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

\[
(\varepsilon) \quad \text{gcd}(a, b) = \text{if } (\text{rem}(a, b) = 0) \text{ then } b \text{ else } \text{gcd}(b, \text{rem}(a, b))
\]

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

\[
a = bq + \text{rem}(a, b) \quad (0 \leq \text{rem}(a, b) < b)
\]

calls\(_{\varepsilon}\)(\(a, b\)) = the number of divisions required to compute \(\text{gcd}(a, b)\)

by the Euclidean algorithm

\[
\leq 2 \log(b) = 2 \log_2(b) \quad (a \geq b \geq 2)
\]

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

\[
a \perp b \iff \text{gcd}(a, b) = 1
\]

- Most relevant complexity: number of required divisions
- Looking for absolute lower bounds, which restrict 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
  
  $$R(\vec{x}) = \text{if } R \text{ holds at } \vec{x} \text{ then } \top \text{ else } \bot$$

- A (partial) structure is a tuple $A = (A, \Phi^A)$
  
  where $\Phi$ is a set of function (and relation) symbols
  
  and $\Phi^A = \{ \phi^A \}_{\phi \in \Phi}$. Here

  $\phi^A : A^{n_\phi} \rightarrow A_s$

  with $s = \text{sort}(\phi) = a$ or $\text{sort}(\phi) = \text{boole}$

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

  $\phi^A : A^{n_\phi} \rightarrow A$ or $\phi^A : A^{n_\phi} \rightarrow \{ \top, \bot \}$

- $N_\varepsilon = (\mathbb{N}, \text{rem}, =_0, =_1)$, the Euclidean structure

- $N_\varepsilon \upharpoonright U = (U, \text{rem} \upharpoonright U, =_0 \upharpoonright U, =_1 \upharpoonright U)$ 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 $\Phi$ and a set $A$, the $(\Phi \cup A)$-terms are defined by

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

where $\phi \in \Phi$, $x \in A$ (viewed as a constant or parameter) and $v_0, v_1, \ldots$ 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 $\text{sort}(t_1) \equiv \text{boole}$ and $\text{sort}(t_2) \equiv \text{sort}(t_3) \equiv \text{sort}(t)$
- $t$ is pure: no parameters (a $\Phi$-term)
- $t$ is closed: no variables
- If $t$ is closed and $A$ is a $\Phi$-structure, then

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

$$A \models t = s \iff \left( \text{den}(A, t) \uparrow \& \text{den}(A, s) \uparrow \right) \text{ or } \text{den}(A, t) = \text{den}(A, s)$$
Recursive (McCarthy) programs — syntax

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

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

where

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

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

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

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

A program is deterministic if its recursive variables are all distinct
Recursive programs — (call-by-value) semantics

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

\[
\text{ClTerms}(E, A) = \text{the set of closed } (\text{voc}(E) \cup A)\text{-terms}
\]

Need to define the relation

\[
E, A \vdash t = w \iff t \in \text{ClTerms}(E, A), w \in A \cup \{ \texttt{tt}, \texttt{ff} \}
\]

& \( w \) is one of the values assigned to \( t \) by \( E \) in \( A \)

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

\[
t_1 = u_1, \ldots, t_{n_\phi} = u_{n_\phi} \in S \text{ & } \phi^A(u_1, \ldots, u_{n_\phi}) = w \\
\implies \phi(t_1, \ldots, t_{n_\phi}) = w \in S
\]

\begin{itemize}
\item (2) Implementations (many, deterministic if \( E \) is deterministic): \\
There is a computation \( t \leftarrow \boxed{s_1 \rightarrow \cdots \rightarrow s_m} \rightleftharpoons w \)
\end{itemize}
The recursive (two stack) machine $\mathcal{P}(E, A)$

- **(pass)** \( \vec{a} \ x : \vec{b} \rightarrow \vec{a} : x \vec{b} \ (x \in A \cup \{\text{tt, ff}\}) \)

- **(e-call)** \( \vec{a} \ \phi : \vec{x} \vec{b} \rightarrow \vec{a} : \phi^A(\vec{x}) \vec{b} \)

- **(i-call)** \( \vec{a} \ p_i : \vec{u} \vec{b} \rightarrow \vec{a} \ E_i(\vec{u}, \vec{p}) : \vec{b} \)

