The axiomatic derivation of absolute lower bounds

Yiannis N. Moschovakis
UCLA and University of Athens

Oberwolfach, 10 November 2011

Algorithms from primitives — the Euclidean algorithm

For $a, b \in \mathbb{N} = \{0, 1, \ldots\}$, $a \geq b \geq 1$,

(ε) $\boxed{\gcd(a, b) = \text{if } (\text{rem}(a, b) = 0) \text{ then } b \text{ else } \gcd(b, \text{rem}(a, b))}$

where $a = \text{iq}(a, b)b + \text{rem}(a, b) \quad (0 \leq \text{rem}(a, b) < b)$

$\text{calls}(\varepsilon, a, b) =$ the number of divisions ε needs to compute $\gcd(a, b)$

$\leq 2\log(b) \qquad (x \geq y \geq 2)$

- Is the Euclidean optimal for computing $\gcd(a, b)$ from rem?
- Is the Euclidean optimal for deciding coprimeness from rem?

$$a \perp\!\!\!\perp b \iff \gcd(a, b) = 1$$

- And is this true among all algorithms from rem, $=_0, =_1$?
- Aim: derive provably robust (with respect to the choice of computation model) and plausibly absolute lower bounds for algorithms which compute a function from specified primitives

Outline

Slogan: *Absolute lower bound results*
are the undecidability facts about decidable problems

(1) Preliminaries
(2) Uniform processes
(3) Comprimeness in $\mathbb{N}$
(4) Polynomial 0-testing

Is the Euclidean algorithm optimal among its peers? (with vDD, 2004)
Arithmetic complexity (with vDD, 2009)

Y. Mansour, B. Schieber, and P. Tiwari (1991)
A lower bound for integer greatest common divisor computations,
Lower bounds for computations with the floor operation
J. Meidânis (1991): *Lower bounds for arithmetic problems*
P. Bürgisser and T. Lickteig (1992) *Verification complexity of linear prime ideals*
P. Bürgisser, T. Lickteig, and M. Shub (1992), *Test complexity of generic polynomials*

(Partial) structures

- A (partial) structure is a tuple $\mathbf{M} = (M, \Phi^{\mathbf{M}})$

  where Φ is a set of function (and relation) symbols

  and $\Phi^{\mathbf{M}} = \{\phi^{\mathbf{M}}\}_{\phi \in \Phi}$, where

  $\boxed{\phi^{\mathbf{M}} : M^{n_\phi} \rightharpoonup M_s}$ i.e., $\boxed{\phi^{\mathbf{M}} : M^{n_\phi} \rightharpoonup M}$ or $\boxed{\phi^{\mathbf{M}} : M^{n_\phi} \rightharpoonup \{\mathrm{tt}, \mathrm{ff}\}}$

- $\mathbf{N}_\varepsilon = (\mathbb{N}, \mathrm{rem}, =_0, =_1)$, the Euclidean structure
- $\mathbf{N}_\varepsilon \restriction U = (U, \mathrm{rem} \restriction U, =_0 \restriction U, =_1 \restriction U)$ where $U \subseteq \mathbb{N}$ and

$$(f \restriction U)(x, y) = w \iff \vec{x} \in U^n, w \in U_s \ \&\ f(\vec{x}) = w$$

- The (equational) diagram of a Φ-structure is the set of its basic equations,

$$\mathrm{eqdiag}(\mathbf{M}) = \{(\phi, \vec{x}, w) : \vec{x}, w \in M \text{ and } \phi^{\mathbf{M}}(\vec{x}) = w\}$$

- We may assume that $\mathbf{M}$ is completely determined by $\mathrm{eqdiag}(\mathbf{M})$

Homomorphisms and substructures

- A homomorphism $\pi : \mathbf{U} \rightarrowtail \mathbf{V}$ is any $\pi : U \to V$ such that for all $\phi \in \Phi, x_1, \ldots, x_n \in U, w \in U_s$, (with $\pi(\mathrm{tt}) = \mathrm{tt}, \pi(\mathrm{ff}) = \mathrm{ff}$)

$$\phi^{\mathbf{U}}(x_1, \ldots, x_n) = w \implies \phi^{\mathbf{M}}(\pi x_1, \ldots, \pi x_n) = \pi w$$

