Recursion and complexity

(Relative complexity in arithmetic and algebra)

Yiannis N. Moschovakis
UCLA and University of Athens

Logic and interactions 2012, CIRM Marseille
30 - 31 January, 2012

Motivating problem: the Euclidean algorithm

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

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

where $\mathrm{rem}(a, b)$ is the remainder of the division of a by b,

$$a = bq + \mathrm{rem}(a, b) \quad (0 \leq \mathrm{rem}(a, b) < b)$$

$$\begin{aligned}\mathrm{calls}_\varepsilon(a, b) &= \text{the number of divisions required to compute } \mathrm{gcd}(a, b) \\ &\quad \text{by the Euclidean algorithm} \\ &\leq 2\log(b) = 2\log_2(b) \qquad (a \geq b \geq 2)\end{aligned}$$

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

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

- Most relevant complexity: number of required divisions
- Looking for absolute lower bounds, which restrict all all algorithms

Outline

1.1. Recursive (McCarthy) Programs.
Preliminaries and notation. The Church-Turing Thesis.

1.2. Uniform processes (The main dish).
An axiomatic approach to the theory of algorithms from specified primitives in the style of *abstract model theory*.

2.1. Lower bounds in arithmetic (arithmetic complexity).
Robust lower bounds for coprimeness (joint with van den Dries)

2.2. Lower bounds in algebra (algebraic complexity).
Robust lower bounds results for 0-testing of polynomials over fields, extending results of Peter Bürgisser (with others).

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

Full proofs and references posted at http://www.math.ucla.edu/~ynm

Outline

- **1.1. Recursive (McCarthy) Programs**.
  Preliminaries and notation. The Church-Turing Thesis.

  1.2. Uniform processes (The main dish).
  An axiomatic approach to the theory of algorithms from specified primitives in the style of *abstract model theory*.

  2.1. Lower bounds in arithmetic (arithmetic complexity).
  Robust lower bounds for coprimeness (joint with van den Dries)

  2.2. Lower bounds in algebra (algebraic complexity).
  Robust lower bounds results for 0-testing of polynomials over fields, extending results of Peter Bürgisser (with others).

(Partial) structures — sets with "given" primitives

- We identify a relation $R \subseteq A^n$ with its characteristic function

$$R(\vec{x}) = \text{if } R \text{ holds at } \vec{x} \text{ then } \mathrm{tt} \text{ else ff}$$

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

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

  and $\Phi^{\mathbf{A}} = \{\phi^{\mathbf{A}}\}_{\phi \in \Phi}$. Here

  $\boxed{\phi^{\mathbf{A}} : A^{n_\phi} \rightharpoonup A_s}$ with $s = \mathrm{sort}(\phi) = \mathtt{a}$ or $\mathrm{sort}(\phi) = \mathtt{boole}$

  i.e., if $n_\phi = \mathrm{arity}(\phi)$, then

$$\boxed{\phi^{\mathbf{A}} : A^{n_\phi} \rightharpoonup A} \text{ or } \boxed{\phi^{\mathbf{A}} : A^{n_\phi} \rightharpoonup \{\mathrm{tt}, \mathrm{ff}\}}$$

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

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

Equational logic of partial terms with conditionals

For a vocabulary Φ and a set A, the $(\Phi \cup A)$-terms are defined by

$$\boxed{t :\equiv \mathrm{tt} \mid \mathrm{ff} \mid x \mid \mathsf{v}_i \mid \phi(t_1, \ldots, t_{n_\phi}) \mid \text{if } t_1 \text{ then } t_2 \text{ else } t_3}$$

where $\phi \in \Phi$, $x \in A$ (viewed as a constant or parameter) and $\mathsf{v}_0, \mathsf{v}_1, \ldots$ is a fixed sequence of individual variables

- Each term is assigned a sort, `boole` or `a`
- If $t \equiv \text{if } t_1 \text{ then } t_2 \text{ else } t_3$, then $\mathrm{sort}(t_1) \equiv \texttt{boole}$ and $\mathrm{sort}(t_2) \equiv \mathrm{sort}(t_3) \equiv \mathrm{sort}(t)$
- t is pure : no parameters (a Φ-term)
  t is closed : no variables
- If t is closed and $\mathbf{A}$ is a Φ-structure, then

$$\mathrm{den}(\mathbf{A}, t) = \text{ the value of } t \text{ in } \mathbf{A} \quad (\text{if } t \text{ converges}, t\downarrow)$$

$$\boxed{\mathbf{A} \models t = s \iff \Big(\mathrm{den}(\mathbf{A}, t)\uparrow \ \& \ \mathrm{den}(\mathbf{A}, s)\uparrow\Big) \text{ or } \mathrm{den}(\mathbf{A}, t) = \mathrm{den}(\mathbf{A}, s)}$$

Recursive (McCarthy) programs — syntax

A Φ-recursive program E of arity n is a syntactic expression

$$\boxed{E \equiv E_0(\vec{x}, \vec{p}) \text{ where } \{p_1(\vec{v}_1) = E_1(\vec{v}_1, \vec{p}), \dots, p_k(\vec{v}_k) = E_k(\vec{v}_k, \vec{p})\}}$$

where

(RP1) $\vec{p} \equiv p_1, \dots, p_k$ is a sequence of (not necessarily distinct) fresh function symbols (not in Φ), the recursive variables of E

(RP2) For $i = 0, \dots, k$, $E_i(\vec{v}_i, \vec{p})$ is a (pure) term in the vocabulary $\boxed{\mathrm{voc}(E) = \Phi \cup \{p_1, \dots, p_k\}}$ whose variables are in the list $\vec{v}_i \equiv v_1, \dots, v_{k_i}$ (with $\vec{v}_0 \equiv \vec{x} \equiv x_1, \dots, x_n$)

(RP3) For $i = 1, \dots, k$, $\mathrm{sort}(p_i(\vec{v}_i)) = \mathrm{sort}(E_i(\vec{v}_i, \vec{p}))$

- $\mathrm{sort}(E) = \mathrm{sort}(E_0(\vec{x}, \vec{p}))$
- the *free variables* of E are $x_1, \dots, x_n$
- the *bound variables* of E are those in the lists $\vec{v}_i$ and the *recursive variables* $p_1, \dots, p_k$

