The axiomatic derivation of absolute lower bounds

Yiannis N. Moschovakis
UCLA and University of Athens

LICS, 27 June, 2008

The Euclidean algorithm

For $x, y \in \mathbb{N} = \{0, 1, \ldots\},\ x \geq y \geq 1,$

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

where $\text{rem}(x, y)$ is the remainder of the division of x by y,

$$x = \text{iq}(x, y) \cdot y + \text{rem}(x, y) \quad (0 \leq \text{rem}(x, y) < y)$$

$c_\varepsilon(x, y) =$ the number of divisions required to compute $\gcd(x, y)$ by the Euclidean algorithm

$\leq 2 \log(y) \quad (x \geq y \geq 2)$

- $\boxed{\text{Is the Euclidean optimal for computing } \gcd(x, y) \text{ from rem?}}$
- $\boxed{\text{Is the Euclidean optimal for deciding coprimeness from rem?}}$

$$x \perp\!\!\!\perp y \iff \gcd(x, y) = 1$$

A partial result

Theorem (van den Dries, ynm)

If an algorithm α decides the coprimeness relation $x \perp\!\!\!\perp y$ on $\mathbb{N}$ from the primitives $\leq, +, \dot{-}, \mathsf{iq}, \mathsf{rem}$, then for infinitely many $a > b$

$$c_\alpha(a, b) > \frac{1}{10} \log\log a \qquad (*)$$

where $c_\alpha(x, y)$ is the minimum number of applications of the primitives which must be executed in sequence in the computation

- $c_\alpha(x, y)$ is a depth (parallel time) complexity
- The result is one log short of establishing the optimality of the Euclidean (and one log $= \infty$ in this context)
- (*) holds for all sufficiently large a, b such that
  - $a^2 = 1 + 2b^2$ (solutions of Pell's equation),
  - or $a = F_{n+1}, b = F_n$ (successive Fibonacci numbers)
- Claim: This applies to all algorithms from the specified primitives

Outline

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

. . . and so they should be (to some extent) a matter of logic

(1) Tweak logic (a bit) so it applies smoothly to computation theory
(2) Three (simple) axioms for elementary algorithms,
a la *abstract model theory*
(3) Lower bounds from these axioms
(4) Lower bounds from stronger hypotheses about algorithms

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

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*

(Partial) algebras and induced subalgebras

- A (partial, pointed) algebra is a structure $\mathbf{M} = (M, 0, 1, \Phi^{\mathbf{M}})$

  where $0, 1 \in M$, Φ is a set of function symbols (the vocabulary)

  and $\Phi^{\mathbf{M}} = \{\phi^{\mathbf{M}}\}_{\phi \in \Phi}$, with $\phi^{\mathbf{M}} : M^{n_\phi} \rightharpoonup M$ for each $\phi \in \Phi$

$\mathbf{N}_\varepsilon = (\mathbb{N}, 0, 1, \mathrm{rem})$, the Euclidean algebra
$\mathbf{N}_u = (\mathbb{N}, 0, 1, S, \mathrm{Pd})$, the *unary numbers*
$\mathbf{N}_b = (\mathbb{N}, 0, 1, \mathrm{Parity}, \mathrm{iq}_2, (x \mapsto 2x), (x \mapsto 2x+1))$, the *binary numbers*
$\mathbf{N} = (\mathbb{N}, 0, 1, +, \dot{-}, \mathrm{iq}, \mathrm{rem}, \cdot)$, the full algebra of numbers

- Restrictions: for $\{0, 1\} \subseteq U \subseteq M$,

$$\mathbf{U} = \mathbf{M} \restriction U = (U, 0, 1, \Phi^{\mathbf{U}}),$$

  where for $\phi \in \Phi$,

$$\phi^{\mathbf{U}}(\vec{x}) = w \iff \vec{x}, w \in U \ \& \ \phi^{\mathbf{M}}(\vec{x}) = w$$

For finite $U \subset \mathbb{N}$, $\mathbf{N}_b \restriction U$ is a finite, properly partial subalgebra of $\mathbf{N}_b$

Subalgebras and embeddings

- An embedding $\iota : \mathbf{U} \rightarrowtail \mathbf{M}$ from one Φ-algebra into another is any injection $\iota : U \rightarrowtail M$ such that

$$\iota(0^{\mathbf{U}}) = 0^{\mathbf{M}}, \quad \iota(1^{\mathbf{U}}) = 1^{\mathbf{M}},$$

and for all $\phi \in \Phi, x_1, \ldots, x_n, w \in U$,

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

- $\mathbf{U} \subseteq_p \mathbf{M}$ if the identity $I : U \rightarrowtail M$ is an embedding