  It is an embedding if it is injective

- Substructures:

$$\begin{aligned} \mathbf{U} \subseteq_p \mathbf{M} &\iff U \subseteq M \\ &\quad \& \ \text{the identity } I : \mathbf{U} \rightarrowtail \mathbf{M} \text{ is an embedding} \\ &\iff U \subseteq M \ \& \ \mathrm{eqdiag}(\mathbf{U}) \subseteq \mathrm{eqdiag}(\mathbf{M}) \end{aligned}$$

- We use finite substructures $\mathbf{U} \subseteq_p \mathbf{M}$ to represent calls to the primitives executed during a computation in $\mathbf{M}$

Algorithms from primitives – the basic intuition

An n-ary algorithm α of $\mathbf{M} = (M, \mathbf{\Phi})$ (or from $\mathbf{\Phi}$) "computes" some n-ary partial function

$$\overline{\alpha} = \overline{\alpha}^{\mathbf{M}} : M^n \rightharpoonup M_s$$

using the primitives in $\mathbf{\Phi}$ as oracles

We understand this to mean that in the course of a "computation" of $\overline{\alpha}(\vec{x})$, the algorithm may request from the oracle for any $\phi^{\mathbf{M}}$ any particular value $\phi^{\mathbf{M}}(\vec{u})$, for arguments $\vec{u}$ which it has already computed, and that if the oracles cooperate, then "the computation" of $\overline{\alpha}(\vec{x})$ is completed in a finite number of "steps"

- The notion of a uniform process attempts to capture minimally (in the style of abstract model theory) these aspects of algorithms from primitives
- It does not capture their effectiveness, but their uniformity —that an algorithm applies "the same procedure" to all arguments in its domain

I The Locality Axiom

A uniform process α *of arity n (and sort s) of a structure* $\mathbf{M} = (M, \Phi^{\mathbf{M}})$ *assigns to each substructure* $\mathbf{U} \subseteq_p \mathbf{M}$ *an n-ary partial function*

$$\overline{\alpha}^{\mathbf{U}} : U^n \rightharpoonup U_s$$

It computes the partial function $\overline{\alpha}^{\mathbf{M}} : M^n \rightharpoonup M_s$

- For an algorithm α, intuitively, $\overline{\alpha}^{\mathbf{U}}$ is the restriction to U of the partial function computed by α when the oracles respond only to questions with answers in eqdiag($\mathbf{U}$)

We write

$$\boxed{\mathbf{U} \vdash \alpha(\vec{x}) = w \iff \vec{x} \in U^n, w \in U_s \text{ and } \overline{\alpha}^{\mathbf{U}}(\vec{x}) = w}$$

II The Homomorphism Axiom

If α is an n-ary uniform process of $\mathbf{M}$, $\mathbf{U}, \mathbf{V} \subseteq_p \mathbf{M}$, *and* $\pi : \mathbf{U} \to \mathbf{V}$ *is a homomorphism, then*

$$\mathbf{U} \vdash \alpha(\vec{x}) = w \implies \mathbf{V} \vdash \alpha(\pi\vec{x}) = \pi w \quad (x_1, \ldots, x_n \in U, w \in U_s)$$

In particular, if $\mathbf{U} \subseteq_p \mathbf{M}$, then $\overline{\alpha}^{\mathbf{U}} \sqsubseteq \overline{\alpha}^{\mathbf{M}}$

- For algorithms: when asked for $\phi^{\mathbf{U}}(\vec{x})$, the oracle for ϕ may consistently provide $\phi^{\mathbf{V}}(\pi\vec{x})$, if π is a homomorphism
- This is obvious for the identity embedding $I : \mathbf{U} \rightarrowtail \mathbf{M}$, but it is a strong restriction for algorithms from rich primitives (stacks, higher type constructs, etc.)
  It can be verified for the standard (deterministic and non-deterministic) computation models

