Fundamenta Informaticae 76 (2007) 1–11
IOS Press

On the Optimality of the Binary Algorithm for the Jacobi Symbol

J. Busch[*]
Department of Mathematics
University of California, Los Angeles
Los Angeles, CA, U.S.A.
jbusch@math.ucla.edu

Abstract. We establish lower bounds on the complexity of computing the following number-theoretic functions and relations from piecewise linear primitives: (i) the Legendre and Jacobi symbols, (ii) pseudoprimality, and (iii) modular exponentiation. As a corollary to the lower bound obtained for (i), an algorithm of Shallit and Sorenson is optimal (up to a multiplicative constant).

Keywords: arithmetic complexity, computational complexity, Jacobi symbol, lower bounds

1. Introduction

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

$$\textbf{\textit{Lin}} = \{+, -, 2\cdot, \mathrm{iq}(x, 2), \mathrm{parity}, \chi_<, \chi_=\} \tag{1}$$

of given, primitive operations, where $\mathrm{iq}(x, 2)$ is the integer quotient in the division of x by 2, $\mathrm{parity}(x)$ is the non-negative remainder in the division of x by 2, and $\chi_<$ and $\chi_=$ are the characteristic functions of the binary relations $<$ and $=$, respectively. We aim to establish lower bounds on the number of calls to the functions in ***Lin*** in the computation of (i)–(iii).

There are few results in the literature which give lower bounds on the complexity of number-theoretic computations relative to a fixed set of given arithmetic operations. Other work in this area includes [12],

Address for correspondence: UCLA Mathematics Department, Box 951555, Los Angeles, CA 90095-1555, U.S.A.
[*]Supported by National Science Foundation VIGRE grant DMS-0502315

where the techniques of [9] and [10] are used to derive lower bounds on the depth of decision trees and the time complexity of random access machines (RAMs) which compute (i)–(iii). These computations are relative to a set of operations which is richer than ***Lin***, and consequently the lower bounds established there—for input values $\leq n$, $\Omega(\log\log\log n)$ for the Jacobi symbol, and $\Omega(\sqrt{\log\log n})$ for pseudoprimality and modular exponentiation—are lower than the ones of the present paper.

To formalize the notion of computation relative to given functions, we assume that algorithms are expressed as McCarthy-style recursive programs [11] over partial algebras of the form

$$\mathbf{A} = (A; 0, 1, \{\phi^{\mathbf{A}}\}_{\phi\in\Phi}), \tag{2}$$

where A is a non-empty set, $0, 1 \in A$ are distinct, and for each formal function symbol ϕ of the signature Φ, $\phi^{\mathbf{A}} : A^n \rightharpoonup A$ is a partial function on A. The lower bounds obtained in this context are quite general, and hold, e.g., for the RAM model of computation. This claim will be made precise in Section 5. Φ-terms E are generated by the recursion

$$E ::= \mathsf{0} \mid \mathsf{1} \mid \mathsf{v} \mid \phi(E_1, \ldots, E_{n_\phi}) \mid \zeta(E_1, \ldots, E_{n_\zeta}) \mid (\mathsf{if}\ (E_1 = \mathsf{0})\ \mathsf{then}\ E_2\ \mathsf{else}\ E_3), \tag{3}$$

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

$$\alpha : \begin{cases} \zeta_0(\vec{\mathsf{v}}_0) &=& E_0(\vec{\mathsf{v}}_0, \zeta_1, \ldots, \zeta_k), \\ \zeta_1(\vec{\mathsf{v}}_1) &=& E_1(\vec{\mathsf{v}}_1, \zeta_1, \ldots, \zeta_k), \\ &\vdots& \\ \zeta_k(\vec{\mathsf{v}}_k) &=& E_k(\vec{\mathsf{v}}_k, \zeta_1, \ldots, \zeta_k), \end{cases} \tag{4}$$

where the variables which occur in the terms are among those displayed. A system of monotone and continuous functional equations is naturally associated to the formal system (4), and has least fixed points $\overline{\zeta}_0, \ldots, \overline{\zeta}_k$ in $\mathbf{A}$ by well-known results. The partial function computed by α in $\mathbf{A}$ is $\overline{\alpha}^{\mathbf{A}}(\vec{x}) = \overline{\zeta}_0(\vec{x})$. It will be convenient to use the model-theoretic notation

$$\mathbf{A} \models \alpha(\vec{x}) = w \iff \overline{\alpha}^{\mathbf{A}}(\vec{x}) = w. \tag{5}$$