(so that for every $\mathbf{M}$, $\mathbf{M} \restriction \{0, 1\} \subseteq_p \mathbf{M}$)

Subalgebras generated from the input, $\mathbf{M} = (M, 0, 1, \Phi^{\mathbf{M}})$

For $\vec{x} = x_1, \ldots, x_n \in M$, set

$$G_0(\vec{x}) = \{0, 1, x_1, \ldots, x_n\}$$
$$G_{m+1}(\vec{x}) = G_m(\vec{x}) \cup \{\phi^{\mathbf{M}}(\vec{u}) \mid \phi \in \Phi, \vec{u} \in G_m(\vec{x}) \text{ and } \phi^{\mathbf{M}}(\vec{u})\downarrow\}$$

so that

$$G_m(\vec{x}) = \{t^{\mathbf{M}}[x_1, \ldots, x_n] \in M \mid t(v_1, \ldots, v_n) \text{ is a term of depth} \leq m\}$$

$\boxed{\mathbf{M} \restriction G_m(\vec{x}) \text{ is the subalgebra of depth } m \text{ generated by } \vec{x}}$

$(\mathbf{M} \restriction \bigcup_m G_m(\vec{x})$ is the subalgebra generated by $\vec{x})$

The complexity of values

- Basic principle: *If an algorithm α computes $f : M^n \to M$ from the primitives of $\mathbf{M} = (M, 0, 1, \Phi^{\mathbf{M}})$, then*

$$\text{if } f(\vec{a}) \notin G_m(\vec{a}), \text{ then } c_\alpha(\vec{a}) \geq m$$

- *No hidden primitives* — growth arguments

$G_m(a)$ in $\mathbf{N}_b$: $0 \;\; 1 \;\; 2 \;\; \cdots \;\; 2^{m+1}-1$ | gap | $\mathrm{iq}(a, 2^m) \cdots \; a \; \cdots$

Theorem (van den Dries)

If an algorithm α computes $\gcd(x, y)$ *from*

$$+, \quad \dot{-}, \quad <, \quad =, \quad \mathrm{iq}, \quad \mathrm{rem}, \quad \cdot,$$

then for all $a > b$ such that $a^2 = 1 + 2b^2$ (Pell pairs)

$$c_\alpha(a+1, b) \geq \frac{1}{4}\sqrt{\log\log b}$$

- *Cannot get lower bounds for decision problems by this method*

I The Locality Axiom

An algorithm α of arity n of an algebra $\mathbf{M} = (M, 0, 1, \Phi^{\mathbf{M}})$ assigns to each subalgebra $\mathbf{U} \subseteq_p \mathbf{M}$ an n-ary (strict) partial function

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

- **M**-algorithms "compute" partial functions, and they can be localized (relativized) to arbitrary subalgebras of **M**

We write

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

II The Embedding Axiom

If α is an n-ary algorithm of $\mathbf{M}$, $\mathbf{U}, \mathbf{V} \subseteq_p \mathbf{M}$, *and* $\iota : \mathbf{U} \rightarrowtail \mathbf{V}$ *is an embedding, then*

$$\mathbf{U} \models \overline{\alpha}(\vec{x}) = w \implies \mathbf{V} \models \overline{\alpha}(\iota\vec{x}) = \iota w \quad (x_1, \ldots, x_n, w \in U)$$

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

- An algorithm treats the primitives of $\mathbf{M}$ as *oracles*: it can request values $\phi^{\mathbf{M}}(\vec{y})$, and use them if they are provided

III The Finiteness Axiom

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

$$\mathbf{M} \models \overline{\alpha}(\vec{x}) = w \implies \textit{there is an } m \textit{ such that } \vec{x}, w \in G_m(\vec{x}) \text{ and } \mathbf{M} \restriction G_m(\vec{x}) \models \overline{\alpha}(\vec{x}) = w$$

In particular,

$$\overline{\alpha}^{\mathbf{M}}(\vec{x})\downarrow \implies \overline{\alpha}(\vec{x}) \in \bigcup_m G_m(\vec{x})$$

