Algebraic Characterization of Computable and Complexity-Theoretic Analysis

Walid Gomaa

INRIA Nancy Grand-Est Research Centre
CARTE team

January 12, 2009

I. Real Computation

- Two independent approaches: Recursive Analysis & Numerical Analysis
- Recursive analysis investigates real computation using classical recursion theory and its technologies
- Numerical analysis is an algorithmic problem-oriented approach
- Compare with classical recursion theory vs. analysis and design of algorithms
- Recursive analysis is the focus of this presentation. It was introduced by Turing [1936], Grzegorczyk [1955], and Lacombe [1955].

Representation of Real Numbers

- Given x, several representations:
  - Binary expansion: $BE_x : \mathbb{N} \to \{0, 1\}$
  - Left cut: $LC_x = \{r \in \mathbb{Q} : r < x\}$
  - Cauchy Sequence: $CF_x : \mathbb{N} \to \mathbb{Q}$
- Recursively-wise they are equivalent
- However, they differ on the sub-recursive as well as the complexity-theoretic level

Cauchy Sequence Representation

- Let $\mathbb{D}$ be the set of dyadic rationals, i.e, any $d \in \mathbb{Q}$ with finite binary representation $d = \frac{k}{2^m}$
- Binary converging Cauchy sequences are adopted:

$$\varphi_x : \mathbb{N} \to \mathbb{D}$$
$$|\varphi_x(n) - x| \leq 2^{-n}$$
$$\{\varphi_x(n)\} \rightsquigarrow x$$

- $x \in \mathbb{R}$ is *computable* if it has a *computable Cauchy sequence*
- Transcendental numbers such as π, e are computable
- The set of computable real numbers forms a real closed field

Complexity of Real Numbers

- $x \in \mathbb{R}$ is in complexity class $\mathcal{C}$ if x has a Cauchy sequence in $\mathcal{C}$
- The class of polytime computable real numbers forms a real closed field
- $PTime^{[BE]} \equiv PTime^{[LC]} \subsetneq PTime^{[CF]}$

- *Type 2 Turing machine* model
- It is a Type 1 machine equipped with *function oracles*
- The oracles are Cauchy functions for the real input arguments

Definition

A real function f is computable over a compact domain iff there exists a T2 machine $M^{()}$ such that for every $x \in dom(f)$ there exists $\varphi \in CF_x$ such that for every $n \in \mathbb{N}$, $M^{\varphi}(n)$ computes $d \in \mathbb{D}$ with $|f(x) - d| \leq 2^{-n}$

Char. of Computable Real Functions in Recursive Analysis

- Let $D \subseteq \mathbb{R}$ be *compact*

Theorem

$f : D \to \mathbb{R}$ is computable iff

1. f has a recursive *modulus of continuity* $m : \mathbb{N} \to \mathbb{N}$:

$$\forall x, y \in dom(f) : \; |x - y| \leq 2^{-m(n)} \implies |f(x) - f(y)| \leq 2^{-n}$$

2. f has a recursive *approximation function* $\psi : \mathbb{D} \times \mathbb{N} \to \mathbb{D}$:

$$\forall d \in \mathbb{D} : \forall n \in \mathbb{N} : |\psi(d, n) - f(d)| \leq 2^{-n}$$

Modulus of Continuity

Given the the distance between x and its approximation $d \in \mathbb{D}$, the modulus is used to estimate how good is the approximating value $f(d)$ to the desired value $f(x)$

$$\begin{aligned} & d \approx x \\ & \psi(d,n) \approx f(d) \\ & f(d) \approx f(x) \\ & \therefore \psi(d,n) \approx f(x) \end{aligned}$$

Proof of the Theorem

- Given $x \in dom(f)$, $\varphi \in CF_x$, and $n \in \mathbb{N}$:
  1. M^{φ} computes $m(n+1)$ and writes it on the oracle tape
  2. The oracle responds with $d = \varphi(m(n+1))$
  3. M^{φ} computes and outputs $e = \psi(d, n+1)$
- $|e - f(x)| \leq |e - f(d)| + |f(d) - f(x)| \leq \overbrace{2^{-(n+1)}}^{\text{by definition of } \psi} + \overbrace{2^{-(n+1)}}^{\text{by modulus } m} = 2^{-n}$

Computability Implies Continuity

Theorem

Let f be a real function. If f is computable, then it is continuous.