III The Finiteness Axiom

If α is an n-ary uniform process of $\mathbf{M}$*, then*

$\mathbf{M} \vdash \alpha(\vec{x}) = w$

$\Longrightarrow$ *there is a finite* $\mathbf{U} \subseteq_p \mathbf{M}$ *generated by* $\vec{x}$ *such that* $\mathbf{U} \vdash \alpha(\vec{x}) = w$

- For every call $\phi(\vec{u})$ to the primitives, the algorithm must construct the arguments $\vec{u}$, and so the entire computation takes place within a finite substructure generated by the input $\vec{x}$

We write

$$\boxed{\mathbf{U} \vdash_c \alpha(\vec{x}) = w \iff \mathbf{U} \text{ is finite, generated by } \vec{x} \text{ and } \mathbf{U} \vdash \alpha(\vec{x}) = w},$$

and we think of $(\mathbf{U}, \vec{x}, w)$ as a computation of α on the input $\vec{x}$

Complexity measures for uniform processes

Suppose α is an n-ary u.p. of $\mathbf{M}$, $\Phi_0 \subseteq \Phi$, $\mathbf{M} \vdash \alpha(\vec{x}) = w$, and μ is a substructure norm in $\mathbf{M}$. Set:

- $\boxed{C_\mu(\alpha, \vec{x}) = \min\{\mu(\mathbf{U}, \vec{x}) : \mathbf{U} \vdash_c \alpha(\vec{x}) = w\}}$
- $\mathrm{calls}_{\Phi_0}(\alpha, \vec{x}) = \min\{|\mathrm{eqdiag}(\mathbf{U} \restriction \Phi_0)| : \mathbf{U} \vdash_c \alpha(\vec{x}) = w\}$
  (the least number of calls to $\phi \in \Phi_0$ α must do to compute $\overline{\alpha}^{\mathbf{M}}(\vec{x})$)
- $\mathrm{size}_\alpha(\vec{x}) = \min\{|U| : \mathbf{U} \vdash_c \alpha(\vec{x}) = w\}$
  (the least number of elements of $\mathbf{M}$ that α must see)
- $\mathrm{depth}_\alpha(\vec{x}) = \min\{\mathrm{depth}_{\vec{x}}(\mathbf{U}) : \mathbf{U} \vdash_c \alpha(\vec{x}) = w\}$
  (the least number of calls α must execute in sequence)

Thm $\boxed{\mathrm{depth}(\alpha, \vec{x}) \leq \mathrm{size}(\alpha, \vec{x}) \leq \mathrm{calls}(\alpha, \vec{x}) \quad (= \mathrm{calls}_\Phi(\alpha, \vec{x}))}$

These notions agree with standard definitions for concrete algorithms

⋆ The forcing and certification relations

Suppose $f : M^n \to M_s$, $f(\vec{x})\downarrow$, $\mathbf{U} \subseteq_p \mathbf{M}$.

- A homomorphism $\pi : \mathbf{U} \to \mathbf{M}$ respects f at $\vec{x}$ if

$$\vec{x} \in U \ \& \ f(\vec{x}) \in U_s \ \& \ \pi(f(\vec{x})) = f(\pi(\vec{x}))$$

$\mathbf{U} \Vdash^{\mathbf{M}} f(\vec{x}) = w \iff$ every homomorphism $\pi : \mathbf{U} \to \mathbf{M}$ respects f at $\vec{x}$

$\mathbf{U} \Vdash^{\mathbf{M}}_c f(\vec{x}) = w \iff \mathbf{U}$ is finite, generated by $\vec{x}$ and $\mathbf{U} \Vdash^{\mathbf{M}} f(\vec{x}) = w$