- **(comp)** \( \vec{a} \ h(t_1, \ldots, t_n) : \vec{b} \rightarrow \vec{a} \ h t_1 \cdots t_n : \vec{b} \)

- **(br)** \( \vec{a} \ \text{if } t_0 \text{ then } t_1 \text{ else } t_2 : \vec{b} \rightarrow \vec{a} \ t_1 \ t_2 \ ? : t_0 : \vec{b} \)

- **(br0)** \( \vec{a} \ t_1 \ t_2 \ ? : \text{tt} \vec{b} \rightarrow \vec{a} \ t_1 : \vec{b} \)

- **(br1)** \( \vec{a} \ t_1 \ t_2 \ ? : \text{ff} \vec{b} \rightarrow \vec{a} \ t_2 : \vec{b} \)

- **States** are sequences of the form $\vec{L} : \vec{R}$, where
  - $\vec{L}$ is a tuple from ClTerms($E, A$) \( \cup \) voc($E$) \( \cup \) \{?\} and $\vec{R}$ a tuple from $A \cup \{\text{tt, ff}\}$

- **Input** $t \leftarrow \underline{t : \text{Terminal states}}$ $w$  **Output** $\underline{w \leftarrow w}$

- The underlined part is the *trigger* for the transition

- **In the external call (e-call)**, $\phi \in \Phi$ and \( \text{arity}(\phi) = n_{\phi} = \text{length of } \vec{x} \)

- **In the internal call (i-call)**, $p_i(\vec{u}) = E_i(\vec{u}, \vec{p})$ is an equation of $E$
A-recursive functions

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

- A partial function \( f : A^n \rightarrow A_s \) is computed by \( E \) in \( A \) if

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

- At most one partial function is computed by \( E \) in \( A \)

\[ \text{rec}(A) = \{ f : A^n \rightarrow A_s : f \text{ is computed in } A \text{ by a deterministic } E \} \]

\[ \text{rec}_{\text{nd}}(A) = \{ f : A^n \rightarrow A_s : f \text{ is computed in } A \text{ by some } E \} \]

In general, \( \text{rec}(A) \subsetneq \text{rec}_{\text{nd}}(A) \)
Recursive programs — complexity measures

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

- \( \text{calls}_{\Phi_0}(E, A)(t = w) = \) the least number of calls to the primitives in \( \Phi_0 \subseteq \Phi \) that \( E \) must execute to prove \( t = w \)
- \( \text{depth}(E, A)(t = w) = \) the least number of calls to the primitives in \( \Phi \) that \( E \) must execute in sequence to prove \( t = w \)
- \( \text{size}(E, A)(t = w) = \) 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 \( f : A^n \rightarrow A_s \), they give complexity measures
  \[
  \text{calls}_{\Phi_0}(E, A, \vec{x}), \text{depth}(E, A, \vec{x}), \text{size}(E, A, \vec{x}) \quad (f(\vec{x}) \downarrow)
  \]
- **Thm.** \( \text{depth}(E, A, \vec{x}) \leq \text{size}(E, A, \vec{x}) \leq \text{calls}_{\Phi}(E, A, \vec{x}) \)
Recursive programs — special forms

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

- **Terms**: \( k = 0 \), so \( E \equiv E_0(\vec{x}) \) is a \( \Phi \)-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**, \( E \equiv E_0(\vec{x}, \vec{p}) \) 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 \( \Phi \)-terms test(\( \vec{u} \)), 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 $\text{rec}(A)$, $\text{rec}_{\text{nd}}(A)$ and the class $\text{tail}(A)$ of tail recursive functions on arbitrary $A$
  (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: \( f : A^n \rightarrow A \) is (recursively) \textit{computable from} \( \phi_1, \ldots, \phi_m \)
\[ \iff f \in \text{rec}(A, \phi_1, \ldots, \phi_m) \]

- RCT: The fundamental algorithmic constructs are calling (composition), branching, and grounded self reference
- The definition of \( \text{rec}(A) \) does not involve any objects outside \( A \)
- \( A \)-recursion is a Tarski logical notion, preserved by permutations

• The \textit{natural numbers} are the structure \( N = (\mathbb{N}, 0, S, =) \)
  (Dedekind: up to isomorphism, \ldots, the modern structuralist view)

•• \( f : \mathbb{N} \rightarrow \mathbb{N} \) is recursive \( \iff f \in \text{rec}(\mathbb{N}, 0, S, =) \)