- A program is deterministic if its recursive variables are all distinct

Recursive programs — (call-by-value) semantics

$$\boxed{E \equiv E_0(\vec{x}, \vec{p}) \text{ where } \{p_1(\vec{v}_1) = E_1(\vec{v}_1, \vec{p}), \ldots, p_k(\vec{v}_k) = E_k(\vec{v}_k, \vec{p})\}}$$

$$\mathrm{ClTerms}(E, \mathbf{A}) = \text{the set of closed } (\mathrm{voc}(E) \cup A)\text{-terms}$$

Need to define the relation

$$E, \mathbf{A} \vdash t = w \iff t \in \mathrm{ClTerms}(E, \mathbf{A}), w \in A \cup \{\mathrm{tt}, \mathrm{ff}\}$$
$$\& \ w \text{ is one of the values assigned to } t \text{ by } E \text{ in } \mathbf{A}$$

- (1) Least-fixed-point semantics (non-deterministic for all E): $t = w$ belongs to the least set S which contains all $w = w$ and is closed under the natural semantic conditions, e.g.,

$$t_1 = u_1, \ldots, t_{n_\phi} = u_{n_\phi} \in S \ \& \ \phi^{\mathbf{A}}(u_1, \ldots, u_{n_\phi}) = w$$
$$\Longrightarrow \ \phi(t_1, \ldots, t_{n_\phi}) = w \in S$$

- (2) Implementations (many, deterministic if E is deterministic): There is a computation $t \hookrightarrow \boxed{s_1 \to \cdots \to s_m} \looparrowright w$ by E in $\mathbf{A}$ which assigns w to t ($\hookrightarrow$ is input, $\looparrowright$ is output)

The recursive (two stack) machine $\mathcal{P}(E, \mathbf{A})$

(pass)	$\vec{a}\ \underline{x :}\ \vec{b}$	$\to\ \vec{a}\ \underline{: x}\ \vec{b}$	$(x \in A \cup \{\mathrm{tt}, \mathrm{ff}\})$
(e-call)	$\vec{a}\ \underline{\phi : \vec{x}}\ \vec{b}$	$\to\ \vec{a}\ \underline{: \phi^{\mathbf{A}}(\vec{x})}\ \vec{b}$	
(i-call)	$\vec{a}\ \underline{p_i : \vec{u}}\ \vec{b}$	$\to\ \vec{a}\ \underline{E_i(\vec{u}, \vec{p}) :}\ \vec{b}$	
(comp)	$\vec{a}\ \underline{h(t_1, \ldots, t_n) :}\ \vec{b}$	$\to\ \vec{a}\ \underline{h\ t_1\ \cdots\ t_n :}\ \vec{b}$	
(br)	$\vec{a}\ \underline{\mathit{if}\ t_0\ \mathit{then}\ t_1\ \mathit{else}\ t_2 :}\ \vec{b}$	$\to\ \vec{a}\ \underline{t_1\ t_2\ ?\ t_0 :}\ \vec{b}$	
(br0)	$\vec{a}\ \underline{t_1\ t_2\ ? : \mathrm{tt}}\ \vec{b}$	$\to\ \vec{a}\ \underline{t_1 :}\ \vec{b}$	
(br1)	$\vec{a}\ \underline{t_1\ t_2\ ? : \mathrm{ff}}\ \vec{b}$	$\to\ \vec{a}\ \underline{t_2 :}\ \vec{b}$	

- States are sequences of the form $\vec{L} : \vec{R}$, where
  $\vec{L}$ is a tuple from $\mathrm{ClTerms}(E, \mathbf{A}) \cup \mathrm{voc}(E) \cup \{?\}$ and $\vec{R}$ a tuple from $A \cup \{\mathrm{tt}, \mathrm{ff}\}$
- Input $t \hookrightarrow \boxed{t :}$ Terminal states $\boxed{: w}$ Output $\boxed{: w} \looparrowright w$
- The underlined part is the trigger for the transition
- In the *external call* (e-call), $\phi \in \Phi$ and $\mathrm{arity}(\phi) = n_\phi =$ length of $\vec{x}$
- In the *internal call* (i-call), $p_i(\vec{u}) = E_i(\vec{u}, \vec{p})$ is an equation of E

$\mathbf{A}$-recursive functions

$$\boxed{E \equiv E_0(\vec{x}, \vec{p}) \text{ where } \{p_1(\vec{v}_1) = E_1(\vec{v}_1, \vec{p}), \ldots, p_k(\vec{v}_k) = E_k(\vec{v}_k, \vec{p})\}}$$

- A partial function $f : A^n \rightharpoonup A_s$ is *computed by E in* $\mathbf{A}$ if

$$\boxed{f(\vec{x}) = w \iff E, \mathbf{A} \vdash E_0(\vec{x}, \vec{p}) = w \quad (\vec{x} \in A^n, w \in A_s)}$$

- At most one partial function is computed by E in $\mathbf{A}$

- $\boxed{\mathbf{rec}(\mathbf{A}) = \{f : A^n \rightharpoonup A_s : f \text{ is computed in } \mathbf{A} \text{ by a deterministic } E\}}$
- $\mathbf{rec}_{\mathrm{nd}}(\mathbf{A}) = \{f : A^n \rightharpoonup A_s : f$ is computed in $\mathbf{A}$ by some $E\}$
- In general $\mathbf{rec}(\mathbf{A}) \subsetneq \mathbf{rec}_{\mathrm{nd}}(\mathbf{A})$

Recursive programs — complexity measures

$$\boxed{E \equiv E_0(\vec{x}, \vec{p}) \text{ where } \{p_1(\vec{v}_1) = E_1(\vec{v}_1, \vec{p}), \ldots, p_k(\vec{v}_k) = E_k(\vec{v}_k, \vec{p})\}}$$