During a computation, a recursive program may perform logical operations, and may also call a given function $\phi^{\mathbf{A}}$, which is regarded as an oracle, and which returns the correct value in one step. If α computes the partial function $\overline{\alpha}$ in a partial algebra $\mathbf{A}$, and $\overline{\alpha}$ converges on input $\vec{x}$, then the basic complexity $c_\alpha^{\mathbf{A}}(\vec{x})$ (or simply $c_\alpha(\vec{x})$ if the underlying algebra is clear) is the maximum number of nested calls to the given functions $\{\phi^{\mathbf{A}}\}_{\phi\in\Phi}$ made by α in the computation of $\overline{\alpha}^{\mathbf{A}}(\vec{x})$ (for a precise, recursive definition of c_α, see [4]). Note that c_α does not count logical steps. It is an abstract measure of strict, parallel, call-by-value time complexity which is dominated by many other common measures, e.g., it is no larger than the total number of calls to the given operations in any sequential or parallel implementation. We extend the model-theoretic notation of (5) as follows:

$$\mathbf{A} \models \alpha^{(m)}(\vec{x}) = w \iff \overline{\alpha}^{\mathbf{A}}(\vec{x}) = w \text{ and } c_\alpha^{\mathbf{A}}(\vec{x}) \leq m. \tag{6}$$

Let $\mathbf{Z} = (\mathbb{Z}; 0, 1, \boldsymbol{Lin})$. The binary algorithm for the Jacobi symbol from [17] can be expressed as a recursive program β over $\mathbf{Z}$ which computes $\left(\frac{a}{n}\right)$ with $c_\beta(a, n) = O(\log\max(a, n))$. This algorithm is described as "probably the most efficient in practice" in [13]. We show in Section 3 that it is optimal: there is a rational constant $C > 0$ and an infinite set $\{(a_k, n_k)\}_{k\in\mathbb{N}}$ such that if α is any recursive program over **Z** which computes the Jacobi symbol, then for all $k \in \mathbb{N}$,

$$c_\alpha(a_k, n_k) > C \log_2 \max(a_k, n_k). \tag{7}$$

In particular, we have that $c_\alpha(a, n) = \Omega_\infty(\log \max(a, n))$, but note that the result is stronger than this, since both the constant C and the input values $\{(a_k, n_k)\}_{k\in\mathbb{N}}$ are independent of α.

For the other functions that we consider, the following upper bounds are known. There is a recursive program μ over **Z** which expresses the binary method for modular exponentiation from [2], such that $\overline{\mu}^{\mathbf{Z}}(b, c, m) = b^c \pmod m$ with $c_\mu(b, c, m) = O(\log^2 m)$. Since deciding whether m is pseudoprime to the base b is reducible to modular exponentiation, the $O(\log^2 m)$ upper bound also holds for pseudoprimality. The lower bounds we obtain in Section 4 for this function and relation are both $\Omega_\infty(\log m)$ where, as above, both the constant and the infinitely many values absorbed by the Ω_∞-notation are independent of the recursive program.

During the review of this paper, one referee quite correctly observed that the proof of Theorem 4.1 also establishes an $\Omega_\infty(\log a)$-lower bound for deciding the predicate "b divides a". There is a known algorithm for deciding this predicate from ***Lin*** with basic complexity $O(\log a)$, and so this furnishes a new proof of the previously known result [14] that this algorithm is optimal.

In Section 2, we establish the existence of a particular imbedding which is parameterized by a single natural number. Following the technique developed in [4], we use this imbedding, with suitably chosen values for the parameter, to obtain lower bounds for (i)–(iii) in Sections 3 and 4.