- "The computation" of $\overline{\alpha}^{\mathbf{M}}(\vec{x})$ takes place within the subalgebra of $\mathbf{M}$ generated by the input, and it is finite: take m large enough so that every y "used in the computation" is in $G_m(\vec{x})$

All elementary algorithms satisfy these axioms

- Explicit computation: $\overline{\alpha}^{\mathbf{U}}(\vec{x}) = t^{\mathbf{U}}[\vec{x}]$, where $t(\vec{v})$ is a Φ-term
- $\overline{\alpha}^{\mathbf{U}}$ is the partial function computed a fixed recursive (McCarthy) program A in the signature Φ
- $\overline{\alpha}^{\mathbf{U}}$ is computed from $\Phi^{\mathbf{U}}$ by any of the familiar machine models of computation—register machines, Random Access Machines, Turing machines, etc.
- $\overline{\alpha}^{\mathbf{U}}$ is computed in PCF above the algebra $\mathbf{U}$
- $\overline{\alpha}^{\mathbf{U}}$ by computed by non-deterministic versions of any of these

Axioms for elementary algorithms

- I, Locality Axiom: An algorithm α of arity n of an algebra $\mathbf{M} = (M, 0, 1, \Phi^{\mathbf{M}})$ assigns to each subalgebra $\mathbf{U} \subseteq_p \mathbf{M}$ an n-ary partial function

$$\overline{\alpha}^{\mathbf{U}} : U^n \rightharpoonup U \qquad (\mathbf{U} \models \overline{\alpha}(\vec{x}) = w \iff \overline{\alpha}^{\mathbf{U}}(\vec{x}) = w)$$

- II, Embedding Axiom: If $\mathbf{U}, \mathbf{V} \subseteq_p \mathbf{M}$, and $\iota : \mathbf{U} \rightarrowtail \mathbf{V}$ is an embedding, then

$$\mathbf{U} \models \overline{\alpha}(\vec{x}) = w \implies \mathbf{V} \models \overline{\alpha}(\iota\vec{x}) = \iota w \quad (x_1, \ldots, x_n, w \in U)$$

- III, Finiteness Axiom:

$$\mathbf{M} \models \overline{\alpha}(\vec{x}) = w \implies \textit{there is an } m \textit{ such that } \vec{x}, w \in G_m(\vec{x}) \text{ and } \mathbf{M} \restriction G_m(\vec{x}) \models \overline{\alpha}(\vec{x}) = w$$

The embedding complexity of an algorithm

If α is an algorithm of $\mathbf{M}$ and $\mathbf{M} \models \overline{\alpha}(\vec{x}) = w$, set

$$\boxed{c^{\iota}_{\alpha}(\vec{x}) = \text{the least } m \text{ such that } \mathbf{M} \restriction G_m(\vec{x}) \models \overline{\alpha}(\vec{x}) = w}$$

This is defined by the Finiteness Axiom

- Intuitively, if $m = c^{\iota}_{\alpha}(\vec{x})$, then any implementation of α will need to "consider" (use) some $u \in M$ of depth m above $\vec{x}$; and so it will need at least m steps to construct this u from the input using the primitives
- If $\overline{\alpha}(\vec{x}) = t^{\mathbf{M}}[\vec{x}]$, then $c^{\iota}_{\alpha}(\vec{x}) \leq \text{depth}(t(\vec{v}))$
- c^{ι}_{α} is majorized in the examples by all usual time-complexity measures, including

$c_{\alpha}(x, y) =$ *the minimum number of applications of the primitives which must be executed in sequence in the computation*

⋆ The embedding complexity of a (computable) function

Fix $f : M^n \to M$. An embedding $\iota : \mathbf{M} \restriction G_m(\vec{x}) \rightarrowtail \mathbf{M}$ respects f at $\vec{x}$ if

$$f(\vec{x}) \in G_m(\vec{x}) \ \& \ \iota(f(\vec{x})) = f(\iota(\vec{x}))$$

Lemma

If some algorithm computes f in $\mathbf{M}$, then for each $\vec{x}$, there is some m such that every embedding $\iota : \mathbf{M} \restriction G_m(\vec{x}) \rightarrowtail \mathbf{M}$ respects f at $\vec{x}$

Proof Take $m = c^{\iota}_{\alpha}(\vec{x})$ for some α such that $f = \overline{\alpha}^{\mathbf{M}}$

$$\boxed{c^{\iota}_{f}(\vec{x}) = \text{the least } m \text{ such that every } \iota : \mathbf{M} \restriction G_m(\vec{x}) \rightarrowtail \mathbf{M} \text{ respects } f \text{ at } \vec{x}}$$