- $\text{calls}_{\Phi_0}(E, \mathbf{A})(t = w)$ = the least number of calls to the primitives in $\Phi_0 \subseteq \Phi$ that E must execute to prove $t = w$
- $\text{depth}(E, \mathbf{A})(t = w)$ = the least number of calls to the primitives in Φ that E must execute in sequence to prove $t = w$
- $\text{size}(E, \mathbf{A})(t = w)$ = the least number of points in A that E must see to prove $t = w$
- These are defined inductively or for each computation of $t = w$ (and then the least of these numbers is selected)
- If E computes $f : A^n \rightharpoonup A_s$, they give complexity measures

$$\text{calls}_{\Phi_0}(E, \mathbf{A}, \vec{x}),\ \text{depth}(E, \mathbf{A}, \vec{x}),\ \text{size}(E, \mathbf{A}, \vec{x}) \qquad (f(\vec{x})\downarrow)$$

- **Thm**. $\text{depth}(E, \mathbf{A}, \vec{x}) \leq \text{size}(E, \mathbf{A}, \vec{x}) \leq \text{calls}_{\Phi}(E, \mathbf{A}, \vec{x})$

Recursive programs — special forms

$$\boxed{E \equiv E_0(\vec{x}, \vec{p}) \text{ where } \{p_1(\vec{v}_1) = E_1(\vec{v}_1, \vec{p}), \ldots, p_k(\vec{v}_k) = E_k(\vec{v}_k, \vec{p})\}}$$

- Terms: $k = 0$, so $E \equiv E_0(\vec{x})$ is a Φ-term
- Finite algorithms with branching:
  for each $i = 1, \ldots, k$, if p_j occurs in $E_i(\vec{v}_i, \vec{p})$, then $j < i$

  Deterministic finite algorithms with or without conditionals include the standard computation models of algebraic complexity (k-step algorithms, computation sequences, algebraic decision trees, etc; Pan, Winograd, Strassen, Bürgisser)
- Tail recursive (or "while") programs,

  $$E \equiv E_0(\vec{x}, p) \text{ where}$$
  $$\{p(\vec{u}) = \text{if test}(\vec{u}) \text{ then out}(\vec{u}) \text{ else } p(\tau_1(\vec{u}), \ldots, \tau_m(\vec{u}))\}$$

  with Φ-terms $\text{test}(\vec{u}), \text{out}(\vec{u}), \tau_j(\vec{u})$

  The standard models of computation are faithfully represented by tail recursions, once their natural primitives are identified

Abstract recursion — further topics

- The relation between $\mathbf{rec}(\mathbf{A}), \mathbf{rec}_{\mathrm{nd}}(\mathbf{A})$ and the class $\mathbf{tail}(\mathbf{A})$ of tail recursive functions on arbitrary $\mathbf{A}$

  (delicate results, Stolbouskin, Taitslin, Tiuryn, etc.)
- Deterministic and non-deterministic functionals, the First Recursion Theorem, etc.
- The Formal Language of Recursion

  (admits "where" as an unrestricted construct)
- Recursive programs as specifications of algorithms

  The meaning of a program is the algorithm it expresses

  Recursive vs. iterative (tail) algorithms
- Second and higher type recursion
  large field with applications to model theory, set theory, etc.

The Recursive Computability and Church-Turing Theses

- RCT: $f : A^n \rightharpoonup A_s$ *is (recursively) computable from* $\phi_1, \ldots, \phi_m$
  $\iff f \in \mathbf{rec}(A, \phi_1, \ldots, \phi_m)$
  - RCT: The fundamental algorithmic constructs are calling (composition), branching, and grounded self reference
  - The definition of $\mathbf{rec}(\mathbf{A})$ does not involve any objects outside $\mathbf{A}$
  - $\mathbf{A}$-recursion is a Tarski logical notion, preserved by permutations
- *The natural numbers are the structure* $\mathbf{N} = (\mathbb{N}, 0, S, =)$

(Dedekind: up to isomorphism, ..., the modern structuralist view)

•• $f : \mathbb{N} \rightharpoonup \mathbb{N}_s$ *is recursive* $\iff f \in \mathbf{rec}(\mathbb{N}, 0, S, =)$

Thm $f \in \mathbf{rec}(\mathbb{N}, 0, S, =)$ *if and only if* f *is Turing computable*

$\boxed{\text{Turing computability on } \mathbb{N} = \text{recursion} + \text{what the numbers are}}$

Outline

1.1. Recursive (McCarthy) Programs.
Preliminaries and notation. The Church-Turing Thesis.

- **1.2. Uniform processes** (The main dish).
An axiomatic approach to the theory of algorithms from specified primitives in the style of *abstract model theory*.

2.1. Lower bounds in arithmetic (arithmetic complexity).
Robust lower bounds for coprimeness (joint with van den Dries).

2.2. Lower bounds in algebra (algebraic complexity).
Robust lower bounds results for 0-testing of polynomials over fields, extending results of Peter Bürgisser (with others).

More about structures

- Reducts: for every $\Phi_0 \subseteq \Phi$,

$$\mathbf{A} \restriction \Phi_0 = (A, \{\phi^{\mathbf{A}} : \phi \in \Phi_0\})$$

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

$$\mathrm{eqdiag}(\mathbf{A}) = \{(\phi, \vec{x}, w) : \phi \in \Phi, \phi^{\mathbf{A}}(\vec{x}) = w\}$$

- A homomorphism $\pi : \mathbf{U} \to \mathbf{V}$ 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 $\pi(\mathrm{tt}) = \mathrm{tt}, \pi(\mathrm{ff}) = \mathrm{ff}$)

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

- π is an embedding if it is injective
  π is an isomorphism if it is a surjective embedding and the inverse map $\pi^{-1} : V \to U$ is also an embedding
- The homomorphic image $\pi[\mathbf{U}]$ has universe $\pi[U]$ and

$$\mathrm{eqdiag}(\pi[\mathbf{U}]) = \{(\phi, \pi(\vec{x}), \pi(w)) : (\phi, \vec{x}, w) \in \mathrm{eqdiag}(\mathbf{U})\}$$

Substructures, generation

- Substructures:

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

$$\begin{aligned}G_0(\mathbf{U}, \vec{x}) &= \{x_1, \dots, x_n\}, \\ G_{m+1}(\mathbf{U}, \vec{x}) &= G_m(\mathbf{U}, \vec{x}) \cup \{\phi^{\mathbf{U}}(u_1, \dots, u_{n_\phi}) : u_1, \dots, u_{n_\phi} \in G_m(\mathbf{U}, \vec{x})\} \\ \mathbf{G}_m(\mathbf{U}, \vec{x}) &= \mathbf{U} \restriction G_m(\mathbf{U}, \vec{x})\end{aligned}$$

