Towards New Characterizations of Small Circuit Classes via Discrete Ordinary Differential Equations (Extended Abstract)*

Melissa Antonelli[1,2], Arnaud Durand[3] and Juha Kontinen[2]

[1]Helsinki Institute for Information Technology (HIIT)
[2]University of Helsinki, Pietari Kalmin katu, 5, Helsinki, Finland
[3]Université Paris Cité, CNRS, IMJ-PRG, France

Abstract

Implicit computational complexity is an active area of theoretical computer science, which aims to provide machine-independent characterizations of relevant complexity classes. One of the seminal works in this field appeared in 1965 when Cobham introduced a function algebra closed under bounded recursion on notation (BRN) to capture $\mathbf{FP}$. Later on, several complexity classes have been characterized using *limited* recursion schemas. In this context, an original approach was recently introduced, showing that ordinary differential equations (ODEs) offer a natural tool for algorithmic design and providing a characterization of $\mathbf{FP}$ by a new ODE-schema. The overall goal of our project is precisely that of generalizing this approach to parallel computation: starting with original ODE-characterizations for the small circuit classes $\mathbf{FAC}^0$ and $\mathbf{FTC}^0$, we aim to uniformly capture the whole hierarchies $\mathbf{FAC}^k$ and $\mathbf{FNC}^k$.

Keywords

Implicit computational complexity, Parallel computation, Ordinary differential equations, Circuit complexity

1. Introduction

As computability theory investigates the limits of what is algorithmically computable, complexity theory classifies functions based on the amount of resources required by a machine to compute them. Taking a different viewpoint, implicit computational complexity aims to provide machine-independent characterizations, to offer remarkable insights on the corresponding classes and related meta-theorem in several domains, from database theory to constraint satisfaction.

One of the major approaches to computability and (implicit) complexity is constituted by the study of recursion. Groundbreaking results in this area were due to Cobham [1], who presented the first implicit function-algebra characterization for the class of poly-time computable functions $\mathbf{FP}$, and Bellantoni and Cook [2]. These works, together with other early results in recursion theory [3, 4, 5, 6], have paved the way to several generalizations based

ICTCS'24: Italian Conference on Theoretical Computer Science, September 11–13, 2024, Torino, Italy

*Extended abstract of the eponymous paper accepted at MFCS '24.

melissa.antonelli@helsinki.fi (M. Antonelli); durand@imj-prg.fr (A. Durand); juha.kontinen@helsinki.fi (J. Kontinen)
0009-0006-9072-4847 (M. Antonelli); 0000-0003-2976-7259 (A. Durand); 0000-0003-0115-5154 (J. Kontinen)

on *limited* recursion schemas, including a few capturing parallel classes [7, 8, 9, 10, 11, 12].[1] Cobham's paper has also inspired alternative (implicit) ways to capture $\mathbf{FP}$, for instance via safe recursion [2] and ramification [27]. In particular, a different descriptive approach, based on discrete ordinary differential equations (ODEs), was recently introduced in [28]. Its objective is to characterize functions computable in a given complexity class as solutions of a corresponding type of ODE. In this vein, in [28], a purely syntactic characterization of $\mathbf{FP}$ was given by linear systems of equations deriving along a logarithmically growing function. Intuitively, the latter condition controls the number of steps, while linearity controls the growth of objects generated during the computation. Recently, this approach has also been generalized to the continuous setting [29, 30].