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

Turing computability on \( \mathbb{N} = \) recursion + 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 $\Phi_0 \subseteq \Phi$,
  
  $$A \upharpoonright \Phi_0 = (A, \{\phi^A : \phi \in \Phi_0\})$$

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

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

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

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

- $\pi$ is an embedding if it is injective
- $\pi$ 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 $\pi[U]$ has universe $\pi[U]$ and

  $$\text{eqdiag}(\pi[U]) = \{(\phi, \pi(\vec{x}), \pi(w)) : (\phi, \vec{x}, w) \in \text{eqdiag}(U)\}$$
Substructures, generation

- Substructures:

  \[ U \subseteq_p A \iff U \subseteq A \]

  & the identity \( I : U \to A \) is an embedding

  \[ \iff U \subseteq A \& \text{eqdiag}(U) \subseteq \text{eqdiag}(A) \]

\[
G_0(U, \bar{x}) = \{x_1, \ldots, x_n\}, \\
G_{m+1}(U, \bar{x}) = G_m(U, \bar{x}) \cup \{\phi^U(u_1, \ldots, u_{n\phi}) : u_1, \ldots, u_{n\phi} \in G_m(U, \bar{x})\} \\
G_m(U, \bar{x}) = U \upharpoonright G_m(U, \bar{x})
\]

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

\[
\text{depth}(U, \bar{x}) = \min\{m : U = G_m(U, \bar{x})\} \quad (U \text{ finite, generated by } \bar{x})
\]

- We use finite \( U \subseteq_p A \) generated by the input \( \bar{x} \) to represent calls to the primitives executed during a computation in \( A \)
Algorithms from primitives – the basic intuition

An $n$-ary algorithm $\alpha$ of $A = (A, \Phi)$ (or from $\Phi$) of sort $s$ “computes” some $n$-ary partial function

$$\bar{\alpha} = \bar{\alpha}^A : A^n \rightarrow A_s \quad (A_s = A \text{ or } A_s = \{tt, ff\})$$

using the primitives in $\Phi$ as oracles

We understand this to mean that in the course of a “computation” of $\bar{\alpha}(\vec{x})$, the algorithm may request from the oracle for any $\phi^A$ any particular value $\phi^A(\vec{u})$, for arguments $\vec{u}$ which it has already computed, and that if the oracles cooperate, then “the computation” of $\bar{\alpha}(\vec{x})$ 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
1. The Locality (or relativization) Axiom

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

\[
\overline{\alpha}^U : U^n \rightarrow U_s
\]

It computes the partial function \( \overline{\alpha}^A : A^n \rightarrow A_s \)

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

We write

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

- True for a program \( E \) which computes some \( f = \overline{E} : A^n \rightarrow A_s \) in \( A \) by

\[
\overline{E}^U(\vec{x}) = w \iff E, U \vdash E_0(\vec{x}, \vec{p}) = w
\]
II The Homomorphism Axiom

If $\alpha$ is an $n$-ary uniform process of $A$, $U, V \subseteq_p A$, and $\pi : U \to V$ is a homomorphism, then

$$U \vdash \alpha(\vec{x}) = w \implies V \vdash \alpha(\pi\vec{x}) = \pi w$$  \hspace{1em} (x_1, \ldots, x_n \in U, w \in U_s)

In particular, if $U \subseteq_p A$, then $\overline{\alpha}^U \subseteq \overline{\alpha}^A$

- For algorithms: when asked for $\phi^U(\vec{x})$, the oracle for $\phi$ may consistently provide $\phi^V(\pi\vec{x})$, if $\pi$ is a homomorphism
- This is obvious for the identity embedding $I : U \hookrightarrow V$, but it is a strong restriction for algorithms from rich primitives (stacks, higher type constructs, etc.)

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

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

If $\alpha$ is an $n$-ary uniform process of $A$, then

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

$\implies$ there is a finite $U \subseteq_p A$ generated by $\vec{x}$ such that $U \vdash \alpha(\vec{x}) = w$

- 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

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

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

- True for a program $E$ which computes some $f = \overline{E} : A^n \rightarrow A_s$ in $A$
Uniform processes need not be effective

**Thm** If a \( \Phi \)-structure \( A \) is generated by the empty tuple, then every \( f : A^n \to A \) is computed by some uniform process of \( A \).