2. An asymmetric imbedding

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

$$\begin{aligned} G_0^{\mathbf{A}}(X) &= \{0, 1\} \cup X, \\ G_{m+1}^{\mathbf{A}}(X) &= G_m^{\mathbf{A}}(X) \cup \{\phi^{\mathbf{A}}(\vec{x}) \mid \vec{x} \in G_m^{\mathbf{A}}(X), \phi \in \Phi\}. \end{aligned} \tag{8}$$

In particular, for $X \subseteq \mathbb{Z}$, the elements of $G_m^{\mathbf{Z}}(X)$ are precisely the values that are generated in **Z** from X in m steps. For $\vec{a} \in \mathbb{Z}^n$, let

$$B_m^{\mathbf{Z}}(\vec{a}) = \left\{ \frac{c_0 + c_1 a_1 + \cdots + c_n a_n}{2^m} \in \mathbb{Z} \mid c_i \in \mathbb{Z}, |c_i| \le 2^{2m} \right\}. \tag{9}$$

Lemma 2.1. For each $m \in \mathbb{N}$, $G_m^{\mathbf{Z}}(\vec{a}) \subseteq B_m^{\mathbf{Z}}(\vec{a})$.

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

Lemma 2.2. Let $p > 1$, $q \ge p$, and $a = 2^{\lfloor \log_2 p \rfloor + 1} q$. Let $x, y, z \in \mathbb{Z}$, $|x|, |y|, |z| < \frac{p}{2}$, $\lambda \ge 1$. Then

$$x + yp + \lambda z a > 0 \iff [z > 0] \vee [z = 0 \wedge y > 0] \vee [z = y = 0 \wedge x > 0]. \tag{10}$$

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

Now assume that $z \neq 0$. Since

$$a = 2^{\lfloor \log_2 p \rfloor + 1} q > 2^{\log_2 p} q = pq \geq p^2, \tag{11}$$

we have that

$$|x + yp| \leq \frac{p}{2} + \frac{p^2}{2} < p^2 \leq |z|p^2 < |z|a \leq \lambda |z| a, \tag{12}$$

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

Lemma 2.3. Let $p > 1$, $q \geq p$, and $a = 2^{\lfloor \log_2 p \rfloor + 1} q$. Suppose that $\lambda \geq 1$, $x_i, y_i \in \mathbb{Z}$ and $|x_i|, |y_i| < \frac{p}{4}$ for $i = 0, 1, 2$. Then

$$x_0 + x_1 p + \lambda x_2 a = y_0 + y_1 p + \lambda y_2 a \iff x_i = y_i \text{ for } i = 0, 1, 2, \tag{13}$$

and

$$\begin{aligned} &x_0 + x_1 p + \lambda x_2 a > y_0 + y_1 p + \lambda y_2 a \iff \\ &\quad [x_2 > y_2] \vee [x_2 = y_2 \wedge x_1 > y_1] \vee [x_2 = y_2 \wedge x_1 = y_1 \wedge x_0 > y_0]. \end{aligned} \tag{14}$$

Proof:
Immediate, by Lemma 2.2. □

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

$$\phi^{\mathbf{A}}(\vec{x}) = w \implies \phi^{\mathbf{B}}(\iota(\vec{x})) = \iota(w), \tag{15}$$

It follows (for a proof, see [4]) that if $\iota : \mathbf{A} \rightarrowtail \mathbf{B}$ is an imbedding, and α is a recursive program over a partial algebra with signature Φ, then

$$\mathbf{A} \models \alpha^{(m)}(\vec{x}) = w \implies \mathbf{B} \models \alpha^{(m)}(\iota(\vec{x})) = \iota(w). \tag{16}$$

If $0^{\mathbf{A}}, 1^{\mathbf{A}} \in U \subseteq A$, then the partial subalgebra $\mathbf{A} \upharpoonright U = (U; 0^{\mathbf{A}}, 1^{\mathbf{A}}, \{\phi^{\mathbf{A} \upharpoonright U}\}_{\phi \in \Phi})$ is defined so that for all $\vec{x}, w \in A$,

$$\phi^{\mathbf{A} \upharpoonright U}(\vec{x}) = w \iff \vec{x}, w \in U \text{ and } \phi^{\mathbf{A}}(\vec{x}) = w. \tag{17}$$