If α computes f in $\mathbf{M}$, then $c^{\iota}_{f}(\vec{x}) \leq c^{\iota}_{\alpha}(\vec{x})$

- To show that m is an absolute lower bound for the computation of the value $f(\vec{x})$, show that $\boxed{f(\vec{x}) \notin G_m(\vec{x})}$ or

  construct $\iota : \mathbf{M} \restriction G_m(\vec{x}) \rightarrowtail \mathbf{M}$ such that $\boxed{\iota f(\vec{x}) \neq f(\iota \vec{x})}$

Outline of a proof

Theorem (van den Dries, ynm)

In $\mathbf{M} = (\mathbb{N}, 0, 1, \leq, +, \dot{-}, \mathrm{iq}, \mathrm{rem})$*: if* a *is sufficiently large and* $a^2 = 1 + 2b^2$ *or* $a = F_{n+1}, b = F_n$*, then*

$$c^{\iota}_{\perp\!\!\!\perp}(a, b) > \frac{1}{10} \log\log(a) \qquad (*)$$

So if α *decides coprimeness in* $\mathbf{M}$*, then* (*) *holds with* $c_\alpha(a, b)$

- If $2^{2^{4m+6}} \leq a$, then every $X \in G_m(a, b)$ can be written uniquely as

$$X = \frac{x_0 + x_1 a + x_2 b}{x_3} \qquad \text{with } x_i \in \mathbb{Z}, \quad |x_i| \leq 2^{2^{4m}}$$

and we can define $\iota : \mathbf{M} \restriction G_m(a, b) \rightarrowtail \mathbf{M}$ using $\lambda = 1 + a!$,

$$\iota(X) = \frac{x_0 + x_1 \lambda a + x_2 \lambda b}{x_3}, \text{ so } (\iota(a), \iota(b)) = (\lambda a, \lambda b)$$

The "universal constant" $\frac{1}{10}$

Detailed version of result

In $\mathbf{M} = (\mathbb{N}, 0, 1, \leq, +, \dot{-}, \mathrm{iq}, \mathrm{rem})$*: if* $1 < \xi < 2$*,* ξ *is a quadratic algebraic irrational,* $C > 0$*,* a *is sufficiently large, and* $a \perp\!\!\!\perp b$*, then*

$$\frac{1}{Cb^2} < \left|\xi - \frac{a}{b}\right| < \frac{1}{b^2} \implies c^{\iota}_{\perp\!\!\!\perp}(a, b) > \frac{1}{K} \log\log a,$$

with $K \geq 2\log(\log C + 19)$

- Liouville: for some C, infinitely many a, b satisfy the hypothesis
- With $\xi = \sqrt{2}$ and $a^2 = 1 + 2b^2$, we can take $C = 5, K \geq 10$
- With $\xi = \frac{1}{2}(1 + \sqrt{5})$ and $a = F_{n+1}, b = F_n$, again, $C = 5, K \geq 10$

$\mathbf{M} = (\mathbb{N}, 0, 1, \text{Parity}, \text{iq}_2, \leq, +, \dot{-}, \text{Presburger functions})$

- (van den Dries, ynm) *If $R(x)$ is one of the relations*

  $$\boxed{x \text{ is prime}}, \quad \boxed{x \text{ is a perfect square}}, \quad \boxed{x \text{ is square free}},$$

  then for some $r > 0$ and infinitely many a, $c_R^{\iota}(a) > r \log(a)$

- (van den Dries, ynm) *For some $r > 0$ and infinitely many a, b,*

  $$c_{\perp\!\!\!\perp}^{\iota}(a, b) > r \log(\max(a, b))$$

- (Joe Busch) *If $R(x, p) \iff \boxed{x \text{ is a square } \bmod p}$,*
  then for some $r > 0$ and a sequence (a_n, p_n) with $p_n \to \infty$,

  $$c_R^{\iota}(a_n, p_n) > r \log(p_n)$$

In the last two examples, the results match up to a multiplicative constant the known binary algorithms, so these are optimal