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

**Proof**

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

\[
\overline{\alpha}^U(\vec{x}) = w \iff f(\vec{x}) = w \land G_{d(\vec{x})}(A, \emptyset) \subseteq_p U
\]

The Homomorphism Axiom holds because if \( G_{d(\vec{x})}(A, \emptyset) \subseteq_p U \), then every homomorphism \( \pi : U \to V \) is the identity on \( G_{d(\vec{x})}(A, \emptyset) \).
Complexity measures for uniform processes

- A substructure norm on $A$ assigns to each $\vec{x} \in A^n$ and each finite $U \subseteq_p A$ generated by $\vec{x}$ a number $\mu(U, \vec{x}) \in \mathbb{N}$

$$C_\mu(\alpha, \vec{x}) = \min \{ \mu(U, \vec{x}) : U \vdash_c \alpha(\vec{x}) = w \}$$

- $\text{calls}_{\Phi_0}(\alpha, \vec{x}) = \min \{|\text{eqdiag}(U \upharpoonright \Phi_0)| : U \vdash_c \alpha(\vec{x}) = w \}$
  (the least number of calls to $\phi \in \Phi_0$ $\alpha$ must do to compute $\overline{\alpha}(\vec{x})$)

- $\text{size}(\alpha, \vec{x}) = \min \{|U| : U \vdash_c \alpha(\vec{x}) = w \}$
  (the least number of elements of $A$ that $\alpha$ must see)

- $\text{depth}(\alpha, \vec{x}) = \min \{ \text{depth}(U, \vec{x}) : U \vdash_c \alpha(\vec{x}) = w \}$
  (the least number of calls $\alpha$ must execute in sequence)

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

*These measures are $\leq$ the standard measures for programs*

*they count only distinct calls*
The forcing and certification relations

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

- A homomorphism \( \pi : U \rightarrow A \) respects \( f \) at \( \vec{x} \) if

\[
\vec{x} \in U^n \land f(\vec{x}) \in U_s \land \pi(f(\vec{x})) = f(\pi(\vec{x}))
\]

\( U \vdash_A f(\vec{x}) = w \iff \) every homomorphism \( \pi : U \rightarrow A \) respects \( f \) at \( \vec{x} \)

\( U \vdash_A f(\vec{x}) = w \iff U \) is finite, generated by \( \vec{x} \) and \( U \vdash_A f(\vec{x}) = w \)

**Thm** If \( \alpha \) is a uniform process of \( A \) which computes \( f : A^n \rightarrow A_s \), then

\[
U \vdash_c \alpha(\vec{x}) = w \implies U \vdash_A f(\vec{x}) = w
\]

**Proof** is immediate by the Homomorphism Axiom
Intrinsic (certification) complexities

Suppose \( f : A^n \rightarrow A_s \) is computed by some uniform process of \( A \) and \( \mu \) is a substructure norm on \( A \)

\[
\text{U} \models^A f(\vec{x}) = w \iff \text{U} \text{ is finite, generated by } \vec{x} \text{ and } \text{U} \models^A f(\vec{x}) = w
\]

\[
\begin{align*}
\text{C}_\mu(A, f, \vec{x}) &= \min \{ \mu(\text{U}, \vec{x}) : \text{U} \models^A f(\vec{x}) = w \} \quad (f(\vec{x}) \downarrow) \\
\text{calls}_{\Phi_0}(A, f, \vec{x}) &= \min \{ |\text{eqdiag}(\text{U} \uparrow \Phi_0)| : \text{U} \models^A f(\vec{x}) = w \} \\
\text{size}(A, f, \vec{x}) &= \min \{ |U| : \text{U} \models^A f(\vec{x}) = w \} \\
\text{depth}(A, f, \vec{x}) &= \min \{ \text{depth}(\text{U}, \vec{x}) : \text{U} \models^A f(\vec{x}) = w \}
\end{align*}
\]

\[
\text{For every uniform process of } A \text{ which computes } f \\
\text{C}_\mu(A, f, \vec{x}) \leq \text{C}_\mu(\alpha, \vec{x}) \quad (f(\vec{x}) \downarrow)
\]
The Homomorphism Test

Lemma
Suppose $\mu$ is a substructure norm (calls $\Phi_0$, size, depth) on a $\Phi$-structure $A$, $f : A^n \to A_s$, $f(\vec{x}) \downarrow$, and

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

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

then $C_\mu(A, f, \vec{x}) \geq m$. 
The best uniform process for $f : A^n \rightarrow A_s$ in $A$

Define $\beta_{f,A}$ by

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

Theorem

The following are equivalent for a $\Phi$-structure $A$ and $f : A^n \rightarrow A_s$:

(i) Some uniform process $\alpha$ of $A$ computes $f$.