The intrinsic complexities of f in $\mathbf{M}$

- $C_\mu(\mathbf{M}, f, \vec{x}) = \min\{\mu(\mathbf{U}, \vec{x}) : \mathbf{U} \Vdash_c f(\vec{x}) = w\}$
- $\mathrm{calls}_{\Phi_0}(\mathbf{M}, f, \vec{x}) = \min\{|\mathrm{eqdiag}(\mathbf{U} \restriction \Phi_0)| : \mathbf{U} \Vdash^{\mathbf{M}}_c \alpha(\vec{x}) = w\}$
- $\mathrm{size}(\mathbf{M}, f, \vec{x}) = \min\{|U| : \mathbf{U} \Vdash^{\mathbf{M}}_c \alpha(\vec{x}) = w\}$
- $\mathrm{depth}(\mathbf{M}, f, \vec{x}) = \min\{\mathrm{depth}_{\vec{x}}(\mathbf{U}) : \mathbf{U} \Vdash^{\mathbf{M}}_c \alpha(\vec{x}) = w\}$

The best uniform process for $f : M^n \rightharpoonup M_s$ in $\mathbf{M}$

Define $\boldsymbol{\beta}_{f,\mathbf{M}}$ by

$$\overline{\boldsymbol{\beta}}^{\mathbf{U}}_{f,\mathbf{M}}(\vec{x}) = w \iff \mathbf{U} \Vdash^{\mathbf{M}} f(\vec{x}) = w \quad (\mathbf{U} \subseteq_p \mathbf{M})$$

Theorem

The following are equivalent for a Φ-structure $\mathbf{M}$ and $f : M^n \rightharpoonup M_s$:

(i) *Some uniform process α of $\mathbf{M}$ computes f.*

(ii) $(\forall \vec{x}, w)\Big(f(\vec{x}) = w \implies (\exists \mathbf{U} \subseteq_p \mathbf{M})[\mathbf{U} \Vdash^{\mathbf{M}}_c f(\vec{x}) = w]\Big)$.

(iii) *$\boldsymbol{\beta}_{f,\mathbf{M}}$ is a uniform process of $\mathbf{M}$ which computes f.*

Moreover, if these conditions hold, then for every uniform process α which computes f in $\mathbf{M}$ and all complexity measures C_μ as above,

$$C_\mu(\mathbf{M}, f, \vec{x}) = C_\mu(\boldsymbol{\beta}_{f,\mathbf{M}}, \vec{x}) \leq C_\mu(\alpha, \vec{x}) \qquad (f(\vec{x})\downarrow).$$

The Homomorphism Test

Lemma

Suppose μ *is a substructure norm* $(\mathsf{calls}_{\Phi_0}, \mathsf{size}, \mathsf{depth})$ *on a* Φ*-structure* $\mathbf{M}$, $f : M^n \rightharpoonup M_s$, $f(\vec{x})\downarrow$, *and*

for every finite $\mathbf{U} \subseteq_p \mathbf{M}$ *which is generated by* $\vec{x}$,

$$\Big(f(\vec{x}) \in U_s \ \&\ \mu(\mathbf{U}, \vec{x}) < m\Big) \implies (\exists \pi : \mathbf{U} \to \mathbf{M})[f(\pi(\vec{x})) \neq \pi(f(\vec{x}))];$$

then $C_\mu(\mathbf{M}, f, \vec{x}) \geq m$.

A lower bound for coprimeness on $\mathbb{N}$

Let $\mathbf{M} = (\mathbf{N}_\varepsilon, \mathbf{\Psi})$ with $\mathbf{\Psi}$ any finite set of *Presburger functions*

Theorem (van den Dries, ynm, 2004, 2009)

If $\xi > 1$ is quadratic irrational, then for some $r > 0$ and all sufficiently large coprime (a, b),

$$\left|\xi - \frac{a}{b}\right| < \frac{1}{b^2} \implies \mathrm{depth}(\mathbf{M}, \perp\!\!\!\perp, a, b) \geq r \log\log a. \qquad (1)$$

In fact, the conclusion of (1) holds with some r

- *for all positive solutions (a, b) of Pell's equation $a^2 = 2b^2 + 1$, and*
- *for all successive Fibonacci pairs (F_{k+1}, F_k). with $k \geq 3$.*

Theorem (Pratt, unpublished)

There is a non-deterministic algorithm ε_{nd} of $\mathbf{N}_\varepsilon$ which decides coprimeness, is at least as effective as the Euclidean everywhere and

$$\mathrm{calls}(\varepsilon_{nd}, F_{k+1}, F_k) \leq K \log\log F_{k+1}$$

- The theorem is best possible from its hypotheses

Non-uniform complexity

Given N, how good can a coprimeness algorithm be if we only insist that it works for n-bit numbers?

