2, being a perfect square, and being square-free. Our knowledge on *upper bounds* for such programs is as follows.

Primeness is decidable in polynomial time by [Agrawal et al. 2004], and so by Proposition 11.5, it is decided by a **Lin**-program α with $C_\alpha(n) \leq C(\log n)^d$ for some positive constants C and d and all $n > 1$. The lower bound of Theorem 12.1 comes within a power of this upper bound, and we do not know of a better lower bound in the literature.

There is an obvious $\mathbb{Z}(\text{div}_2)$-program α that decides "being a power of 2" with $C_\alpha(n) \leq C\log n$ for some positive constant C and all $n > 0$, so for this unary relation, the lower bound of Theorem 12.1 is matched by a proportional upper bound.

From [Mansour et al. 1991b] one can extract a $\mathbb{Z}(2)$-program α that decides "being a perfect square" with $C_\alpha(n) \leq C \log\log n$ for some positive constant C and all sufficiently large n, while Theorem 12.1 gives a $\sqrt{\log\log}$ lower bound.

It seems to be open whether "being square-free" is even decidable in polynomial time.

For coprimeness, the lower bound in (i) of Theorem 12.2 is matched by a proportional upper bound for the binary gcd algorithm. For $\mathbb{Z}(1)$, we do not know of any work which gets closer to the euclidean log upper bound than the log log lower bound in (ii) of Theorem 12.2.

REFERENCES

AGRAWAL, M., KAYAL, N., AND SAXENA, N. 2004. Primes is in P. *Annals of Mathematics, Second Series 160*, 781–793.

VAN DEN DRIES, L. 2003. Generating the greatest common divisor, and limitations of primitive recursive algorithms. *Foundations of computational mathematics 3*, 297–324.

VAN DEN DRIES, L. AND MOSCHOVAKIS, Y. N. 2004. Is the Euclidean algorithm optimal among its peers? *Bulletin of Symbolic Logic 10*, 390–418.

KNUTH, D. E. 1973. *The Art of Computer Programming. Fundamental Algorithms*, Second ed. Addison-Wesley.

MANSOUR, Y., SCHIEBER, B., AND TIWARI, P. 1991a. A lower bound for integer greatest common divisor computations. *Journal of the Association for Computing Machinery 38*, 453–471.

MANSOUR, Y., SCHIEBER, B., AND TIWARI, P. 1991b. Lower bounds for computations with the floor operation. *SIAM Journal on Computing 20*, 315–327.

MCCARTHY, J. 1963. A basis for a mathematical theory of computation. In *Computer programming and formal systems*, P. Braffort and D. Herschberg, Eds. North-Holland, 33–70.

MEIDÂNIS, J. 1991. Lower bounds for arithmetic problems. *Information Processing Letters 38*, 83–87.

MOSCHOVAKIS, Y. N. 2003. On primitive recursive algorithms and the greatest common divisor function. *Theoretical Computer Science 301*, 1–30.

POIZAT, B. 1995. *Les Petits Cailloux.* Nur al-Mantiq wal-Ma'rifah.

ROSE, H. 1994. *A Course in Number Theory*, second ed. Clarendon Press.

SMORYŃSKI, C. 1991. *Logical Number Theory I.* Springer.

VON ZUR GATHEN, J. AND GERHARD, J. 1999. *Modern Computer Algebra.* Cambridge University Press.

Received August 2006; revised March 2007; accepted March 2007