Although small circuit classes have been characterized in multiple ways, the questions of whether they can be studied through ODE lenses and whether this would shed some new light on their features are still open. To us, these questions are *interesting* as, for a descriptive approach based on ODEs to make sense and be fruitful, it has to be able to cope with very subtle and restricted modes of computation. They are also *challenging* as even simple and useful mathematical functions may not be computable in the classes we are considering (e.g. multiplication is not in $\mathbf{FAC}^0$); consequently, tools at hand and the naturalness of the approach are drastically restricted. Our project aims to investigate these questions and to find natural ODE-oriented function algebras to capture small circuit classes. So far, we have focused on the characterization of functions computable by families of polynomial size and constant depth circuits ($\mathbf{FAC}^0$), possibly including majority gates ($\mathbf{FTC}^0$). In particular, in [31], we have captured both these classes by means of special ODE-schemas, obtained by deriving along the logarithmic function and intuitively allowing for bit shifting operations through restricted forms of linear equations. These case studies are intended as the first step towards a uniform characterization of other relevant classes in the $\mathbf{FAC}^k$ and $\mathbf{FNC}^k$ hierarchies.

2. Capturing Complexity Classes via ODEs

As anticipated, a foundational work in recursion theory was established by Cobham [1], who captures $\mathbf{FP}$ relying on the so-called bounded recursion on notation (BRN) schema:

$$f(0, \mathbf{y}) = g(\mathbf{y})$$
$$f(\mathsf{s}_i(x), \mathbf{y}) = h_i(f(x, \mathbf{y}), x, \mathbf{y}) \quad \text{for } x \neq 0, i \in \{0, 1\}$$

$$f(x, \mathbf{y}) \leq k(x, \mathbf{y}) \quad \text{for all } x, \mathbf{y}.$$

In BRN, the growth of the defined function is controlled by another function k (in $\mathbf{FP}$), while the number of induction steps is kept under control by the application of the binary successor functions $\mathsf{s}_i(x) = 2x + i$, $i \in \{0, 1\}$. However, such a schema is in a sense not fully satisfactory as it imposes an explicit bound on recursion in the form of an already known function.

[1] Other implicit characterizations based on schemas and restrictions on first-order programs have been recently introduced [13, 14, 15, 16, 17]. We thank the anonymous reviewer for pointing out this research direction. Alternative, related approaches to capture small circuit classes have also been provided in the framework of model- [18, 19, 20, 21, 9] and proof-theory [10, 11, 22, 23, 24, 25, 26].

Cobham's seminal work not only led to a variety of implicit characterizations for classes other than $\mathbf{FP}$, but also inspired alternative approaches to capture this class. Among them, the proposal by [28] has the peculiarity of neither imposing any explicit bound on the recursion schema nor assigning specific roles to variables. Indeed, it is based on special discrete ODEs, which combine two peculiar features: *deriving along specific functions*, so to control the number of computation steps, and *linearity*, namely a special syntactic form of the equation allowing to control the object size.

Recall that the *discrete derivative of* $\mathbf{f}(x)$ is defined as $\Delta\mathbf{f}(x) = \mathbf{f}(x+1) + \mathbf{f}(x)$ and that ODEs are expressions of the form:

$$\frac{\partial \mathbf{f}(x, \mathbf{y})}{\partial x} = \mathbf{h}\big(\mathbf{f}(x, \mathbf{y}), x, \mathbf{y}\big),$$

where $\frac{\partial \mathbf{f}(x,\mathbf{y})}{\partial x}$ stands for the derivative of $\mathbf{f}(x, \mathbf{y})$ considered as a function of x, for $\mathbf{y}$ fixed. When some initial value $\mathbf{f}(0, \mathbf{y}) = \mathbf{g}(\mathbf{y})$ is added, this is called *Initial Value Problem* (IVP). In addition, let $\mathsf{sg} : \mathbb{Z} \to \mathbb{Z}$ be the sign function over $\mathbb{Z}$, taking value 1 for $x > 0$ and 0 otherwise. A sg-*polynomial expression* $P(x_1, \ldots, x_h)$ is an expression built over the signature $\{+, -, \times\}$, the function sg and a set of variables $X = \{x_1, \ldots, x_h\}$, plus integer constants. A sg-polynomial expression P is said to be *essentially linear* in a set of variables x, if there exist sg-polynomial expressions Q_1 and Q_2 such that $P = Q_1 \times x + Q_2$ and, in Q_1 and Q_2, x occurs only under the scope of sg.