- $\mathbf{U} \subseteq_p \mathbf{A}$ is generated by $\vec{x} \in U^n$ if $U = G_\infty(\mathbf{U}, \vec{x}) = \bigcup_m G_m(\mathbf{U}, \vec{x})$

  $\mathrm{depth}(\mathbf{U}, \vec{x}) = \min\{m : U = G_m(\mathbf{U}, \vec{x})\}$ ($\mathbf{U}$ finite, generated by $\vec{x}$)

- We use finite $\mathbf{U} \subseteq_p \mathbf{A}$ generated by the input $\vec{x}$ to represent calls to the primitives executed during a computation in $\mathbf{A}$

Algorithms from primitives – the basic intuition

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

$$\overline{\alpha} = \overline{\alpha}^{\mathbf{A}} : A^n \rightharpoonup A_s \quad (A_s = A \text{ or } A_s = \{\mathrm{tt}, \mathrm{ff}\})$$

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 $\phi^{\mathbf{A}}$ any particular value $\phi^{\mathbf{A}}(\vec{u})$, 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 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

I The Locality (or relativization) Axiom

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

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

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

- 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

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

- True for a program E which computes some $f = \overline{E} : A^n \rightharpoonup A_s$ in $\mathbf{A}$ by

$$\overline{E}^{\mathbf{U}}(\vec{x}) = w \iff E, \mathbf{U} \vdash E_0(\vec{x}, \vec{p}) = w$$

II The Homomorphism Axiom

If α is an n-ary uniform process of $\mathbf{A}$, $\mathbf{U}, \mathbf{V} \subseteq_p \mathbf{A}$, 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 $\mathbf{U} \subseteq_p \mathbf{A}$, then $\overline{\alpha}^{\mathbf{U}} \sqsubseteq \overline{\alpha}^{\mathbf{A}}$

- 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 $I : \mathbf{U} \rightarrowtail \mathbf{V}$, but it is a strong restriction for algorithms from rich primitives (stacks, higher type constructs, etc.)

- True for a program E which computes some $f = \overline{E} : A^n \rightharpoonup A_s$ in $\mathbf{A}$

  (and so for the standard, deterministic and non-deterministic models of computation once their natural primitives are identified)

III The Finiteness Axiom

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

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

$\Longrightarrow$ *there is a finite* $\mathbf{U} \subseteq_p \mathbf{A}$ *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 $\vec{x}$

- True for a program E which computes some $f = \overline{E} : A^n \rightharpoonup A_s$ in $\mathbf{A}$

Uniform processes need not be effective

Thm *If a Φ-structure $\mathbf{A}$ is generated by the empty tuple, then every $f : A^n \to A_s$ is computed by some uniform process of $\mathbf{A}$*

So every $f : \mathbb{N}^n \to \mathbb{N}$ is computed by a uniform process of $(\mathbb{N}, 0, S)$

Proof
Let $d(\vec{x}) = \min\{m : \vec{x}, f(\vec{x}) \in G_m(\mathbf{A}, \emptyset) \cup \{\mathrm{tt}, \mathrm{ff}\}\}$ and define α by

$$\overline{\alpha}^{\mathbf{U}}(\vec{x}) = w \iff f(\vec{x}) = w \;\&\; \mathbf{G}_{d(\vec{x})}(\mathbf{A}, \emptyset) \subseteq_p \mathbf{U}$$

The Homomorphism Axiom holds because if $\mathbf{G}_{d(\vec{x})}(\mathbf{A}, \emptyset) \subseteq_p \mathbf{U}$, then every homomorphism $\pi : \mathbf{U} \to \mathbf{V}$ is the identity on $\mathbf{G}_{d(\vec{x})}(\mathbf{A}, \emptyset)$

Complexity measures for uniform processes

- A substructure norm on $\mathbf{A}$ assigns to each $\vec{x} \in A^n$ and each finite $\mathbf{U} \subseteq_p \mathbf{A}$ generated by $\vec{x}$ a number $\mu(\mathbf{U}, \vec{x}) \in \mathbb{N}$
- $\boxed{C_\mu(\alpha, \vec{x}) = \min\{\mu(\mathbf{U}, \vec{x}) : \mathbf{U} \vdash_c \alpha(\vec{x}) = w\}}$
- $\text{calls}_{\Phi_0}(\alpha, \vec{x}) = \min\{|\text{eqdiag}(\mathbf{U} \restriction \Phi_0)| : \mathbf{U} \vdash_c \alpha(\vec{x}) = w\}$
  (the least number of calls to $\phi \in \Phi_0$ α must do to compute $\overline{\alpha}^{\mathbf{A}}(\vec{x})$)
- $\text{size}(\alpha, \vec{x}) = \min\{|U| : \mathbf{U} \vdash_c \alpha(\vec{x}) = w\}$
  (the least number of elements of $\mathbf{A}$ that α must see)
- $\text{depth}(\alpha, \vec{x}) = \min\{\text{depth}(\mathbf{U}, \vec{x}) : \mathbf{U} \vdash_c \alpha(\vec{x}) = w\}$
  (the least number of calls α must execute in sequence)

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

These measures are $\le$ the standard measures for programs

(they count only distinct calls)

⋆ The forcing and certification relations

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

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

$$\boxed{\vec{x} \in U^n \;\&\; f(\vec{x}) \in U_s \;\&\; \pi(f(\vec{x})) = f(\pi(\vec{x}))}$$

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

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

Thm *If α is a uniform process of $\mathbf{A}$ which computes $f : A^n \rightharpoonup A_s$, then*

$$\mathbf{U} \vdash_c \alpha(\vec{x}) = w \implies \mathbf{U} \Vdash^{\mathbf{A}}_c f(\vec{x}) = w$$

Proof is immediate by the Homomorphism Axiom

$\star$ Intrinsic (certification) complexities

Suppose $f : A^n \rightharpoonup A_s$ is computed by some uniform process of $\mathbf{A}$ and μ is a substructure norm on $\mathbf{A}$

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

- $C_\mu(\mathbf{A}, f, \vec{x}) = \min\{\mu(\mathbf{U}, \vec{x}) : \mathbf{U} \Vdash_c^{\mathbf{A}} f(\vec{x}) = w\} \quad (f(\vec{x})\downarrow)$

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

- For every uniform process of $\mathbf{A}$ which computes f

$$C_\mu(\mathbf{A}, f, \vec{x}) \le C_\mu(\alpha, \vec{x}) \quad (f(\vec{x})\downarrow)$$

The Homomorphism Test

Lemma

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

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

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

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

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

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

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

Theorem

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

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

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

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

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

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

Outline

1.1. Recursive (McCarthy) Programs.
Preliminaries and notation. The Church-Turing Thesis.

1.2. Uniform processes (The main dish).
An axiomatic approach to the theory of algorithms from specified primitives in the style of *abstract model theory*.