Polytime Complexity of Computable Real Functions

- Assume compact domains
- Complexity as a function of n rather than $|n|_2 = O(\log n)$

Theorem

$f : D \to \mathbb{R}$ is *PTime*-computable iff

1. The modulus of f over D is a polynomial function
2. The approximation function is polynomial time computable in n

- *Machine independent resource-free algebraic* characterization of computational and complexity classes
- Descriptive complexity is a *model-theoretic resource-free* characterization of complexity classes
- Problem with descriptive complexity: characterization of relations rather than functions

Function Algebras Cont'd

$$\mathcal{F} = [\mathcal{B}; \mathcal{O}]$$

- $\mathcal{B}$ is a set of basic functions
- $\mathcal{O}$ is a set of operations
- $\mathcal{F}$ consists of $\mathcal{B}$ and its closure under $\mathcal{O}$

- The class of *elementary functions*:

$$\mathcal{E} = [0, s, u, \ominus; Comp, BSum, BProd] = \mathcal{E}^2$$

- $\ominus$ is the *cutoff subtraction*: $x \ominus y = \max\{0, x - y\}$
- Bounded Sum: $f = BSum(g)$, $(\bar{x}, y) \xrightarrow{f} \sum_{z<y} g(\bar{x}, z)$
- Bounded Product: $f = BProd(g)$, $(\bar{x}, y) \xrightarrow{f} \prod_{z<y} g(\bar{x}, z)$.
- $\mathcal{E}^2$: second level of *Grzegorczyk hierarchy*

- The class of *primitive recursive functions*:

$$\mathcal{PR} = [0, s, u; Comp, Rec]$$

- Primitive recursive: $h = Rec(g, f)$,

$$h(0, \bar{y}) = g(\bar{y})$$
$$h(x + 1, \bar{y}) = f(x, \bar{y}, h(x, \bar{y}))$$

- $\mathcal{PR}$ gives all *total* recursive functions
- $\mathcal{PR}$ is the union of the Grzegorczyk hierarchy
- $\mathcal{PR}$ is closed under space and time complexity

- *Cobham's class* [1964]: no machine model, but it contains explicit resource bounds

- Bellantoni-Cook class [BC92]

$$B = [0, u, s_i, pr, cond; SComp, SRN]$$

- Successor over notation: $s_i(;x) = x \circ i = 2x + i$

- Predecessor: $pr(;xi) = x$

- Conditional:

$$cond(;x,y,z) = \begin{cases} y & x \equiv_2 0 \\ z & ow \end{cases}$$

- Safe composition: from $h, \bar{g}_1, \bar{g}_2 \in B$

$$f(\bar{x};\bar{y}) = h(\bar{g}_1(\bar{x};);\bar{g}_2(\bar{x};\bar{y}))$$

  - Safe arguments can't be placed in normal positions, but the opposite can occur
  - Asymmetry in the definition, hence adding a function operating on safe arguments is generally more powerful than adding the same function on normal arguments

Capturing Polynomial Time Cont'd

- Safe Predicative recursion: from $g, h_0, h_1 \in B$

$$f(0, \bar{y}; \bar{z}) = g(\bar{y}; \bar{z})$$
$$f(s_i(;x), \bar{y}; \bar{z}) = h_i(x, \bar{y}; \bar{z}, f(x, \bar{y}; \bar{z}))$$

  - The recurrence variable must be in normal position on the LHS
  - The recurred value must be in safe position on the RHS, this what actually controls the growth rate of the function by preventing nested recursions
  - Operations of safe inputs do not increase the input length by more than an additive constant

Examples

$$add(0; y) = y$$
$$add(x+1; y) = s(; add(x; y))$$
$$mul(0; y) = 0$$
$$mul(x+1; y) = add(; mul(x, y)) \qquad \text{can't be done}$$
$$mul(0, y;) = 0$$
$$mul(x+1, y;) = add(y; mul(x, y))$$

no safe argument in *mul*, hence exponentiation can not be defined

Examples

$$
\begin{aligned}
add(0; y) &= y \\
add(x+1; y) &= s(; add(x; y)) \\
mul(0; y) &= 0 \\
mul(x+1; y) &= add(; mul(x, y)) \qquad \textit{can't be done} \\
mul(0, y;) &= 0 \\
mul(x+1, y;) &= add(y; mul(x, y))
\end{aligned}
$$

