ON TYPE-2 COMPLEXITY CLASSES

Chung-Chih Li
Jim Royer

Syracuse University

CONSTABLE'S QUESTIONS

Constable, in 1973, asked:

Q1 What is the type-2 analogue of PF?

Q2 What is the computational complexity theory of the type-2 effectively continuous functionals?

(Symes' 1971 thesis.)

Terminology

$\mathrm{PF} =_{\mathrm{def}}$ the polynomial-time computable functions (of type $\mathrm{N} \to \mathrm{N}$)

type-2 $=_{\mathrm{def}}$ functions of type $(\mathrm{N} \to \mathrm{N})^k \times \mathrm{N}^\ell \to \mathrm{N}$

effectively continuous functionals

$=_{\mathrm{def}}$ . . . not this talk . . .

Q1: WHAT IS HIGHER-TYPE PTIME?

Why this as a first step?

- Force of habit.
- We need landmarks to ground our work.
- Most examples are too simple or too complex.
- So we need to proceed by analogy.
- In ordinary complexity theory, $\mathbf{P}$ and $\mathbf{PF}$ are useful and flexible reference classes.

But how to proceed?

TWO STANDARD WAYS TO DEFINE A SUBRECURSIVE CLASS

Synthetic/Implicit/P.L.-Based/...

Given a restrictive P.L. or function algebra $\boldsymbol{F}$.
Then the class = the $\boldsymbol{F}$-computable functions.

Example

Cobham's characterization of PF.

Analytic/Explicit/Machine-Based/...

Given

(a) a general machine or P.L. + a cost model
(b) a way of measuring the size of an input, and
(c) a family of bounding functions,

Then the class = the things computable through that model under those resource bounds.

Example

PF = the functions computable on a TM (with the usual cost model) in time polynomial in the length of the input.

THE TYPE-2 BASIC FEASIBLE FUNCTIONALS

BFF_2

$=$ the type-2 PV^ω computable functionals, where
$PV^\omega \approx$ the simply-typed λ calculus $+ \mathcal{R} + \mathrm{PF}$
(Cook-Urquhart)

$=$ the type-2 BTLP computable functionals, where
BTLP $=$ Bounded Typed Loop Programs
(Cook-Kapron)

⋮

$=$ the basic polynomial-time functionals, where
these are the functionals computed by OTMs that
run in time (2nd-order) polynomial in
the (type-0 and -1) lengths of their inputs.
(Kapron-Cook)

There are other notions of type-2 poly-time (e.g., BC),

Q2: WHAT IS THE COMPLEXITY THEORY OF THE EFF. CONTINUOUS FUNCTIONALS?

We don't have an answer to that question yet.

But if we change the question to:

> Q2′: What is the complexity theory of the computable functions of type $(\mathrm{N} \to \mathrm{N}) \times \mathrm{N} \to \mathrm{N}$?

then we have a start at an answer.

Why is Q2′ easier than Q2?

BUT WHAT FLAVOR OF COMPLEXITY THEORY?

CONCRETE

Specific machine/cost models.

Hartmanis and Stearns, 1965

AXIOMATIC/ABSTRACT

Machine independent

Blum, 1967

We'll choose the concrete version of complexity theory
. . . matters are strange enough already.

First, we take a quick look at type-1 (ordinary, old-fashion, . . .) complexity theory.

CONVENTIONS

- $\mathrm{N} \equiv \{0,1\}^*$.

- $M_0,\ M_1, \ldots$ — a standard indexing of TMs.

- $\varphi_i(x) =_{\mathrm{def}}$ the result of running M_i on input x.
  This may be undefined.

- $\Phi_i(x) =_{\mathrm{def}}$ the number of steps taken by M_i on x.
  This may be ∞. (N.B. $\infty \neq \perp$.)

- A TM must read its entire input string.
  $\therefore$ Distinct inputs yield distinct computations.

- Suppose $f, g\colon \mathrm{N} \to \mathrm{N}$
  - $f \leq g$ means $(\forall x)\ f(x) \leq g(x)$.
  - $f \leq^* g$ means $(\overset{\infty}{\forall} x)\ f(x) \leq g(x)$.
  - Similarly, with $f = g$, $f =^* g$, $f < g$, $f <^* g$, etc.

TYPE-1 COMPLEXITY CLASSES

DEFINITION. For each computable $t\colon \mathrm{N} \to \mathrm{N}$,

$$\mathcal{C}(t) =_{\text{def}} \{\, \varphi_i \mid i \in \mathrm{N} \text{ \& } \varphi_i \text{ is total \& } \Phi_i \leq^* t \,\}.$$

We say that $\mathcal{C}(t)$ is the complexity class **named** by t.