Primality in binary

- If $\mathrm{Prime}(p) \iff p$ is prime, then in

$$\mathbf{N}_b = (\mathbb{N}, 0, 1, \mathrm{Parity}, \mathrm{iq}_2, (x \mapsto 2x), (x \mapsto 2x+1)),$$

for all sufficiently large primes p,

$$c^{\iota}_{\mathrm{Prime}}(p) \geq \frac{1}{6} \log p \qquad (*)$$

- This should follow trivially from number-theoretic results, because it takes at least i applications of the primitives of $\mathbf{N}_b$ to read i bits of the input; we should have 1 in the place of $\frac{1}{6}$
- Theorem (Tao). *For infinitely many primes p, if p' is constructed by changing any bit in the binary expansion of p except the highest, then p' is not prime*
- Tao found subsequently that this result is implicit in a paper of Cohen and Selfridge from 1975 and explicitly noted in a 2000 paper by Sun, and he obtained more general results

Non-uniform complexity

What if you are only interested in deciding $R(\vec{x})$ for n-bit numbers ($< 2^n$) and you are willing to use a different algorithm for each n?

- The lookup algorithm: For each k-ary relation R on $\mathbb{N}$ and each n, there is an $\mathbf{N}_b$-term (with conditionals) $t_n(\vec{v})$ of depth $\leq n = \log(2^n)$ which decides $R(\vec{x})$ for all $\vec{x} < 2^n$.

- Non-uniform lower bounds on depth are never greater than log
- The best ones establish the optimality of the lookup algorithm (and are most interesting when some uniform algorithm matches the lookup up to a multiplicative constant)
- They are mostly about "the size" of $t(\vec{v})$
- They do not follow from Axiom I – III

Recursive (McCarthy) programs of $\mathbf{M} = (M, 0, 1, \Phi^{\mathbf{M}})$

Explicit Φ-terms (with p_i^n partial function variables)

$$A :\equiv 0 \mid 1 \mid \mathrm{v}_i \mid \phi(A_1, \ldots, A_n) \mid \mathrm{p}_i^n(A_1, \ldots, A_n) \mid \text{if } (A_0 = 0) \text{ then } A_1 \text{ else } A_2,$$

Recursive program (only $\vec{x}_i, p_1, \ldots, p_K$ occur in each part A_i):

$$A \quad : \quad \begin{cases} \mathrm{p}_A(\vec{x}_0) = A_0 \\ \mathrm{p}_1(\vec{x}_1) = A_1 \\ \qquad \vdots \\ \mathrm{p}_K(\vec{x}_K) = A_K \end{cases} \qquad (A_0 : \text{ the head}, (A_1, \ldots, A_K) : \text{ the body})$$

The elementary algorithms of $\mathbf{M}$ *are expressed by recursive programs*

They satisfy Axioms I – III, but also have a rich structure

A non-uniform lower bound result for elementary algorithms

If α is the algorithm expressed by a recursive program in **M**, let

$$c^s_\alpha(\vec{x}) = \text{the number of calls to the primitives made in the computation of } \overline{\alpha}(\vec{x}) \ \geq c^\iota_\alpha(\vec{x})$$

Theorem (van den Dries, ynm)

Let $\mathbf{M} = (\mathbb{N}, 0, 1, \leq, +, \dot{-}, \mathrm{iq}, \mathrm{rem})$. *There is some* $r > 0$, *such that for all sufficiently large* n *and every* **M***-elementary algorithm* α *which decides coprimeness for all* $x, y < 2^n$, *there exist* $a, b < 2^n$ *such that*

$$c^s_\alpha(a, b) > r \log n \geq r \log \log(\max(a, b))$$

The proof is by the embedding method, but uses special properties of recursive programs (the computation space)

Remarks

(1) A technique for deriving lower bounds for decision problems which are absolute, i.e., they hold of all computational models

(2) Main limitation: in its current version, it only yields lower bounds which are no better than $O(n)$ (linear in the length of the input)

(3) *Problem*: prove that the Euclidean algorithm is optimal for computing the gcd in the algebra $\mathbf{N}_\varepsilon = (\mathbb{N}, 0, 1, \mathrm{rem})$

(4) *Problem*: prove an $O(n^2)$ lower bound for *primality* in $\mathbf{N}_b = (\mathbb{N}, 0, 1, \mathrm{Parity}, \mathrm{iq}_2, (x \mapsto 2x), (x \mapsto 2x+1))$

Comment: (4) may need some number theory, but it will also need some logical analysis of computation (since the entire input is known in $O(n)$ steps)