Lemma 2.4. If $2^{2m+3} < p$, $q \geq p$, $a = 2^{\lfloor \log_2 p \rfloor + 1} q$, and $\lambda \geq 1$, then there exists an imbedding $\iota : \mathbf{Z} \upharpoonright G_m^{\mathbf{Z}}(p, a) \rightarrowtail \mathbf{Z}$ given by

$$\iota\left(\frac{x_0 + x_1 p + x_2 a}{2^m}\right) = \frac{x_0 + x_1 p + \lambda x_2 a}{2^m}. \tag{18}$$

In particular, $\iota(p) = p$, $\iota(a) = \lambda a$.

Proof:
By Lemma 2.1, each $x \in G^{\mathbf{Z}}_m(p,a)$ may be expressed

$$x = \frac{x_0 + x_1 p + x_2 a}{2^m}, \tag{19}$$

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

$$|x_i| \le 2^{2m} < \frac{2^{2m+3}}{4} < \frac{p}{4}. \tag{20}$$

Thus, Lemma 2.3 implies that ι is a well-defined, order-preserving injection with domain $G^{\mathbf{Z}}_m(p,a)$. To see that $\mathrm{range}(\iota) \subseteq \mathbb{Z}$, observe that $\lfloor \log_2 p \rfloor \ge 2m+3$, and therefore $2^{2m+3} \mid a$. Now,

$$2^m \mid x_0 + x_1 p + x_2 a, \tag{21}$$

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

$$2^m \mid (x_0 + x_1 p + x_2 a) + (\lambda - 1)x_2 a = x_0 + x_1 p + \lambda x_2 a. \tag{22}$$

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

The proof will be complete once we show that (15) holds for each of the functions in ***Lin***. We treat the case for $\mathrm{iq}(x,2)$, and verification of the other cases is similar.

Suppose that $x, y = \mathrm{iq}(x,2) \in G^{\mathbf{Z}}_m(p,a)$. Let

$$x = \frac{x_0 + x_1 p + x_2 a}{2^m}, \quad y = \frac{y_0 + y_1 p + y_2 a}{2^m}, \tag{23}$$

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

$$\mathrm{iq}(x,2) = \frac{1}{2}\left(\frac{x_0 + x_1 p + x_2 a}{2^m} - 1\right) = \frac{x_0 - 2^m + x_1 p + x_2 a}{2^{m+1}} = y. \tag{24}$$

Since

$$|x_0 - 2^m| \le 2^{2m} + 2^m \le 2^{2m+1} = \frac{2^{2m+3}}{4} < \frac{p}{4}$$

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

$$2y_0 = x_0 - 2^m, 2y_1 = x_1, 2y_2 = x_2. \tag{25}$$

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

$$\mathrm{iq}(\iota(x),2) = \frac{x_0 - 2^m + x_1 p + \lambda x_2 a}{2^{m+1}} = \frac{2y_0 + 2y_1 p + 2\lambda y_2 a}{2^{m+1}} = \iota(y). \tag{26}$$

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

3. Legendre and Jacobi symbols

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

$$x^2 \equiv a \pmod{m}, \tag{27}$$

then a is said to be a quadratic residue modulo m. If p is an odd prime, then the Legendre symbol $\left(\frac{a}{p}\right)$ is

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is a quadratic residue modulo } p \text{ and } p \nmid a, \\ 0 & \text{if } p \mid a, \\ -1 & \text{otherwise.} \end{cases} \tag{28}$$

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

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{k_1} \cdots \left(\frac{a}{p_n}\right)^{k_n}. \tag{29}$$