A SAMPLE ELEMENTARY RESULT (Rabin, 1960).
For each computable t,
there is some 0–1 valued $f\colon \mathrm{N} \to \mathrm{N}$
such that $f \notin \mathcal{C}(t)$.

Why are complexity classes interesting?

Because they describe boundaries.

OUR GOAL:
TYPE-2 COMPLEXITY CLASSES

Recall the type-1 definition

$$\mathcal{C}(t) =_{\text{def}} \{\, \varphi_i \mid i \in \mathrm{N} \;\&\; \varphi_i \text{ is total } \&\; \Phi_i \leq^* t \,\}.$$

QUESTIONS:

- What replaces φ and Φ?
  What is the machine/cost model?
- What replaces t?
  What is a type-2 complexity bound?
- What replaces $\leq^*$?
  What is the right notion of "finitely many exceptions"?

OUR TYPE-2 MACHINE MODEL

- We use Oracle Turing Machines (OTMs) with function ($\mathbb{N} \to \mathbb{N}$) oracles.
- query $\cong$ call to an oracle: $f(x) = ?$
- Each OTM step has unit cost.
- Each OTM **must** read all of each oracle answer.
  This convention gives us the answer-length cost model.

Why OTMs?

CONVENTIONS, II

- $\mathcal{F}$ = the finite functions over N
- σ ranges over $\mathcal{F}$
- $\overline{\sigma}$ = the mod. of σ that is 0 every place σ is undefined.

- $\widehat{\mathsf{M}}_0,\ \widehat{\mathsf{M}}_1, \dots$ — a standard indexing of OTMs.

- $\widehat{\varphi}_i(f, x) =_{\text{def}}$ the result of running $\widehat{\mathsf{M}}_i$ on oracle f and input x. This may be undefined.

- $\widehat{\Phi}_i(f, x) =_{\text{def}}$ the # of steps taken by $\widehat{\mathsf{M}}_i$ on f and x. This may be ∞.

CONVENTIONS, II — TOO

- $\mathrm{Queries}_i(f, x, n) =_{\mathrm{def}}$ the set of queries $\widehat{\mathsf{M}}_i$ makes on f and x within its first n steps.
- $\mathrm{Use}_i(f, x, n) =_{\mathrm{def}}$

$$\{\, (y, f(y)) : y \in \mathrm{Queries}_i(f, x, n) \,\}$$

- $\mathrm{Queries}_i(f, x) =_{\mathrm{def}} \bigcup_n \mathrm{Queries}_i(f, x, n)$
- $\mathrm{Use}_i(f, x) =_{\mathrm{def}} \bigcup_n \mathrm{Use}_i(f, x, n)$

- $\widehat{\Phi}_i(\sigma, x) =_{\mathrm{def}}$

$$\begin{cases} \widehat{\Phi}_i(\overline{\sigma}, x), & \text{if } \mathrm{Queries}_i(\overline{\sigma}, x) \subseteq \{\, y \mid \sigma(y)\downarrow \,\}; \\ n, & \text{otherwise, where } n = \#\text{ of steps taken up to the issuance of the first query, } y\text{, such that } \sigma(y)\uparrow. \end{cases}$$

TYPE-2 COMPLEXITY BOUNDS

What we don't do and why

Suppose $F\colon (\mathrm{N} \to \mathrm{N}) \times \mathrm{N} \to \mathrm{N}$ is computable.
Then we might say that

> the complexity of $\widehat{\mathsf{M}}_i$ is everywhere bounded by F
>
> iff
>
> for all f and all x, $\widehat{\Phi}_i(f,x) \leq F(f,x)$.

(Symes 71, Kapron 91, Seth 94)

What is wrong with this?

THEOREM.
Suppose that $\widehat{\Phi}_i \leq T$ and $\widehat{\varphi}_b = T$. Then, for all f and x
$\mathrm{Queries}_i(f,x) \subseteq \mathrm{Queries}_b(f,x)$.

$\therefore$ In order for $\widehat{\Phi}_i \leq F$ to hold, $\widehat{\varphi}$-program b on input (f,x) has to make all possible queries that $\widehat{\mathsf{M}}_i$ could make on input (f,x). $\therefore$ The clerk is in control.

TYPE-2 COMPLEXITY BOUNDS

Our approach

- We make the bounding function a passive observer of the computation it is bounding.
- The bounding function determines a bounding value based on what is has seen so far.

DEFINITION. Suppose $\beta\colon \mathcal{F} \times \mathrm{N} \to \mathrm{N}$ is computable.