- **2.1. Lower bounds in arithmetic** (arithmetic complexity).
Robust lower bounds for coprimeness (joint with van den Dries).

2.2. Lower bounds in algebra (algebraic complexity).
Robust lower bounds results for 0-testing of polynomials over fields, extending results of Peter Bürgisser (with others).

Recall the method

- (Partial) Φ-structure $\mathbf{A} = (A, \{\phi^{\mathbf{A}}\}_{\phi\in\Phi})$, $f : A^n \rightharpoonup A_s$
- Homomorphism $\pi : \mathbf{U} \to \mathbf{V}$
- For each substructure norm μ $(\mathrm{calls}_{\Phi_0}, \mathrm{size}, \mathrm{depth})$ we defined the intrinsic complexity measure $\boxed{\vec{x} \mapsto C_\mu(\mathbf{A}, f, \vec{x}) \in \mathbb{N} \cup \{\infty\}}$
- If α computes f in $\mathbf{A}$, then

$$C_\mu(\mathbf{A}, f, \vec{x}) \le C_\mu(\alpha, \vec{x}) \qquad (f(\vec{x})\downarrow)$$

Lemma (The Homomorphism Test)

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

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

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

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

The motivating conjecture

The Euclidean algorithm for coprimeness:

$$\perp\!\!\!\perp(x, y) = \mathrm{eq}_1(\mathrm{gcd}(x, y))$$
$$\text{where } \{\mathrm{gcd}(x, y) = \text{if } \mathrm{eq}_0(\mathrm{rem}(x, y)) \text{ then } y \text{ else } \mathrm{gcd}(y, \mathrm{rem}(x, y))$$

- Tail recursion in $\mathbf{N}_\varepsilon = (\mathbb{N}, \mathrm{eq}_0, \mathrm{eq}_1, \mathrm{rem})$

$$\mathrm{calls}_{\{\mathrm{rem}\}}(\varepsilon, a, b) \leq 2 \log b \qquad (a \geq b \geq 2)$$

- Conjecture: *For some $r > 0$ and infinitely many $a \geq b \geq 1$,*

$$\mathrm{calls}_{\{\mathrm{rem}\}}(\mathbf{N}_\varepsilon, \perp\!\!\!\perp, a, b) \geq r \log a$$

We will discuss four relevant results

(a) Stein's binary algorithm α_{st} for the gcd and $\perp\!\!\!\perp$

Thm The gcd satisfies the following recursive equation for $x, y \geq 1$:

$$\gcd(x,y) = \begin{cases} x & \text{if } x = y, \\ 2\gcd(\mathrm{iq}_2(x), \mathrm{iq}_2(y)) & \text{ow., if } \mathrm{parity}(x) = \mathrm{parity}(y) = 0, \\ \gcd(\mathrm{iq}_2(x), y) & \text{ow., if } \mathrm{parity}(x) = 0, \mathrm{parity}(y) = 1, \\ \gcd(x, \mathrm{iq}_2(y)) & \text{ow., if } \mathrm{parity}(x) = 1, \mathrm{parity}(y) = 0, \\ \gcd(x \dot{-} y, y) & \text{ow., if } x > y, \\ \gcd(x, y \dot{-} x) & \text{otherwise.} \end{cases}$$

where $x \dot{-} y = \text{if } x < y \text{ then } 0 \text{ else } x - y$, $\mathrm{iq}_2(x) = \mathrm{iq}(x, 2)$

- α_{st} is a tail recursive program of the structure

$$\mathbf{N}_{\mathrm{st}} = (\mathbb{N}, 0, 1, =, <, \mathrm{parity}, \mathrm{em}, \mathrm{iq}_2, \dot{-}\,)$$

with $\mathrm{em}(x) = 2x$, and for some C,

$$\boxed{\mathrm{calls}(\alpha_{\mathrm{st}}, x, y) \leq C \max\{\log x, \log y\} \quad (x, y \geq 2)}$$

Stein is suboptimal from its primitives

Thm (van den Dries-ynm, 2004, 2009) *If* $b > 2$ *and* $a = b^2 - 1$ *then* $a \perp\!\!\!\perp b$ *and*

$$\boxed{\mathrm{depth}(\mathbf{N}_{\mathrm{st}}, \perp\!\!\!\perp, a, b) \geq \frac{1}{10} \log a}$$

It follows that for some K *and all* $b > 2$*,* $a = b^2 - 1$*,*

$$\mathrm{depth}(\alpha_{\mathrm{st}}, a, b) \leq K \mathrm{depth}(\mathbf{N}_{\mathrm{st}}, \perp\!\!\!\perp, a, b),$$
$$\mathrm{calls}(\alpha_{\mathrm{st}}, a, b) \leq K \mathrm{calls}(\mathbf{N}_{\mathrm{st}}, \perp\!\!\!\perp, a, b)$$

- A uniform process α of a Φ-structure **A** is suboptimal for $f : A^n \to A_s$ relative to a substructure norm μ, if for some $K > 0$,

$$\text{for infinitely many } \vec{a}, C_\mu(\alpha, \vec{a}) \leq K C_\mu(\mathbf{A}, f, \vec{a})$$

- α_{st} is suboptimal for gcd and $\perp\!\!\!\perp$ relative to both depth and calls