Proposition 3.1. (Shallit and Sorenson [17])
The recursive program β below computes the Jacobi symbol $\left(\frac{a}{n}\right)$ over **Z** with

$$c_\beta(a, n) = O(\log \max(a, n)), \tag{30}$$

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

$$\zeta_\beta(a, n) = \begin{cases} 0 & \text{if } a = 0 \text{ and } n \neq 1, \\ 1 & \text{if } a = 0 \text{ and } n = 1, \\ \zeta_\beta(a/2, n) & \text{if } a \text{ is even, and} \\ & \quad n \equiv 1 \pmod 8 \text{ or } n \equiv 7 \pmod 8, \\ -\zeta_\beta(a/2, n) & \text{if } a \text{ is even, and} \\ & \quad n \equiv 3 \pmod 8 \text{ or } n \equiv 5 \pmod 8, \\ \zeta_\beta(a - n, n) & \text{if } a > n, \\ \zeta_\beta(n - a, a) & \text{if } a \not\equiv 3 \pmod 4 \text{ or } n \not\equiv 3 \pmod 4, \\ -\zeta_\beta(n - a, a) & \text{if } a \equiv 3 \pmod 4 \text{ and } n \equiv 3 \pmod 4. \end{cases} \tag{31}$$

Proof:
The correctness of the algorithm is shown in [17], and follows from basic properties of the Jacobi symbol which may be found, e.g., in [7] and [8].

Observe that the remainder in division by 4, the remainder in division by 8, and multiplication on $\{-1, 0, 1\}$ are all definable by ***Lin***-terms of fixed, finite depth, and therefore have constant complexity. In particular, β is a program over **Z**. To determine its complexity, notice that in computing $\left(\frac{a}{n}\right)$ from (31), at worst, one of the arguments a, n is halved at every other step. Each such step involves no more than some fixed number of calls to the functions in ***Lin***, and so $c_\beta(a, n) = O(\log a + \log n) = O(\log \max(a, n))$.
□

Theorem 3.1. Let α be a recursive program over $\mathbf{Z}$ such that for all $(a,p) \in \mathbb{Z}^2$ with p an odd prime,

$$\mathbf{Z} \models \alpha(a,p) = \left(\frac{a}{p}\right), \tag{32}$$

where (-) is the Legendre symbol. There is an infinite sequence $\{(a_k, p_k)\}_{k\in\mathbb{N}}$ such that for all $k \in \mathbb{N}$, p_k is prime, $p_k \nmid a_k$, and

$$c_\alpha(a_k, p_k) > \frac{1}{20} \log_2 \max(a_k, p_k). \tag{33}$$

Proof:
Since 3 and 8 are relatively prime, we may fix, by Dirichlet's theorem on the prime numbers of an arithmetic progression, an increasing enumeration $\{p_k\}_{k\in\mathbb{N}}$ of primes such that each $p_k \equiv 3 \pmod 8$. For each k, let q_k be the least prime greater than p_k, and let $a_k = 2^{\lfloor \log_2 p_k \rfloor + 1} q_k$. Clearly, $p_k \nmid a_k$.

Fix $k \in \mathbb{N}$. Let $(a,p) = (a_k, p_k)$ and $m = c_\alpha(a,p)$. Suppose for a contradiction that $2^{2m+3} \leq p$. Then there is an imbedding ι as defined in Lemma 2.4 with $\lambda = 2$. We have

$$\mathbf{Z} \models \alpha^{(m)}(a,p) = \left(\frac{a}{p}\right). \tag{34}$$

So by Absoluteness [4, Lemma 2],

$$\mathbf{Z} \restriction G_m^{\mathbf{Z}}(a,p) \models \alpha(a,p) = \left(\frac{a}{p}\right). \tag{35}$$

Absoluteness expresses a general fact about programs: if γ is a recursive program over any partial algebra $\mathbf{A}$, and $\mathbf{A} \models \gamma^{(k)}(\vec{x}) = w$, then the entire "computation" of $\overline{\gamma}^{\mathbf{A}}(\vec{x})$ takes place in $G_k^{\mathbf{A}}(\vec{x})$, and moreover $\mathbf{A} \restriction G_k^{\mathbf{A}}(\vec{x}) \models \gamma^{(k)}(\vec{x}) = w$.