(a) We say that β determines a weak type-2 time bound iff it satisfies the following:

Nontriviality: $\beta(\sigma, x) \geq |x| + 1$.

Convergence: $\lim_{\tau \to f} \beta(\tau, x){\downarrow} < \infty$.

Boundedness: $\sup_{\tau \subset f} \beta(\tau, x) = \lim_{\tau \to f} \beta(\tau, x)$.

$\mathrm{WB} =_{\mathrm{def}}$ the collection of all such β's.

(b) We say that β determines a strong type-2 time bound iff $\beta \in \mathrm{WB}$ and satisfies:

Monotonicity: $\sigma \subseteq \sigma'$ implies $\beta(\sigma, x) \leq \beta(\sigma', x)$.

$\mathrm{SB} =_{\mathrm{def}}$ the collection of all such β's.

TYPE-2 COMPLEXITY BOUNDS
Our approach, continued

DEFINITION.

- The run time of $\widehat{\varphi}$-program i on input (f, x) is bounded by β (written $\widehat{\varphi}_{i,\beta}(f, x)\Downarrow$) iff for each n,

  $\widehat{\Phi}_i(\sigma_n, x) \le \beta(\sigma_n, x)$, where $\sigma_n = \mathrm{Use}_i(f, x, n)$.

- The computation of $\widehat{\varphi}$-program i on input (f, x) is clipped by β (written $\widehat{\varphi}_{i,\beta}(f, x)\Uparrow$) iff not $\widehat{\varphi}_{i,\beta}(f, x)\Downarrow$.
- $E_{i,\beta} = \{\, (f, x) \mid \widehat{\varphi}_{i,\beta}(f, x)\Uparrow \,\}$.
  We call $E_{i,\beta}$ the exception set for i and β.
- The run time of $\widehat{\varphi}$-program i is everywhere bounded by β iff $E_{i,\beta}$ is empty.

EXAMPLE. Suppose $\widehat{\varphi}_i$ is total and, for each σ and x,

$$\beta(\sigma, x) = \widehat{\Phi}_i(\sigma, x).$$

Then $\beta \in \mathrm{SB}$ and (no surprise) the run time of $\widehat{\varphi}$-program i is everywhere bounded by β.

TYPE-2 ALMOST EVERYWHERE BOUNDS

The short version

Since we are working in function spaces,

finite $\cong$ compact in some topology.

But which topology?

What we don't do: Use $\mathcal{B}$, the Baire space topology.
Why? $E_{i,\beta}$ is compact in $\mathcal{B}$ iff $E_{i,\beta} = \emptyset$.

OUR APPROACH

DEFINITION.
(a) The induced topology for $\widehat{\varphi}$-program i (written $\mathcal{I}_i$) is . . . not today . . .

(b) $\widehat{\varphi}$-program i is almost everywhere bounded by β iff $E_{i,\beta}$ is compact in $\mathcal{I}_i$ (iff there are only finitely many computations $\ni \widehat{\varphi}_{i,\beta}(f,x)\Uparrow$).

TYPE-2 ALMOST EVERYWHERE BOUNDS

The longer version

Since we are working in function spaces,

finite $\cong$ compact in some topology.

But which topology?

DEFINITION.

(a) $((\sigma, x)) =_{\text{def}} \{ (f, x) \mid f \supset \sigma \}$.

(b) $\mathcal{B}$, the Baire space topology on $(\mathrm{N} \to \mathrm{N}) \times \mathrm{N}$, is the topology obtained by taking

$$\{ ((\sigma, x)) \mid \sigma \in \mathcal{F},\ x \in \mathrm{N} \}$$

as basic open sets.

What we don't do: Use $\mathcal{B}$.

Why? $E_{i,\beta}$ is compact in $\mathcal{B}$ iff $E_{i,\beta} = \emptyset$.

TYPE-2 ALMOST EVERYWHERE BOUNDS

Our approach

DEFINITION.

(a) The induced topology for $\widehat{\varphi}$-program i (written $\mathcal{I}_i$) is the topology on $(\mathrm{N} \to \mathrm{N}) \times \mathrm{N}$ determined by taking

$$\{ ((\sigma, x)) \mid (\exists f)[\sigma \subseteq \mathrm{Use}_i(f, x)],\ x \in \mathrm{N} \}$$

as the basic open sets.

(b) $\widehat{\varphi}$-program i is almost everywhere bounded by β iff $E_{i,\beta}$ is compact in $\mathcal{I}_i$.

NOTE.