(ii) $(\forall \vec{x}, w) \left( f(\vec{x}) = w \implies (\exists U \subseteq_p A)[U \models^A f(\vec{x}) = w] \right)$.

(iii) $\beta_{f,A}$ is a uniform process of $A$ which computes $f$.

Moreover, if these conditions hold, then for every uniform process $\alpha$ which computes $f$ in $A$ and all complexity measures $C_\mu$ as above,

$$C_\mu(A, f, \vec{x}) = C_\mu(\beta_{f,A}, \vec{x}) \leq C_\mu(\alpha, \vec{x}) \ (f(\vec{x}) \downarrow).$$
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 $A = (A, \{\phi^A\}_{\phi \in \Phi})$, $f : A^n \rightarrow A_s$
- Homomorphism $\pi : U \rightarrow V$
- For each substructure norm $\mu$ (calls $\Phi_0$, size, depth) we defined the intrinsic complexity measure $\vec{x} \mapsto C_\mu(A, f, \vec{x}) \in \mathbb{N} \cup \{\infty\}$
- If $\alpha$ computes $f$ in $A$, then

\[ C_\mu(A, f, \vec{x}) \leq C_\mu(\alpha, \vec{x}) \quad (f(\vec{x}) \downarrow) \]

Lemma (The Homomorphism Test)

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

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

\[ \left( f(\vec{x}) \in U_s \land \mu(U, \vec{x}) < m \right) \implies \left( \exists \pi : U \rightarrow A \right)[f(\pi(\vec{x})) \neq \pi(f(\vec{x}))]; \]

*then* $C_\mu(A, f, \vec{x}) \geq m$. 
The motivating conjecture

The Euclidean algorithm for coprimeness:

\[ \bot(x, y) = eq_1(gcd(x, y)) \]

where \( \{gcd(x, y) = \text{if } eq_0(rem(x, y)) \text{ then } y \text{ else } gcd(y, rem(x, y)) \} \)

- Tail recursion in \( \mathbb{N}_\varepsilon = (\mathbb{N}, eq_0, eq_1, rem) \)

\[
\text{calls}_{\{\text{rem}\}}(\varepsilon, a, b) \leq 2 \log b \quad (a \geq b \geq 2)
\]

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

\[
\text{calls}_{\{\text{rem}\}}(\mathbb{N}_\varepsilon, \bot, a, b) \geq r \log a
\]

We will discuss four relevant results
(a) Stein’s binary algorithm $\alpha_{st}$ for the gcd and $\bot$

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

$$\gcd(x, y) = \begin{cases} 
  x & \text{if } x = y, \\
  2 \gcd(iq_2(x), iq_2(y)) & \text{ow., if } \text{parity}(x) = \text{parity}(y) = 0, \\
  \gcd(iq_2(x), y) & \text{ow., if } \text{parity}(x) = 0, \text{parity}(y) = 1, \\
  \gcd(x, iq_2(y)) & \text{ow., if } \text{parity}(x) = 1, \text{parity}(y) = 0, \\
  \gcd(x - y, y) & \text{ow., if } x > y, \\
  \gcd(x, y - x) & \text{otherwise.}
\end{cases}$$

where $x - y = \text{if } x < y \text{ then } 0 \text{ else } x - y$, $iq_2(x) = iq(x, 2)$

• $\alpha_{st}$ is a tail recursive program of the structure

$$N_{st} = (\mathbb{N}, 0, 1, =, <, \text{parity}, \text{em}, iq_2, \div)$$

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

$$\text{calls}(\alpha_{st}, x, y) \leq C \max\{\log x, \log y\} \quad (x, y \geq 2)$$
Stein is suboptimal from its primitives

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

$$\text{depth}(\mathcal{N}_{st}, \perp, a, b) \geq \frac{1}{10} \log a$$

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

$$\text{depth}(\alpha_{st}, a, b) \leq K \text{depth}(\mathcal{N}_{st}, \perp, a, b),$$

$$\text{calls}(\alpha_{st}, a, b) \leq K \text{calls}(\mathcal{N}_{st}, \perp, a, b)$$