Proof of the suboptimality of Stein's algorithm

For $\mathbf{N}_{\mathrm{st}} = (\mathbb{N}, 0, 1, =, <, \mathrm{parity}, \mathrm{em}, \mathrm{iq}_2, \dot{-}\,), \quad b > 2, a = b^2 - 1,$

$$\text{show } \boxed{\mathrm{depth}(\mathbf{N}_{\mathrm{st}}, \perp\!\!\!\perp, a, b) \geq \frac{1}{10} \log a}$$

Must prove that for every finite $\mathbf{U} \subseteq_p \mathbf{N}_{\mathrm{st}}$, generated by a, b,

$$\mathrm{depth}(\mathbf{U}, a, b) < \frac{1}{10} \log a \implies (\exists \pi : \mathbf{U} \to \mathbf{N}_{\mathrm{st}})\Big(\pi(a), \pi(b) \text{ not coprime}\Big)$$

Lemma (Very easy)

If $2^{2m+3} < b$, then every $x \in G_m(\mathbf{N}_{\mathrm{st}}, a, b)$ can be expressed uniquely in the form

$$x = \frac{x_0 + x_1 a + x_2 b}{2^m} \quad \text{with } x_i \in \mathbb{Z}, |x_i| \leq 2^{2m} \text{ for } i \leq 2$$

Proof of Thm Set $\pi(x) = \dfrac{x_0 + x_1 \lambda a + x_2 \lambda b}{2^m}$ with $\lambda = 1 + 2^m$. Then

$\pi : \mathbf{G}_m(\mathbf{N}_{\mathrm{st}}, a, b) \rightarrowtail \mathbf{N}_{\mathrm{st}}$ is an embedding and $\pi(a) = \lambda a, \pi(b) = \lambda b$

Additional and related results about Presburger primitives

- The primitives of $\mathbf{N}_{\mathrm{st}}$ are (piecewise linear) Presburger functions, elementarily definable in the additive semigroup $(\mathbb{N}, 0, 1, +, =)$
- For every Presburger structure $\mathbf{A} = (\mathbb{N}, \mathbf{\Phi})$, there is an $r > 0$ such that for all $b > 2, a = b^2 - 1$,

$$\mathrm{depth}(\mathbf{A}, \perp\!\!\!\perp, a, b) \geq r \log a$$

- For each of the unary relations

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

  and every Presburger structure $\mathbf{A}$, there is some $r > 0$ such that for infinitely many a, $R(a)$ and $\mathrm{depth}(\mathbf{A}, R, a) \geq r \log a$
- Divisibility. Let $x \mid y \iff x$ divides y. For every Presburger structure $\mathbf{A}$, there is an $r > 0$ such that for infinitely many a, b,

$$a \mid b \ \&\ \mathrm{depth}(\mathbf{A}, \mid, a, b) \geq r \log b$$

(b) A lower bound for coprimeness on $\mathbb{N}$ from rem

Let $\mathbf{A} = (\mathbb{N}, \mathrm{iq}, \mathrm{rem}, \mathbf{\Psi})$, with $\mathbf{\Psi}$ a 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),

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

In particular, the conclusion of $(\star)$ holds with some r

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

- ξ is irrational and $a\xi^2 + b\xi + c = 0$ with some $a, b, c \in \mathbb{Z}$
- Infinitely many (a, b) satisfy the hypothesis of $(\star)$
- Pell pairs: infinitely many, hyp. of $(\star)$ holds with $\xi = \sqrt{2}$
- $F_0 = 0, F_1 = 1, F_{k+2} = F_k + F_{k+1}$
  (F_{k+1}, F_k) satisfies the hyp. of $(\star)$ with $\xi = \frac{1+\sqrt{5}}{2}$

How much number theory is needed?

- *For every irrational real number $\xi > 0$, there are infinitely many coprime pairs (a, b) such that*

$$\left|\xi - \frac{a}{b}\right| < \frac{1}{b^2}.$$

These are the good approximations of ξ.

- *Liouville's Theorem for degree 2: For every quadratic irrational ξ, there is a number $C > 0$ such that for all $x, y \in \mathbb{Z}$,*

$$\left|\xi - \frac{x}{y}\right| > \frac{1}{Cy^2}.$$

- *If $\xi > 1$ is a quadratic irrational, then there is a number $c = c(\xi) > 1$ such that every interval $(2^k, 2^{ck})$ contains a good approximation (a, b) of ξ, i.e., $2^k < a, b < 2^{ck}$.*

- For the specific examples, we also need the quoted basic facts about Pell pairs and Fibonacci numbers

The gist of the proof

Let $\mathbf{A} = (\mathbb{N}, \mathrm{iq}, \mathrm{rem}, \mathbf{\Psi})$, with $\mathbf{\Psi}$ a 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),

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

Lemma (Not so easy)

For every quadratic irrational $\xi > 1$, there is a number $\ell = \ell(\xi)$ such that for all but finitely many good approximations (a, b) of ξ and every $m < \frac{1}{2\ell} \log\log a$, every number in $G_m(\mathbf{A}, a, b)$ can be expressed uniquely in the form

$$x = \frac{x_0 + x_1 a + x_2 b}{x_3} \quad \textit{with } x_i \in \mathbb{Z}, |x_i| < 2^{2^{\ell m}} \textit{ for } i \leq 3.$$

Set $\pi : \mathbf{G}_m(\mathbf{A}, a, b) \rightarrowtail \mathbf{A}$ by $\pi(x) = \dfrac{x_0 + x_1 \lambda a + x_2 \lambda b}{x_3}$, with $\lambda = 1 + a!$

(c) How much off are we?