Definition 1 (Linear λ-ODE). *Given* $\mathbf{g} : \mathbb{N}^p \to \mathbb{Z}^d$, $\mathbf{h}, \lambda : \mathbb{N}^{p+1} \to \mathbb{Z}^d$ *and* $\mathbf{u} : \mathbb{Z} \times \mathbb{N}^{p+2} \to \mathbb{Z}^d$, *the function* $\mathbf{f} : \mathbb{N}^{p+1} \to \mathbb{Z}^d$ *is* linear λ-ODE *definable from* $\mathbf{g}$, $\mathbf{h}$ *and* $\mathbf{u}$ *if it is the solution of the IVP with initial value* $\mathbf{f}(0, \mathbf{y}) = \mathbf{g}(\mathbf{y})$ *and such that:*

$$\frac{\partial \mathbf{f}(x, \mathbf{y})}{\partial \lambda} = \mathbf{u}\big(\mathbf{f}(x, \mathbf{y}), \mathbf{h}(x, \mathbf{y}), x, \mathbf{y}\big) \tag{1}$$

with $\mathbf{u}$ *essentially linear in the list of terms* $\mathbf{f}(x, \mathbf{y})$.

For $x \neq 0$, *let* $\ell(x)$ *denote the length of* x *written in binary, i.e.* $\lceil \log_2(x+1) \rceil$, *and* $\ell(0) = 0$. *For* $\lambda = \ell$, *the schema above is called* linear length ODE, *ℓ-ODE.*

Example 1 (Function $2^{\ell(x)}$). *The function* $x \mapsto 2^{\ell(x)}$ *can be seen as the solution of the IVP with initial value* $f(0) = 1$ *and such that* $\frac{\partial f(x)}{\partial \ell} = f(x)$. *The solution of this system is of the form* $f(x) = \prod_{u=0}^{\ell(x)-1} 2 = 2^{\ell(x)}$.

One of the main results of [28] is the implicit characterization of $\mathbf{FP}$ by the algebra made of basic functions $0, 1, \pi_i^p, \ell, +, -, \times, \mathsf{sg}$ and closed under composition ($\circ$) and ℓ-ODE:

$$\mathbb{LDL} = [0, 1, \pi_i^p, \ell, +, -, \mathsf{sg}; \circ, \ell\text{-ODE}].$$

3. First Characterizations of Small Circuit Classes

So far, our investigation of parallel complexity via ODEs has focussed on the smallest classes in the hierarchies $\mathbf{FAC}^k$ and $\mathbf{FTC}^k$.

Definition 2 (Classes $\mathbf{FAC}^k$ and $\mathbf{FTC}^k$). *For $k \in \mathbb{N}$, $\mathbf{AC}^k$ (resp., $\mathbf{TC}^k$) is the class of languages recognized by a* **Dlogtime**-*uniform family of Boolean circuits (resp., circuits including majority gates) of polynomial size and depth $O((\log n)^k)$. We denote by $\mathbf{FAC}^k$ and $\mathbf{FTC}^k$ the corresponding function classes.*

In particular, in [31], we have provided the first implicit characterizations of $\mathbf{FAC}^0$ and $\mathbf{FTC}^0$ in the ODE setting. To do so, our key ingredient is the introduction of new function algebras, the defining feature of which is the presence of special ODE schemas, intuitively corresponding to left- and right-shifting.[2] The schema below intuitively corresponds to left-shifting and (possibly) adding a bit.

Definition 3 (ℓ-ODE$_2$ Schema). *Given $g : \mathbb{N}^p \to \mathbb{N}, h : \mathbb{N}^{p+1} \to \mathbb{N}$ and $k : \mathbb{N}^p \to \mathbb{N}$, the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is defined by ℓ-ODE$_2$ from g, h and k if it is the solution of the IVP with initial value $f(0, \mathbf{y}) = g(\mathbf{y})$ and such that:*

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = \left(2^{\ell(k(\mathbf{y}))} - 1\right) \times f(x, \mathbf{y}) + h(x, \mathbf{y}) \tag{2}$$

where $h(x, \mathbf{y}) \in \{0, 1\}$ and, if, for some $x, \mathbf{y}$, $h(x, \mathbf{y}) \neq 0$, then $k(\mathbf{y}) \neq 0$.