- A uniform process $\alpha$ of a $\Phi$-structure $\mathcal{A}$ is **suboptimal** for $f : \mathcal{A}^n \rightarrow \mathcal{A}_s$ relative to a substructure norm $\mu$, if for some $K > 0$,

  for infinitely many $\vec{a}$, $C_\mu(\alpha, \vec{a}) \leq KC_\mu(\mathcal{A}, f, \vec{a})$

- $\alpha_{st}$ is suboptimal for gcd and $\perp$ relative to both depth and calls
Proof of the suboptimality of Stein’s algorithm

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

show $\text{depth}(\mathbb{N}_{st}, \bot, a, b) \geq \frac{1}{10} \log a$

Must prove that for every finite $U \subseteq_{p} \mathbb{N}_{st}$, generated by $a, b$,

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

Lemma (Very easy)

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

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

Proof of Thm Set $\pi(x) = \frac{x_0 + x_1 \lambda a + x_2 \lambda b}{2^m}$ with $\lambda = 1 + 2^m$. Then $\pi : \mathbb{G}_m(\mathbb{N}_{st}, a, b) \hookrightarrow \mathbb{N}_{st}$ is an embedding and $\pi(a) = \lambda a, \pi(b) = \lambda b$
Additional and related results about Presburger primitives

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

$$\text{depth}(A, \bot, a, b) \geq r \log a$$

- For each of the unary relations

  $x$ is prime, $x$ is a perfect square, $x$ is square free

and every Presburger structure $A$, there is some $r > 0$ such that for infinitely many $a$, $R(a)$ and $\text{depth}(A, R, a) \geq r \log a$

- **Divisibility.** Let $x \mid y \iff x$ divides $y$. For every Presburger structure $A$, there is an $r > 0$ such that for infinitely many $a, b$,

$$a \mid b \& \text{depth}(A, \mid, a, b) \geq r \log b$$
(b) A lower bound for coprimeness on \( \mathbb{N} \) from \( \text{rem} \)

Let \( \mathcal{A} = (\mathbb{N}, i_q, \text{rem}, \Psi) \), with \( \Psi \) a finite set of \textit{Presburger functions}.

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

\[ \text{If } \xi > 1 \text{ is quadratic irrational, then for some } r > 0 \text{ and all sufficiently large coprime } (a, b), \]

\[
\left| \xi - \frac{a}{b} \right| < \frac{1}{b^2} \implies \text{depth}(\mathcal{A}, \bot, a, b) \geq r \log \log a. \tag{\star} \]

\text{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 \(k \geq 3\).

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

- Liouville’s Theorem for degree 2: For every quadratic irrational $\xi$, 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 $\xi$, 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 $A = (\mathbb{N}, \text{iq}, \text{rem}, \Psi)$, with $\Psi$ 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 $(a, b)$,

$$\left| \xi - \frac{a}{b} \right| < \frac{1}{b^2} \implies \text{depth}(A, \perp, a, b) \geq r \log \log a.$$  \hfill (*)

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 $\xi$ and every $m < \frac{1}{2\ell} \log \log a$, every number in $G_m(A, a, b)$ can be expressed uniquely in the form

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

Set $\pi : G_m(A, a, b) \mapsto A$ by $\pi(x) = \frac{x_0 + x_1 \lambda a + x_2 \lambda b}{x_3}$, with $\lambda = 1 + a!$.
(c) How much off are we?

- **Conjecture**: For some $r > 0$ and infinitely many $a \geq b \geq 1$,
  
  \[ \text{calls}_{\{ \text{rem} \}}(N_{\varepsilon}, \bot, a, b) \geq r \log a \]

- **We have**: For some $r > 0$ and all $(F_{k+1}, F_k)$ with $k \geq 3$,
  
  \[ \text{depth}_{\{ \text{rem} \}}(N_{\varepsilon}, \bot, F_{k+1}, F_k) \geq r \log \log F_{k+1} \]

**Theorem (Pratt, unpublished)**

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

\[ \text{calls}(\varepsilon_{nd}, F_{k+1}, F_k) \leq K \log \log F_{k+1} \]

- 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?