$\widehat{\varphi}$-program i is almost everywhere bounded by β

iff

there are only finitely many computations $\ni \widehat{\varphi}_{i,\beta}(f, x)\Uparrow$.

(Similar to some ideas of Symes (1971).)

TYPE-2 COMPLEXITY CLASSES FINALLY!

Recall the definition of a type-1 complexity class:

$$\mathcal{C}(t) =_{\text{def}} \{\, \varphi_i \mid i \in \mathrm{N} \ \&\ \varphi_i \text{ is total } \&\ \Phi_i \leq^* t \,\}.$$

DEFINITION. For each $\beta \in \mathrm{WB}$,

$$\widehat{\mathrm{C}}(\beta) =_{\text{def}}$$
$$\{\, \widehat{\varphi}_i \mid i \in \mathrm{N} \ \&\ \widehat{\varphi}_i \text{ total } \&\ E_{i,\beta} \text{ is compact in } \mathcal{I}_i \,\}$$

$\widehat{\mathrm{C}}(\beta)$ is the type-2 complexity class named by β.

SOME SAMPLE ELEMENTARY RESULTS

THEOREM (Li).
Suppose $F \in \widehat{\mathrm{C}}(\beta)$. Then there is a $\widehat{\varphi}$-program i for F and a $c \in \mathrm{N}$ such that $E_{i,c\cdot\beta} = \emptyset$.

THEOREM (Li).
Suppose $\beta \in \mathrm{WB}$. Then there is an 0–1-valued total computable F such that $F \notin \widehat{\mathrm{C}}(\beta)$.

UNIONS OF COMPLEXITY CLASSES

THE SITUATION AT TYPE-1

THEOREM (McCreight & Meyer).
Suppose that $t\colon \mathrm{N}^2 \longrightarrow \mathrm{N}$ is computable and nondecreasing in its first argument. Then there is a computable $g\colon \mathrm{N} \longrightarrow \mathrm{N}$ such that $C(g) = \bigcup_i C(\lambda x\,.\,t(i,x))$.

COROLLARY. PF is a complexity class.

THE SITUATION AT TYPE-2

THEOREM (Li). BFF_2 is not a type-2 complexity class.

THEOREM (Li). Suppose $\beta_0,\ \beta_1, \ldots \in \mathrm{WB}$ satisfy some very strong conditions. Then $\bigcup_i \widehat{\mathrm{C}}(\beta_i)$ may fail to be a complexity class.

THEOREM (Li). Suppose $\beta_0,\ \beta_1, \ldots \in \mathrm{WB}$ satisfy some even stronger conditions. Then for some $\beta \in \mathrm{WB}$, $\widehat{\mathrm{C}}(\beta) = \bigcup_i \widehat{\mathrm{C}}(\beta_i)$.

COROLLARY (Li). Suppose $\beta \in \mathrm{WB}$. Then $\mathbf{O}(\beta)$ is a complexity class, where $\mathbf{O}(\beta) = \bigcup_{a,b\in\mathrm{N}} \widehat{\mathrm{C}}(a \cdot \beta + b)$.

GAPS AND COMPRESSIONS

THE SITUATION AT TYPE-1

THE OPERATOR GAP THEOREM (Constable, Young).
For each computable $\Theta\colon (\mathrm{N} \to \mathrm{N}) \to (\mathrm{N} \to \mathrm{N})$, there is an increasing, computable t such that $\mathcal{C}(t) = \mathcal{C}(\Theta(t))$, in fact, there is no i with $t \leq^* \Phi_i \leq^* \Theta(t)$.

THE COMPRESSION THEOREM (Blum).
There is a computable $r\colon \mathrm{N}^2 \to \mathrm{N}$ such that for all i with φ_i total, we have $C(\Phi_i) \subsetneq C(\lambda x \,.\, r(x, \Phi_i(x)))$.

THE SITUATION AT TYPE-2

Gaps go away.

THE INFLATION THEOREM (Li).
There is a recursive operator Θ such that,
for each $\beta \in \mathrm{WB}$, $\Theta(\beta) \in \mathrm{WB}$ and $\widehat{\mathrm{C}}(\beta) \subsetneq \widehat{\mathrm{C}}(\Theta(\beta))$.

QUESTIONS AND DIRECTIONS

- The "natural" bounds tend to be strong bounds.
  - How strong are the SB versions of the type-2 union theorem? (Not very - Li.)
  - And similarly for the inflation theorem? (Very - Li.)

- If we are stuck naming big classes with unions, what is the theory of unions of type-2 classes?

- What about Constable's Q2?

- What about beyond type-2?

- Complexity + realizers?