Observe that, since this schema is introduced to characterize $\mathbf{FAC}^0$, the constraint imposing $k(\mathbf{y}) \neq 0$, when there exist $x, \mathbf{y}$ such that $h(x, \mathbf{y}) = 1$, is really essential. If we omit it, ℓ-ODE$_2$ will be too strong, as able to capture binary counting (which is not in $\mathbf{FAC}^0$). However, this schema is not as weak as it may seem since, together with sg, it suffices to express bounded quantification.

The second new schema we need corresponds to the (basic) right-shifting operation.

Definition 4 (ℓ-ODE$_3$ Schema). *Given $g : \mathbb{N}^p \to \mathbb{N}$, the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is defined by ℓ-ODE$_3$ from g if it is the solution of the IVP with initial value $f(0, \mathbf{y}) = g(\mathbf{y})$ and such that:*

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = -\left\lceil \frac{f(x, \mathbf{y})}{2} \right\rceil \tag{3}$$

where $\lceil \frac{z}{2} \rceil$ is a shorthand for $z - (z \div 2)$, and $\div 2$ denotes integer division by 2.

Remarkably, the following closure property holds for both schemas.

Proposition 1. *If f is defined by ℓ-ODE$_2$ or ℓ-ODE$_3$ from functions in $\mathbf{FAC}^0$, then f is in $\mathbf{FAC}^0$ as well.*

It is relying on these schemas that we define the following ODE-style class:

$$\mathbb{ACDL} = [0, 1, \pi_i^p, \ell, +, -, \div 2, \text{sg}; \circ, \ell\text{-ODE}_2, \ell\text{-ODE}_3].$$

[2] Observe that our schemas are defined using $\times$. This is acceptable since, the "kind of multiplication" we consider to define them is limited to special cases (namely, multiplication by 2^i), which are provably computable in $\mathbf{FAC}^0$.

Notice that all its basic functions and (restricted) schemas are natural in the context of differential equations and calculus. Of course, in $\mathbb{ACDL}$, multiplication is not allowed. In addition, the λ-ODE schema is here substituted by the two schemas ℓ-ODE$_2$ and ℓ-ODE$_3$, characterized by a very limited form of "multiplication" and, as said, intuitively capturing left and right shifting.

This class is shown able to characterize $\mathbf{FAC}^0$.

Theorem 1. $\mathbf{FAC}^0 = \mathbb{ACDL}$.

In particular, our proof that $\mathbf{FAC}^0 \subseteq \mathbb{ACDL}$ is *indirect*, namely we show that any basic function and schema defining Clote's function algebra for $\mathbf{FAC}^0$ [32, 7] can be simulated in our setting by functions and schemas of $\mathbb{ACDL}$ (as done for $2^{\ell(x)}$ in Example 1).

As a byproduct, an ODE-characterization for $\mathbf{FTC}^0$ is also established (this time passing through Clote and Takeuti's function algebra [22]) by simply considering an extension of $\mathbb{ACDL}$, obtained by endowing it with the basic function $\times$:

$$\mathbb{TCDL} = [0, 1, \pi_i^p, \ell, +, -, \div 2, \times, \mathsf{sg}; \circ, \ell\text{-ODE}_2, \ell\text{-ODE}_3]$$

In addition, an alternative characterization of $\mathbf{FTC}^0$ can be introduced by substituting the ℓ-ODE$_2$ schema in the definition of $\mathbb{ACDL}$ with its more liberal version ℓ-ODE$_2^*$.