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

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

Theorem (van den Dries-ynm 2009)

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

$$\text{calls}(A, \bot, 2^N) \geq \text{size}(A, \bot, 2^N) \geq r \log N.$$

- 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 depth($A, \bot, 2^N$).
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 $F$:

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

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

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):

\[
N_F(a_0, \ldots, a_n, x) \iff a_0 + a_1x + a_2x^2 + \cdots + a_nx^n = 0
\]

Decide using Horner’s Rule: \( n \) multiplications, \( n \) additions, one \( = - \) test

**Theorem**

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

1. \( \text{calls}\{\cdot, \div\}(R, N_R, \vec{a}, x) = n \)
2. \( \text{calls}\{\cdot, \div, =\}(R, N_R, \vec{a}, x) = n + 1 \)
3. \( \text{calls}\{+, -\}(R, N_R, \vec{a}, x) = n - 1 \) (Somewhat unexpected)
4. \( \text{calls}\{+, -, =\}(R, N_R, \vec{a}, x) = n \) (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 \( \text{calls}_{\{+,-\}}(\mathbb{R}, N_\mathbb{R}, \vec{a}, x) \leq n - 1 \)

\[
N_\mathbb{R}(a_0, \ldots, a_n, x) \iff a_0 + a_1x + a_2x^2 + \cdots + a_nx^n = 0
\]

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

\[
w = a_1 + a_2x + \cdots + a_nx^{n-1}
\]

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 calls\( \{+, -, =\}(R, N_R, \vec{a}, x) \leq n \)

(Horner’s rule requires calls\( \{+, -, =\}(R, N_R, \vec{a}, x) = n + 1 \) in this case)

Proof for the case \( n = 1 \). If \( a_0, a_1, x \) are algebraically independent real numbers, \( U \subseteq_p R \) and

\[
eq\text{eqdiag}(U) = \{a_1x = u, \overbrace{a_0 + u = w}^{\text{mark}}, \frac{x}{w} = v\},
\]

then every homomorphism \( \pi : U \to R \) 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 \( U \models^R a_0 + a_1x \neq 0 \), and so calls\( \{+, -, =\}(R, N_R, a_1, a_2, b) \leq 1 \).

- Used division rather than \( = \)-test. Not an algorithm of \( R \)
Thm. If $n \geq 1$ and $\vec{a}, x$ are algebraically independent, then
\[ \text{calls}\{+,-,=\}(\mathbb{R}, N_{\mathbb{R}}, \vec{a}, x) \geq n \]

By the hypothesis, $a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \neq 0$;
so to appeal to the Homomorphism Test, we need to prove that

**Lemma** If $U \subseteq_p \mathbb{R}$ is finite, generated by $\vec{a}, x$, and
\[ |\text{eqdiag}(U \upharpoonright \{+,-,=\})| < n, \]
then there exists a homomorphism $\pi : U \rightarrow \mathbb{R}$ such that
\[ \pi(a_0) + \pi(a_1)\pi(x) + \cdots + \pi(a_n)\pi(x)^n = 0 \]

- 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 $\bar{u} = u_1, \ldots, u_k$,
  - $F[\bar{u}]$ is the polynomial ring of all finite sums ($\text{finite } X$)
    \[
    \chi(\bar{u}) = \sum f_{b_1,\ldots,b_k} u_1^{b_1} \cdots u_k^{b_k} \quad (b_1, \ldots, b_k \in X \subset \mathbb{N}, f_{b_1,\ldots,b_k} \in F)
    \]
  - $F(\bar{u})$ is the field of rational functions
    \[
    \chi = \frac{\chi_n(\bar{u})}{\chi_d(\bar{u})} \quad (\chi_n(\bar{u}), \chi_d(\bar{u}) \in F[\bar{u}], \chi_d(\bar{u}) \neq 0)
    \]

- A partial field homomorphism $\pi : F_1 \rightarrow F_2$ is a field homomorphism $\pi : F'_1 \rightarrow F_2$ on some subfield $F'_1 \subseteq F_2$.
  