$\mathbf{M} = (\mathbf{N}_\varepsilon, \mathbf{\Psi})$ as before. For any N, and any one of the intrinsic complexities as above, let

$$C_\mu(\mathbf{M}, f, N) = \max\{ C_\mu(\mathbf{M} \restriction [0, 2^N), f, a, b) : a, b < 2^N \}$$

Theorem (van den Dries, ynm 2009)

For some rational number $r > 0$ and all sufficiently large N,

$$\mathrm{calls}(\mathbf{M}, \perp\!\!\!\perp, 2^N) \geq \mathrm{size}(\mathbf{M}, \perp\!\!\!\perp, 2^N) \geq r \log N.$$

- Non-uniform lower bound for $\mathrm{depth}(\mathbf{M}, \perp\!\!\!\perp, 2^N)$?

Horner's rule

For any field F and $n \geq 1$, the value of an n'th degree polynomial can be computed using no more than n multiplications and n additions in F:

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = a_0 + x\left(a_1 + a_2 x + \cdots + a_n x^{n-1}\right)$$

Divisions might help:

$$1 + x + x^2 + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1}$$

Theorem (Pan 1966, (Winograd 1967, 1970))

Every straight line algorithm from the real field operations requires at least n multiplications/divisions and at least n additions/subtractions to compute $a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$, *when* $\vec{a}, x$ *are algebraically independent real numbers*

The optimality of Horner's rule for polynomial 0-testing

The nullity relation on a field F:

$$N_F(a_0, \ldots, a_n, x) \iff a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = 0$$

Theorem

Let $\mathbf{R} = (\mathbb{R}, 0, 1, +, -, \cdot, \div, =)$. *If* $n \geq 1$ *and* $a_0, \ldots, a_n, x$ *are algebraically independent, then:*

(1) $\text{calls}_{\{\cdot, \div\}}(\mathbf{R}, N_{\mathbb{R}}, \vec{a}, x) = n$

(2) $\text{calls}_{\{+,-\}}(\mathbf{R}, N_{\mathbb{R}}, \vec{a}, x) = n - 1$

(3) $\text{calls}_{\{+,-,=\}}(\mathbf{R}, N_{\mathbb{R}}, \vec{a}, x) = n + 1$

For algebraic decision trees, (1) is due to Bürgisser and Lickteig (1992), and a result equivalent to (3) is due to Bürgisser, Lickteig and Shub (1992)

The lemma for $\mathrm{calls}_{\{+,-\}}(\mathbf{R}, N_{\mathbb{R}}, \vec{a}, x) = n - 1$

$\mathrm{Roots}(a_1, \ldots, a_n) = \{a_i^{\frac{1}{m}} : m > 0, i = 1, \ldots, n\} \qquad (a_1, \ldots, a_n > 0)$

An operation $u \circ v$ is trivial if $u, v \in \mathbb{K}(x, z)$

Lemma

Suppose $n \geq 2$, $\overline{g} \in \mathbb{K}$ (= real algebraic numbers), $\overline{g} \neq 0$, $z, a_1, \ldots, a_n, x$ are positive, algebraically independent real numbers, and $\mathbf{U}$ is a finite substructure of $\mathbf{R}$ generated by

$$(U \cap \mathbb{K}) \cup \{x, z\} \cup (U \cap \mathrm{Roots}(a_1, \ldots, a_n))$$

which has $< (n-1)$ non-trivial additions and subtractions. Then there is a field homomorphism $\pi : \mathbb{K}(x, z, \mathrm{Roots}(\vec{a})) \to \mathbb{K}(x, \mathrm{Roots}(\vec{a}))$ such that

(a) $\pi(u) = u$ *for every* $u \in \mathbb{K}(x)$,

(b) π *is totally defined and injective on* U, *and*

(c) $\pi(z) + \overline{g}\left(\pi(a_1)x^1 + \cdots + \pi(a_n)x^n\right) = 0.$