Definition 5 (ℓ-ODE$_2^*$ Schema). *Let $g : \mathbb{N}^p \to \mathbb{N}$, $h : \mathbb{N}^{p+1} \to \mathbb{N}$ and $k : \mathbb{N}^p \to \mathbb{N}$, where h takes values in $\{0, 1\}$. Then, the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is defined by ℓ-ODE$_2^*$ from g, h and k when it is the solution of the IVP with initial value $f(0, \mathbf{y}) = g(\mathbf{y})$ and such that:*

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = \left(2^{\ell(k(\mathbf{y}))} - 1\right) \times f(x, \mathbf{y}) + h(x, \mathbf{y}). \tag{4}$$

Indeed, if $\mathbf{k}(\mathbf{y}) = 0$, the binary counting function, that outputs the sum of the bits of x, can be expressed via ℓ-ODE$_2^*$. Let $\mathsf{bit}(x, y)$ be a special bit function returning 1 when the $\ell(x)^{th}$ bit of y is 1 (which can be rewritten already in $\mathbb{ACDL}$). Then, $\mathsf{bcount}(x) = f(x, x)$, where f is the solution of the IVP with initial value $f(0, \mathbf{y}) = \mathsf{bit}(0, \mathbf{y})$ and such that $\frac{\partial f(x,\mathbf{y})}{\partial \ell} = \mathsf{bit}(x, \mathbf{y})$. It is easy to see that the ℓ-ODE$_2^*$ schema is enough not only to express binary counting but, more in general, to capture majority computation.

Proposition 2. $\mathbf{FTC}^0 = \mathbb{TCDL} = [0, 1, \pi_i^p, \ell, +, -, \div 2, \mathsf{sg}; \circ, \ell\text{-ODE}_2^*, \ell\text{-ODE}_3]$

4. Future Work

We conceive our characterizations of $\mathbf{FAC}^0$ and $\mathbf{FTC}^0$ as the first step in a project aiming to capture several other relevant classes, starting with $\mathbf{FAC}^k$ and $\mathbf{FNC}^k$. Indeed, the restrictions of linear ODE schemas we have adopted are surprisingly natural, and we believe that a similar analysis would also make it possible to capture computation corresponding to k-BRN and w-BRN [32, 7, 8]. We are currently exploring this promising path, to obtain a uniform characterization of the (entire) mentioned hierarchies through the prism of ODEs. Another challenging direction for future research would be to develop logical and proof-theoretical counterparts to ODE-style algebras, for instance by introducing *natural* rule systems (oriented by the ODE design) to syntactically characterize the corresponding classes.

Acknowledgments

The first two authors thank Maupertuis Program, on behalf of the Institut Français de Finlande, the Embassy of France in Finland, the French Ministry of Higher Education and Research and the Finnish Society of Sciences and Letters, for financially supporting their research. They also thank the ANR Project Difference. The first author is grateful to HIIT for supporting her work since 2023.

References

[1] A. Cobham, The intrinsic computational difficulty of functions, in: Logic, Methodology and Philosophy of Science: Proc. 1964 International Congress, 1965, pp. 24–30.

[2] S. Bellantoni, S. Cook, A new recursion-theoretic characterization of poly-time functions, Comput. Complex. 2 (1992) 97–110.

[3] A. Grzegorczyk, Some classes of recursive functions, Rozptawy Matematyczne 4 (1953).

[4] J. Bennett, On Spectra, Ph.D. thesis, Princeton University, 1962.

[5] R. Ritchie, Classes and predictably computable functions, in: Trans. Am. Math. Soc., volume 106, 1963, pp. 139–173.

[6] J. Lind, Computing in Logarithmic Space, Ph.D. thesis, Massachusetts Institute of Technology, 1974.

[7] P. Clote, Sequential, machine-independent characterizations of the parallel complexity classes AlogTIME, AC^k, NC^k and NC, in: Progress in Computer Science and Applied Logic, Birkhäuser, 1990.