- Conjecture: For some $r > 0$ and infinitely many $a \geq b \geq 1$,

$$\mathrm{calls}_{\{\mathrm{rem}\}}(\mathbf{N}_\varepsilon, \perp\!\!\!\perp, a, b) \geq r \log a$$

- We have: For some $r > 0$ and all (F_{k+1}, F_k) with $k \geq 3$,

$$\mathrm{depth}_{\{\mathrm{rem}\}}(\mathbf{N}_\varepsilon, \perp\!\!\!\perp, F_{k+1}, F_k) \geq r \log\log F_{k+1}$$

Theorem (Pratt, unpublished)

There is a non-deterministic $\mathbf{N}_\varepsilon$-program of ε_{nd} which decides coprimeness, is not less effective than the Euclidean for all inputs and

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

- The conclusion of our theorem is best possible for its hypotheses
- The Conjecture may still be true—for some infinite set of inputs other than the successive Fibonacci numbers
- Pratt's proof depends on classical (but not easy) properties of the Fibonacci numbers

(d) Non-uniform complexity

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

$\mathbf{A} = (\mathbb{N}, \mathrm{iq}, \mathrm{rem}, \mathbf{\Psi})$ with Presburger $\mathbf{\Psi}$ as before. For any N, and any one of the intrinsic complexities as above, let

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

Theorem (van den Dries-ynm 2009)

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

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

- The proof requires a simple new idea (which introduces the size measure) but no more number theory
- We do not know how to derive a non-uniform lower bound for $\mathrm{depth}(\mathbf{A}, \perp\!\!\!\perp, 2^N)$.

Outline

1.1. Recursive (McCarthy) Programs.
Preliminaries and notation. The Church-Turing Thesis.

1.2. Uniform processes (The main dish).
An axiomatic approach to the theory of algorithms from specified primitives in the style of *abstract model theory*.

2.1. Lower bounds in arithmetic (arithmetic complexity).
Robust lower bounds for coprimeness (joint with van den Dries).

- **2.2. Lower bounds in algebra** (algebraic complexity).
Robust lower bounds results for 0-testing of polynomials over fields, extending results of Peter Bürgisser (with others).

Horner's rule for polynomial evaluation

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

$$a_0 + a_1 x \quad \text{(1 multiplication and 1 addition)}$$

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

Subtractions and divisions might help, e.g., using

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

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

Every computation sequence in the real field $(\mathbb{R}, 0, 1, +, -, \cdot, \div)$ *requires at least* n *multiplications/divisions and at least* n *additions/subtractions to compute* $a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ *if* $\vec{a}, x$ *are algebraically independent real numbers*

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

The nullity relation on a field F (0-testing):

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

Decide using Horner's Rule: n multiplications, n additions, one $=$ - test

Theorem

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

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

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

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

(4) $\text{calls}_{\{+,-,=\}}(\mathbf{R}, N_{\mathbb{R}}, \vec{a}, x) = n$ (Horner's Rule not optimal)

For algebraic decision trees, (1) is due to Bürgisser and Lickteig (1992) and results like (2) – (4) are due to Bürgisser, Lickteig and Shub (1992)

Lemma. If $n \geq 1$, then $\text{calls}_{\{+,-\}}(\mathbf{R}, N_{\mathbb{R}}, \vec{a}, x) \leq n - 1$

$$N_{\mathbb{R}}(a_0, \dots, a_n, x) \iff a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = 0$$

Proof. Using Horner's rule and $\leq (n-1)$ additions, compute

$$w = a_1 + a_2 x + \cdots + a_n x^{n-1}$$

and then follow the following steps to check if $a_0 + wx = 0$ using only multiplications and equality tests:

- Give the correct answer if $w = 0$ or $a_0 = 0$
- Ow., if $a_0^2 \neq (xw)^2$, give output ff
- Ow., $a_0 = \pm xw$, so if $a_0 = xw$, give output ff
- Ow., give output tt

(The algorithm works in every field of characteristic $\neq 2$)

Lemma. If $n \geq 1$ and $\vec{a}, x$ are algebraically independent, then $\text{calls}_{\{+,-,=\}}(\mathbf{R}, N_{\mathbb{R}}, \vec{a}, x) \leq n$

(Horner's rule requires $\text{calls}_{\{+,-,=\}}(\mathbf{R}, N_{\mathbb{R}}, \vec{a}, x) = n + 1$ in this case)

Proof for the case $n = 1$. If a_0, a_1, x are algebraically independent real numbers, $\mathbf{U} \subseteq_p \mathbf{R}$ and

$$\text{eqdiag}(\mathbf{U}) = \{a_1 x = u, \boxed{a_0 + u = w}, \frac{x}{w} = v\},$$

then every homomorphism $\pi : \mathbf{U} \to \mathbf{R}$ must be defined on v and satisfy

$$\pi(v) = \frac{\pi(b)}{\pi(w)},$$

so $\pi(w) = \pi(a_0) + \pi(a_1)\pi(x) \neq 0$; Hence $\mathbf{U} \Vdash_c^{\mathbf{R}} a_0 + a_1 x \neq 0$, and so

$$\text{calls}_{\{+,-,=\}}(\mathbf{R}, N_{\mathbb{R}}, a_1, a_2, b) \leq 1.$$

- Used division rather than $=$ - test. Not an algorithm of $\mathbf{R}$

Thm. If $n \geq 1$ and $\vec{a}, x$ are algebraically independent, then $\text{calls}_{\{+,-,=\}}(\mathbf{R}, N_{\mathbb{R}}, \vec{a}, x) \geq n$

By the hypothesis, $a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \neq 0$;

so to appeal to the Homomorphism Test, we need to prove that

Lemma *If $\mathbf{U} \subseteq_p \mathbf{R}$ is finite, generated by $\vec{a}, x$, and*

$$|\text{eqdiag}(\mathbf{U} \restriction \{+,-,=\})| < n,$$

then there exists a homomorphism $\pi : \mathbf{U} \to \mathbf{R}$ such that

$$\pi(a_0) + \pi(a_1)\pi(x) + \ldots + \pi(a_n)\pi(x)^n = 0$$

- Follows from a much stronger lemma, proved by induction on n

How much field theory is needed? (Very little)

- For a field F and indeterminates $\vec{u} = u_1, \ldots, u_k$,
  - $F[\vec{u}]$ is the polynomial ring of all finite sums (finite X)

    $$\chi(\vec{u}) = \sum f_{b_1,\ldots,b_k} u_1^{b_1} \cdots u_k^{b_k} \quad (b_1, \ldots, b_k \in X \subset \mathbb{N}, f_{b_1,\ldots,b_k} \in F)$$

  - $F(\vec{u})$ is the field of rational functions

    $$\chi = \frac{\chi_n(\vec{u})}{\chi_d(\vec{u})} \quad (\chi_n(\vec{u}), \chi_d(\vec{u}) \in F[\vec{u}], \chi_d(\vec{u}) \neq 0)$$

- A partial field homomorphism $\pi : F_1 \rightharpoonup F_2$ is a field homomorphism $\pi : F_1' \to F_2$ on some subfield $F_1' \subseteq F_2$.