By (16),

$$\mathbf{Z} \models \alpha(\iota(a), \iota(p)) = \iota\left(\frac{a}{p}\right), \tag{36}$$

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

$$\mathbf{Z} \models \alpha(\iota(a), \iota(p)) = \left(\frac{\iota(a)}{\iota(p)}\right), \tag{37}$$

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

$$\left(\frac{\iota(a)}{\iota(p)}\right) = \left(\frac{2a}{p}\right) = -\left(\frac{a}{p}\right). \tag{38}$$

Thus

$$\mathbf{Z} \models \alpha(\iota(a), \iota(p)) = -\left(\frac{a}{p}\right), \tag{39}$$

a contradiction. Therefore $2^{2m+3} > p$, and so $2m + 3 > \log_2 p$. Since $-1 \notin G_0(a,p)$, we have that $m > 0$, and so $5m \geq 2m + 3 > \log_2 p$. Hence

$$c_\alpha(a,p) > \frac{1}{5} \log_2 p. \tag{40}$$

By a straightforward computation, this yields

$$c_\alpha(a,p) > \frac{1}{20}\log_2 a = \frac{1}{20}\log_2 \max(a,p). \tag{41}$$

□

Corollary 3.1. Let β be the recursive program defined by (31). There is a rational number $r > 0$ and an infinite sequence $\{(a_k, p_k)\}_{k\in\mathbb{N}}$ such that if α is any recursive program over **Z** which computes the Jacobi symbol, then for every $k \in \mathbb{N}$,

$$c_\alpha(a_k,p_k) > rc_\beta(a_k,p_k). \tag{42}$$

Proof:
Let $\{(a_k, p_k)\}_{k\in\mathbb{N}}$ be as in Theorem 3.1, and by Proposition 1, let $M \in \mathbb{N}$ be such that for all $a, n > 0$ with odd n, $c_\beta(a,n) \le M \log_2 \max(a,n)$. For pairs (a,n) with prime n, the Jacobi symbol agrees with the Legendre symbol, and so for all $k \in \mathbb{N}$,

$$c_\alpha(a_k,p_k) > \frac{1}{20}\log_2\max(a_k,p_k) \ge \frac{1}{20M}c_\beta(a_k,p_k). \tag{43}$$

□

4. Pseudoprimality and modular exponentiation

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

$$\mathrm{PSP}(a,b) = \begin{cases} 1 & \text{if } b \text{ is pseudoprime to the base } a, \\ 0 & \text{otherwise.} \end{cases} \tag{44}$$

Theorem 4.1. Let α be a recursive program over **Z** which decides PSP. There is an infinite sequence $\{(a_k, p_k)\}_{k\in\mathbb{N}}$ such that for each k, p_k is pseudoprime to the base a_k and

$$c_\alpha(a_k,p_k) > \frac{1}{5}\log_2 p_k. \tag{45}$$

Proof:
Let $\{p_k\}_{k\in\mathbb{N}}$ be an increasing enumeration of the odd primes. For each k, let $q_k = p_{k+1}$, and let $a_k = 2^{\lfloor \log_2 p_k \rfloor + 1} q_k$. Fix $k \in \mathbb{N}$. Let $(a,p) = (a_k,p_k)$, and $m = c_\alpha(a,p)$. Suppose for a contradiction that $2^{2m+3} \le p$.

Since $\gcd(a,p) = 1$, we have by Fermat's Little Theorem that $a^{p-1} \equiv 1 \pmod p$. Since α decides PSP,

$$\mathbf{Z} \models \alpha^{(m)}(a,p) = 1, \tag{46}$$

and so by Absoluteness,

$$\mathbf{Z} \restriction G_m^{\mathbf{Z}}(a,p) \models \alpha(a,p) = 1. \tag{47}$$

Let ι be given by Lemma 2.4, with $\lambda = p$. Since ι is an imbedding, we obtain

$$\mathbf{Z} \models \alpha(\iota(a), \iota(p)) = 1. \tag{48}$$

Since $p \mid \lambda$, $(\lambda a)^{p-1} \equiv 0 \pmod p$, and so $\neg\mathrm{PSP}(\lambda a, p)$. Therefore

$$\mathbf{Z} \models \alpha(\iota(a), \iota(p)) = 0, \tag{49}$$

a contradiction. Hence $c_\alpha(a,p) > \frac{1}{5}\log_2 p$ as in (40). $\square$