[8] P. Clote, On polynomial size Frege proofs of certain combinatorial principles, in: P. Clote, K. J. (Eds.), Arithmetic, Proof Theory, and Computational Complexity, Clarendon Press, 1993, pp. 166–184.

[9] C. K.J., C. Laflamme, An algebra and a logic for NC^1, Inf. Comput. 87 (1990) 240–262.

[10] B. Allen, Arithmetizing uniform NC, Ann. Pure Appl. Logic 53 (1991) 1–50.

[11] P. Clote, G. Takeuti, Bounded arithmetic NC, ALogTime, L and NL, Ann. Pure and Appl. Logic 56 (1992) 73–117.

[12] G. Bonfante, R. Kahle, J.-Y. Marion, I. Oitavem, Two function algebras defining functions in NC^k boolean circuits, Inf. Comput. (248) 82–103.

[13] G. Bonfante, J.-Y. Marion, R. Péchoux, A characterization of alternating log time by first order funcitonal programs, in: Proc. LPAR, 2006, pp. 90–104.

[14] D. Leivant, A characterization of NC by tree recurrence, in: Proc. FOCS, 1998, pp. 716–724.

[15] D. Leivant, J.-Y. Marion, A characterization of alternating log time by ramified recurrence, TCS 236 (2000) 193–208.

[16] J.-Y. Marion, R. Péchoux, A characterization of NC^k, in: Proc. TAMC, 2008, pp. 136–147.

[17] K.-H. Niggl, H. Wunderlich, Implicit characterizations of FPT and NC revisited, The Journal of Logic and Algebraic Programming 79 (2010) 47–60.

[18] N. Immerman, Languages that capture complexity classes, SIAM J. Comput. 16 (1987) 760–778.

[19] D. Barrington, N. Immerman, H. Staubing, On uniformity within NC^1, J. of Comput. and Syst. Sc. 41 (1990) 274–306.
[20] Y. Gurevich, H. Lewis, A logic for constant-depth circuit, Inf. Control 61 (1984) 65–74.
[21] S. Lindell, A purely logical characterization of circuit uniformity, in: 7th Structure in Complexity Theory Conf., 1992, pp. 185–192.
[22] P. Clote, G. Takeuti, First order bounded arithmetic and small complexity classes, in: Feasible Mathematics II, Clote, P.G. and Remmell, J.B., 1995, pp. 154–218.
[23] J. Johannsen, A bounded arithmetic theory for constant depth threshold circuit, in: P. Hájek (Ed.), GÖDEL '96: Logical foundations of mathematics, computer science and physics, 1996, pp. 224–234.
[24] S. Cook, P. Nguyen, Theories for TC^0 and other small circuit classes, Log. Meth. Comput. Sci. 2 (2006).
[25] T. Arai, A bounded arithmetic AID for Frege systems, Ann. Pure Appl. Logic 103 (2000) 155–199.
[26] S. Cook, T. Morioka, Quantified propositional calculus and second-order theory for NC^1, Arch. Math. Logic 44 (2005) 711–749.
[27] D. Leivant, Y.-Y. Marion, Lambda calculus characterizations of poly-time, Fundam. Inform. 19 (1993) 167–184.
[28] O. Bournez, A. Durand, A characterization of functions over the integers computable in polynomial time using discrete differential equations, Comput. Complex. 32 (2023).
[29] M. Blanc, O. Bournez, A characterization of polynomial time computable functions from the integers to the reals using discrete ordinary differential equations, in: Proc. MCU, 2022, pp. 58–74.
[30] M. Blanc, O. Bournez, A characterization of functions computable in polynomial time and space over the reals with discrete ordinary differential quations, in: Proc. MFCS, 2023, pp. 21:1–21:15.
[31] M. Antonelli, A. Durand, J. Kontinen, A new characterization of FAC^0 via discrete ordinary differential equations, in: Proc. MFCS, 2024.
[32] P. Clote, A sequential characterization of the parallel complexity class NC, Technical Report, Boston College, 1988.