  It is proper on $U \subseteq F_1'$, if $\Big(x \in U \;\&\; \pi(x) = 0\Big) \implies x = 0$

- A substitution $v \mapsto \psi(v, \vec{u})$ defines a partial field homomorphism $\pi : F(v, \vec{u}) \rightharpoonup F(v, \vec{u})$

  $$\pi\Big(\frac{\chi_n(v, \vec{u})}{\chi_d(v, \vec{u})}\Big) = \frac{\chi_n(\psi(v, \vec{u}), \vec{u})}{\chi_d(\psi(v, \vec{u}), \vec{u})}$$

  defined when $\chi_d(\psi(v, \vec{u}), \vec{u}) \neq 0$ (a subfield of $F(v, \vec{u})$)

Algebraic independence in $\mathbb{R}$

- $a_1, \ldots, a_k$ in $\mathbb{R}$ are algebraically independent, if there is no $\chi(u_1, \ldots, u_k) \in \mathbb{Q}[\vec{u}]$ such that

$$\chi(a_1, \ldots, x_a) = 0$$

- $\mathbb{K}$ is the field of algebraic real numbers, satisfying $q_0 + q_1 x + \cdots + q_n x^n = 0$ with some $q_0, \ldots, q_n \in \mathbb{Q}$
- For positive real numbers $a_1, \ldots, a_n$,

$$\mathrm{Roots}(\vec{a}) = \{{a_i}^b \mid i = 1, \ldots, n, b \in \mathbb{Q}\}$$

- For reals $\vec{u} = u_1, \ldots, u_k$ and positive reals $\vec{a}$,

$$\begin{aligned}\mathbb{K}^*(\vec{u}; \vec{a}) &= \mathbb{K}(\{u_1, \ldots, u_k\} \cup \mathrm{Roots}(a_1, \ldots, a_n)) \\ &= \text{the rational functions of algebraic numbers,} \\ &\qquad u_1, \ldots, u_k \text{ and rational powers of } a_1, \ldots, a_n\end{aligned}$$

- $\mathbf{K}^*(\vec{u}; \vec{a}) = (\mathbb{K}^*(\vec{u}; \vec{a}), 0, 1, +, -, \cdot, \div, =)$

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

- For $\mathbf{U} \subseteq_p \mathbf{K}^*(x,z;\vec{a})$ and $\phi \in \{+,-,=\}$,

  $(\phi, u, v, w) \in \mathrm{eqdiag}(\mathbf{U})$ is trivial if $u, v \in \mathbb{K}(x,z)$

- *Suppose*
  - *$n \in \mathbb{N}$, $\overline{g} \in \mathbb{K}$, $\overline{g} \neq 0$,*
  - *$x, z, a_1, \ldots, a_n$ are algebraically independent with $a_1, \ldots, a_n > 0$,*
  - *$\mathbf{U} \subseteq_p \mathbf{K}^*(x,z;\vec{a})$ is finite, generated by*

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

  - $\mathrm{eqdiag}(\mathbf{U})$ *has $< n$ non-trivial $\{+,-,=\}$-entries;*

- *Then there is a partial field homomorphism*

  $$\pi : \mathbb{K}^*(x,z;\vec{a}) \to \mathbb{K}^*(x;\vec{a}) \text{ such that:}$$

  (a) *π is total and proper on U;*
  (b) *π is the identity on $\mathbb{K}(x)$; and*
  (c) $\pi(z) + \overline{g}\Big(\pi(a_1)x + \cdots + \pi(a_n)x^n\Big) = 0.$

Proof is by induction on n, using a sequence of substitutions

The good news

- Recursive programs on first order structures $\mathbf{A} = (A, \mathbf{\Phi})$
  - Simulate faithfully "all" models of relative computation
  - Much complexity theory can be studied directly for them
  - Express faithfully all relative algorithms?
- Uniform processes on first order structures $\mathbf{A} = (A, \mathbf{\Phi})$
  - Definition motivated by properties of recursive programs
  - They capture the uniformity, not the effectiveness of algorithms
  - The carry a rich theory of complexity
  - They suggest the definition of intrinsic complexity measures for functions and relations
  - They justify the Homomorphism Test which can ground the derivation of robust (absolute) lower bounds
- Applications to arithmetic and algebraic complexity
  - Coprimeness on $\mathbb{N}$, from various primitives
  - Testing polynomials for 0

The bad news

- The structure of binary numbers

$$\mathbf{N}_b = (\mathbb{N}, 0, \mathrm{parity}, \mathrm{iq}_2, \mathrm{em}, \mathrm{om}, \mathrm{eq}_0),$$

  where $\mathrm{em}(x) = 2x, \quad \mathrm{om}(x) = 2x + 1$
- $|x|$ = the length of the binary expansion of x, $\sim \log x$
- Thm *For every unary relation* $R : \mathbb{N} \to \{\mathrm{tt}, \mathrm{ff}\}$ (e.g., $\mathrm{Prime}(x)$)

$$\mathrm{calls}(\mathbf{N}_b, R, x) \leq |x| - 1$$

- Proof If $x = x_0 + 2x_1 + 2^2 x_2 + \cdots + 2^m x_m$ with $|x| = m + 1$ and

$$\mathrm{eqdiag}(\mathbf{U}) = \{2x_m + x_{m-1} = u_1,\ 2u_1 + x_{m-2} = u_2, \\ \ldots, 2u_{m-1} + x_0 = u_m\},$$

  then every $\pi : \mathbf{U} \to \mathbf{N}_b$ fixes x, so $\mathbf{U} \Vdash_c^{\mathbf{N}_b} R(x) = w$ (correct w)
- *Cannot prove by the Homomorphism Method that for all* $\mathbf{N}_b$*-algorithms* α *and some* $r > 0$, $\mathrm{calls}(\alpha, \mathrm{Prime}, p) \geq r(\log p)^2$
- Ultimately, we need to analyze algorithms (recursive programs?)