Corollary 4.1. Let α be a recursive program over $\mathbf{Z}$ which computes $f(b,c,m) = b^c \pmod m$. There is an infinite sequence $\{(b_k, c_k, m_k)\}_{k\in\mathbb{N}}$ such that for each k,

$$c_\alpha(b_k, c_k, m_k) > \frac{1}{10}\log_2 m_k. \tag{50}$$

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

$$\zeta_\gamma(\mathsf{x}_0, \mathsf{x}_1) = (\mathsf{if}\ (\zeta_\alpha(\mathsf{x}_0, \mathsf{x}_1 - 1, \mathsf{x}_1) = 1)\ \mathsf{then}\ 1\ \mathsf{else}\ 0). \tag{51}$$

Then γ is a binary recursive program which decides PSP, and the ***Lin***-calls to $\chi_=$ and $-$ in the head contribute 2 to the complexity, so $c_\gamma(x,y) = 2 + c_\alpha(x, y-1, y)$. Let $\{(a_k, p_k)\}_{k\in\mathbb{N}}$ be given by Theorem 4.1, and for each k, let $(b_k, c_k, m_k) = (a_k, p_k - 1, p_k)$. Then

$$c_\alpha(b_k, c_k, m_k) > \frac{1}{5}\log_2 m_k - 2 > \frac{1}{10}\log_2 m_k, \tag{52}$$

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

5. Lower bounds in other models

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

Corollary 5.1. There is a rational constant $k > 0$, such that for infinitely many n, if α is a recursive program which computes the Legendre symbol for $x, y < 2^n$, then

$$\sup\{c_\alpha(x,y) \mid x, y < 2^n\} > kn. \tag{53}$$

Proof:
Let (a,p) be an element of the sequence of Theorem 3.1, and let n be least such that $2a < 2^n$. Let α be a recursive program which computes the Legendre symbol for $x, y < 2^n$. Then α must compute the Legendre symbol for (a,p) and $(\iota(a), \iota(p)) = (2a, p)$, and so, by Theorem 3.1,

$$c_\alpha(a,p) > \frac{1}{20}\log_2 a > \frac{1}{20}(n-2) > \frac{1}{40}n, \tag{54}$$

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

There is an extensive literature on size lower bounds for Boolean circuits which compute number-theoretic functions and predicates, e.g., [1, 3, 5, 6, 18]. Among these results, those which give lower bounds on circuit breadth are apparently difficult to relate to lower bounds on depth like Corollary 5.1, cf. Introduction of [1]. The results which do address depth are also difficult to compare to ours, because the circuit model has different primitive operations and also assumes a different representation of the input.

Turning our attention to uniform models of computation, we first recall a key definition. Let $\mathbf{A} = (A; 0^{\mathbf{A}}, 1^{\mathbf{A}}, \{\phi^{\mathbf{A}}\}_{\phi\in\Phi})$ and $\mathbf{B} = (B; 0^{\mathbf{B}}, 1^{\mathbf{B}}, \{\phi^{\mathbf{B}}\}_{\phi\in\Phi}, \{\psi^{\mathbf{B}}\}_{\psi\in\Psi})$ be partial algebras. **B** is a logical extension (or an inessential extension) of **A** if:

(LOG1) $A \subseteq B$, $0^{\mathbf{A}} = 0^{\mathbf{B}}$, and $1^{\mathbf{A}} = 1^{\mathbf{B}}$,

(LOG2) For n-ary $\phi \in \Phi$,

$$\phi^{\mathbf{B}}(x_1, \ldots, x_n) = \begin{cases} \phi^{\mathbf{A}}(x_1, \ldots, x_n) & \text{if } x_1, \ldots, x_n \in A, \\ \text{undefined} & \text{if some } x_i \notin A. \end{cases} \tag{55}$$