  It is proper on $U \subseteq F'_1$, if \( (x \in U \& \pi(x) = 0) \implies x = 0 \)

- A substitution $v \mapsto \psi(v, \bar{u})$ defines a partial field homomorphism $\pi : F(v, \bar{u}) \rightarrow F(v, \bar{u})$
  \[
  \pi \left( \frac{\chi_n(v, \bar{u})}{\chi_d(v, \bar{u})} \right) = \frac{\chi_n(\psi(v, \bar{u}), \bar{u})}{\chi_d(\psi(v, \bar{u}), \bar{u})}
  \]
  defined when $\chi_d(\psi(v, \bar{u}), \bar{u}) \neq 0$ (a subfield of $F(v, \bar{u})$)
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 $a_1, \ldots, a_n$,
  $$\text{Roots}(\vec{a}) = \{a_i^b \mid i = 1, \ldots, n, b \in \mathbb{Q}\}$$

- For reals $\vec{u} = u_1, \ldots, u_k$ and positive reals $\vec{a}$,
  $$\mathbb{K}^*(\vec{u}; \vec{a}) = \mathbb{K}(\{u_1, \ldots, u_k\} \cup \text{Roots}(a_1, \ldots, a_n))$$
  $$= \text{the rational functions of algebraic numbers,}$$
  $$u_1, \ldots, u_k \text{ and rational powers of } a_1, \ldots, a_n$$

- $\mathbb{K}^*(\vec{u}; \vec{a}) = (\mathbb{K}^*(\vec{u}; \vec{a}), 0, 1, +, -, \cdot, \div, =)$
The lemma for calls $\{+, -, =\}(\mathbb{R}, \mathbb{N}_\mathbb{R}, \vec{a}, x) \geq n$

- For $U \subseteq p \mathbb{K}^*(x, z; \vec{a})$ and $\phi \in \{+, -, =\}$,
  \[
  (\phi, u, v, w) \in \text{eqdiag}(U) \text{ is trivial if } u, v \in \mathbb{K}(x, z)
  \]

- **Suppose**
  - $n \in \mathbb{N}$, $\bar{g} \in \mathbb{K}$, $\bar{g} \neq 0$,
  - $x, z, a_1, \ldots, a_n$ are algebraically independent with $a_1, \ldots, a_n > 0$,
  - $U \subseteq p \mathbb{K}^*(x, z; \vec{a})$ is finite, generated by
    \[
    (U \cap \mathbb{K}) \cup \{x, z\} \cup (U \cap \text{Roots}(a_1, \ldots, a_n))
    \]
  - $\text{eqdiag}(U)$ has $< n$ non-trivial $\{+, -, =\}$-entries;

- **Then** there is a partial field homomorphism
  \[
  \pi : \mathbb{K}^*(x, z; \vec{a}) \rightarrow \mathbb{K}^*(x; \vec{a}) \text{ such that:}
  \]
  (a) $\pi$ is total and proper on $U$;
  (b) $\pi$ is the identity on $\mathbb{K}(x)$; and
  (c) $\pi(z) + \bar{g}\left(\pi(a_1)x + \cdots + \pi(a_n)x^n\right) = 0$.

**Proof** is by induction on $n$, using a sequence of substitutions
Recursive programs on first order structures $\mathcal{A} = (A, \Phi)$
- 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 $\mathcal{A} = (A, \Phi)$
- Definition motivated by properties of recursive programs
- They capture the uniformity, not the effectiveness of algorithms
- They 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

\[ N_b = (\mathbb{N}, 0, \text{parity}, iq_2, \text{em}, \text{om}, eq_0), \]

where \( \text{em}(x) = 2x, \) \( \text{om}(x) = 2x + 1 \)

- \( |x| = \) the length of the binary expansion of \( x, \sim \log x \)

- **Thm** For every unary relation \( R : \mathbb{N} \to \{\text{tt}, \text{ff}\} \) (e.g., Prime(\( x \)))

\[
\text{calls}(N_b, R, x) \leq |x| - 1
\]

- **Proof** If \( x = x_0 + 2x_1 + 2^2x_2 + \cdots + 2^mx_m \) with \( |x| = m + 1 \) and

\[
eq \text{eqdiag}(U) = \{2x_m + x_{m-1} = u_1, \ 2u_1 + x_{m-2} = u_2, \ \ldots, 2u_{m-1} + x_0 = u_m\},
\]

then every \( \pi : U \to N_b \) fixes \( x, \) so \( U \models_{N_b} R(x) = w \) (correct \( w \))

- **Cannot prove by the Homomorphism Method that for all**

\( N_b\)-algorithms \( \alpha \) **and some** \( r > 0, \) \( \text{calls}(\alpha, \text{Prime}, p) \geq r(\log p)^2 \)

- Ultimately, we need to analyze algorithms (recursive programs?)