no safe argument in *mul*, hence exponentiation can not be defined

$$add(0; y) = y$$
$$add(x+1; y) = s(; add(x; y))$$
$$mul(0; y) = 0$$
$$mul(x+1; y) = add(; mul(x, y)) \qquad \textit{can't be done}$$
$$mul(0, y;) = 0$$
$$mul(x+1, y;) = add(y; mul(x, y))$$

no safe argument in *mul*, hence exponentiation can not be defined

- The class of functions built in analogy with $\mathcal{E}$ [C01]:

$$\mathcal{L} = [0, 1, -1, \pi, u, \theta_3; Comp, LI]$$

- θ_3 is a $\mathcal{C}^2$ function gives a physically realistic way to *sense inequalities without introducing discontinuities*:

$$\theta_3(x) = \begin{cases} x^3 & x \geq 0 \\ 0 & ow \end{cases}$$

- Linear integration: $f = LI(g, h_1, h_2)$ is the maximal solution of

$$\begin{aligned} f(0, \bar{y}) &= g(\bar{y}) \\ \delta_x f(x, \bar{y}) &= h_1(x, \bar{y}) f(x, \bar{y}) + h_2(x, \bar{y}) \end{aligned}$$

Examples

All of the following functions are in $\mathcal{L}$:

1. $f(x) = exp(x)$, solution of

$$f(0) = 1$$
$$f' = f$$

2. $f(x) = exp^{[m]}(x)$, by composition

3. $f(x) = sin(x)$, first component of the solution of

$$h_1(0) = 0, \; h_2(0) = 1$$
$$h_1' = h_2$$
$$h_2' = -h_1$$

- $\mathcal{E} = [0, s, u, \ominus; Comp, BSum, BProd] = \mathcal{E}^2$ (elementary functions over $\mathbb{N}$)
- $\mathcal{L} = [0, 1, -1, \pi, u, \theta_3; Comp, LI]$ (built in analogy with $\mathcal{E}$)
- $\mathcal{E}(\mathbb{R})$: Elementary functions over $\mathbb{R}$ (the corresponding functional is elementary)

Theorem

1. $DP(\mathcal{L}) = \mathcal{E}$
2. $\mathcal{L} \subsetneq \mathcal{E}(\mathbb{R})$
3. $\mathcal{L} + Lim = \mathcal{E}(\mathbb{R})$ (over compact domains)

1. Let $D \times I$ be a *compact set* where $D \subseteq \mathbb{R}^k$ and $I \subseteq \mathbb{R}$

$$f : D \times I \to \mathbb{R}$$

2. $\forall \bar{x} \in D$, $f(\bar{x}, y)$ is monotonically increasing on I
3. $\forall \bar{x} \in D$, $f(\bar{x}, y)$ has a unique root y_0 in the interior of I
4. $\delta_y f(\bar{x}, y) \mid_{y_0} > 0$

Unique Min:

$$!\mu(f) : D \to \mathbb{R}$$
$$\bar{x} \mapsto y_0$$

- $\mathcal{L}+!\mu = [0, 1, -1, \pi, u, \theta_3; Comp, LI, !\mu]$
- $\mathcal{R}ec$: recursive functions over $\mathbb{N}$

Theorem (BH06)

$$\mathcal{R}ec = DP(\mathcal{L}+!\mu)$$

- $f : D \times I \longrightarrow \mathbb{R}$
- for all $\bar{x} \in D$, $\|\delta_t f(\bar{x}, t)\| \rightsquigarrow 0$ as $t \rightsquigarrow \infty$

<u>Limit:</u>

$$
\begin{aligned}
Lim(f) &: D \to \mathbb{R} \\
\bar{x} &\longmapsto \lim_{t \to \infty} f(\bar{x}, t)
\end{aligned}
$$

- $\mathcal{L}^*_{!\mu} = [0, 1, u, \theta_3; Comp, LI, !\mu, Lim]$
- $\mathcal{R}ec(\mathbb{R})$: Recursive functions over $\mathbb{R}$ (the corresponding functional is recursive)

Theorem (BH06)

$$\mathcal{L}^*_{!\mu} = \mathcal{R}ec(\mathbb{R})$$

- Undergoing project with M. Bournez and M. Hainry
- Focus on *PTime*

General plan:

1. Definition of polynomial time computation over open bounded domains
2. Definition of polynomial computation over open unbounded domains
3. *PTime* basic functions
4. Operations that sustain feasibility and strong enough to capture to the whole *PTime*