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

$$\rho(\psi^{\mathbf{B}}(x_1, \ldots, x_n)) = \psi^{\mathbf{B}}(\rho(x_1), \ldots, \rho(x_n)). \tag{56}$$

It is shown in [15] that if a lower bound for computing a function in a partial algebra **A** is obtained through the imbedding method, then the same lower bound holds in all logical extensions of **A**, essentially because imbeddings can be lifted to logical extensions using (LOG1)–(LOG3). In [4], RAMs are shown to be represented over certain logical extensions as recursive programs with basic complexity no larger than RAM time complexity.[1] Therefore the lower bounds for (i)–(iii) obtained in this paper hold for the time complexity of RAMs with primitive operations comprised of the functions in ***Lin***.

Acknowledgments. The author wishes to thank his advisor, Yiannis N. Moschovakis, for introducing him to this subject. He is also grateful to the anonymous referees for their useful suggestions.

References

[1] Allender, E., Saks, M., Shparlinski, I.: A lower bound for primality, *J. Comput. System Sci.*, **62**(2), 2001, 356–366, ISSN 0022-0000.

[2] Bach, E., Shallit, J.: *Algorithmic number theory. Vol. 1*, Foundations of Computing Series, MIT Press, Cambridge, MA, 1996, ISBN 0-262-02405-5.

[3] Bernasconi, A., Damm, C., Shparlinski, I.: Circuit and decision tree complexity of some number theoretic problems, *Information and Computation*, **168**(2), 2001, 113–124, ISSN 0890-5401.

[4] van den Dries, L., Moschovakis, Y. N.: Is the Euclidean algorithm optimal among its peers?, *Bulletin of Symbolic Logic*, **10**(3), 2004, 390–418.

[1]In a similar fashion, the author has shown that PCF [16] can be represented in a logical extension with basic complexity smaller than the length of the reduction sequence, and so the lower bound results of the present paper apply to PCF. Full details and related results will appear in a forthcoming communication.

[5] von zur Gathen, J.: Computing powers in parallel, *SIAM J. Comput.*, **16**(5), 1987, 930–945, ISSN 0097-5397.

[6] von zur Gathen, J., Seroussi, G.: Boolean circuits versus arithmetic circuits, *Information and Computation*, **91**(1), 1991, 142–154, ISSN 0890-5401.

[7] Hardy, G. H., Wright, E. M.: *An introduction to the theory of numbers*, Fifth edition, The Clarendon Press Oxford University Press, New York, 1979, ISBN 0-19-853170-2; 0-19-853171-0.

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

[9] Mansoor, Y., Schieber, B., Tiwari, P.: A lower bound for integer greatest common divisor computations, *Journal of the Association for Computing Machinery*, **38**, 1991, 453–471.

[10] Mansoor, Y., Schieber, B., Tiwari, P.: Lower bounds for computations with the floor operation, *SIAM Journal on Computing*, **20**, 1991, 315–327.

[11] McCarthy, J.: A basis for a mathematical theory of computation, in: *Computer programming and formal systems*, North-Holland, Amsterdam, 1963, 33–70.

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

[13] Meyer Eikenberry, S., Sorenson, J. P.: Efficient algorithms for computing the Jacobi symbol, *Journal of Symbolic Computation*, **26**(4), 1998, 509–523, ISSN 0747-7171.

[14] Moschovakis, Y. N.: Arithmetic complexity, 2004, Unpublished notes.

[15] Moschovakis, Y. N.: Recursion and complexity, in: *New Computational Paradigms: First Conference on Computability in Europe, CiE 2005, Amsterdam, The Netherlands, June 8-12, 2005. Proceedings* (S. B. Cooper, B. Löwe, L. Torenvliet, Eds.), vol. 3526/2005, Springer Lecture Notes in Computer Science, 2005, 350–357.

[16] Plotkin, G. D.: LCF considered as a programming language, *Theoret. Comput. Sci.*, **5**(3), 1977/78, 223–255, ISSN 0304-3975.

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

[18] Woods, A. R.: Subset sum "cubes" and the complexity of primality testing, *Theoret. Comput. Sci.*, **322**(1), 2004, 203–219, ISSN 0304-3975.