Recursion and Proof Theoretical Characterizations of Small Circuit Classes with Modulo Counting via Discrete Differential Equations

Melissa Antonelli
CFvW, University of Tübingen, Germany

Arnaud Durand
Université Paris Cité and Sorbonne Université, CNRS, IMJ-PRG, F-75013 Paris, France

Rui Li
Institut Polytechnique de Paris, Ecole Polytechnique, LIX and Université Paris Cité, IMJ-PRG, CNRS, France

Abstract

The paper proposes an implicit (i.e., machine-independent) complexity approach to studying computation by polynomial-size, constant-depth circuits with gates counting modulo a constant through the lens of discrete ordinary differential equations (ODEs). So far, recursion-theoretic characterizations have been provided for functions computed by circuits of constant depth, including gates counting modulo 2 and 6 only (i.e., for the classes $\mathbf{FAC}^0[\mathbf{2}]$ and $\mathbf{FAC}^0[\mathbf{6}]$, resp.). In this paper, it is shown that considering ODE schemas, rather than bounded recursion, allows for a more fine-grained analysis, leading to (uniform) characterizations for *all* classes $\mathbf{FAC}^0[\mathbf{n}]$ ($n \in \mathbb{N}$), i.e. functions computed by circuits including counting modulo n gates. Inspired by the syntactic form of the ODE schemas, we go further in this direction and present first-order bounded theories for capturing provably total functions in each of these classes.

2012 ACM Subject Classification Theory of computation → Computability; Theory of computation → Complexity theory and logic; Theory of computation → Circuit complexity

Keywords and phrases Implicit complexity, circuit complexity, small circuit classes with counting, discrete ODEs, recursion theory, bounded arithmetic

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.162

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Funding The first author thanks CFvW for funding her research since 2026. The first and second authors acknowledge support from the CARACAL project, and all three authors acknowledge support from the ANR grant "Difference" (ANR-20-CE48-0002).

1 Introduction

Circuit complexity has been studied extensively from multiple angles in recent decades. Just to mention a concrete objective, understanding the computational power of polynomial-size, $\log^k n$ depth circuit classes, $\mathbf{AC}^k$, and their bounded fan-in counterpart, $\mathbf{NC}^k$, has received significant attention. Although some of the few known lower bounds have been established precisely in relation to Boolean and arithmetic circuits, many *fundamental* questions in this field are still open; for example, the separation between levels of the mentioned hierarchies has currently been established only for levels 0 and 1. Indeed, from the celebrated proofs of [20, 29], it is known that Parity, the set of words having an even number of 1's, cannot be decided by polynomial-size, constant-depth circuits, i.e., in $\mathbf{AC}^{\mathbf{0}}$. This result illustrates the (almost) complete inability of constant-depth circuits to count, motivating the study of their extension via counting modulo $n \in \mathbb{N}$ or majority gates, which gave rise to the

classes $\mathbf{AC}^0[n]$ and $\mathbf{TC}^0$, resp. In this setting, the parity lower bound directly implies that $\mathbf{AC}^0 \subsetneq \mathbf{AC}^0[\mathbf{2}]$. More recently, related questions have gained interest even in less traditional fields of computer science; for instance, driven by the link between circuit resources and neural networks [28], as early as the 1990s, the expressive power of feed-forward and graph networks was related to threshold Boolean circuits [2, 27] and constant-depth arithmetic circuits over the reals [5], resp.

An alternative approach to studying these foundational questions is the investigation of circuit expressive power from machine-independent perspectives; namely, through the prism of logic (especially finite model theory [19, 22]), proof complexity, and recursion theory. The contributions of this paper fall precisely within this line of research, starting from a newly proposed perspective in the latter area [9]. Generally speaking, in recursion theory capturing complexity classes has been done by considering (closure under) restricted forms of primitive recursion. This is, for example, the idea behind Cobham's characterization of poly-time computable functions (**FP**) via the so-called *bounded recursion on notation* (BRN) schemas [16]. Following this approach, in a series of works [12, 13, 14], Clote introduced the algebra $\mathcal{A}_0$ to capture functions in the log-time hierarchy, equivalent to $\mathbf{FAC}^0$ (the function analogue of $\mathbf{AC}^0$), using as a main tool a restricted form of (bounded) recursion, called *concatenation recursion on notation* (CRN). This result was extended to higher levels in the hierarchies **FAC** and **FNC** (the function versions of **AC** and **NC**). Around the same time, an alternative algebra (and logic), based on a schema called *upward tree recursion*, was introduced in [17] to characterize $\mathbf{FNC}^1$, while in [1], an original recursion-theoretic characterization (and arithmetic *à la Buss*) was presented to capture **FNC**. Other small circuit classes, including $\mathbf{FTC}^0$, the class of functions computed by polynomial-size, constant-depth unbounded fan-in circuits with threshold gates, were later considered by Clote and Takeuti, who also introduced first-order bounded theories for some of them [15]. This approach allowed for the characterization of the classes $\mathbf{FAC}^0[\mathbf{2}]$ and $\mathbf{FAC}^0[\mathbf{6}]$, but it failed to extend to other modulo classes.

In the broader context of implicit characterizations of complexity, even beyond circuit-based models, a new, machine-independent approach via differential equations has recently emerged and has been shown to be suitable for both discrete [9] and continuous settings [6, 7, 8]. In the discrete case, one can describe a function by an ordinary differential equation (ODE) it has to satisfy. Informally, given three functions g, h and t, a function f is said to be the solution of a (generalized) ODE (a.k.a., initial value problem) if it satisfies $\frac{\partial f(x,\mathbf{y})}{\partial t(x)} = h\big(x, \mathbf{y}, f(x,\mathbf{y})\big)$ together with some initial condition $f(0,\mathbf{y}) = g(\mathbf{y})$. Recall that here $\frac{\partial f(x,\mathbf{y})}{\partial t(x)}$ stands for the derivative of $f(x,\mathbf{y})$ along the values of $t(x)$ (the simpler case of the *discrete* derivative of $f(x,\mathbf{y})$ on the variable x is simply $\frac{\partial f(x,\mathbf{y})}{\partial x} = f(x+1,\mathbf{y}) - f(x,\mathbf{y})$). Intuitively, this new framework to define functions offers *three different levels of control*:

- the choice of t, the function to derive along, influences the *growth speed*, i.e. the number of steps, or (in parallel context) the *width* of computation;
- the function h (and g) controls the *growth* of intermediate values and the *feasibility of iterating* elementary computation steps;
- constraints on calls to f in h, e.g. allowing "unrestricted" calls to f or variously limiting them, influence *memory* use or *parallelization depth*.

This framework appears to be significantly more flexible and expressive than standard limited recursion, imposing bounds on the value h. Additionally, unlike the latter, the ODE approach offers a way to talk about discrete and continuous computations in a relatively similar way, often focusing on similar syntactical restrictions [7, 8]. Such an approach was initially used to establish a correspondence between functions computable in poly-time and

solutions of specific ODEs, whose defining equations are of a special (essentially) linear form and obtained by deriving along functions of specific growth rate [10] (Sec. 2.1). In this vein, previous investigation on small circuit classes has already demonstrated a, quite striking, correspondence between simple syntactical constraints on these ODE schemas and computation performed by circuits, allowing for a fine-grained definition of the corresponding classes. However, although different circuit classes have been considered in [3, 4], the study of those that include counting gates has been developed only partially and has been so far restricted to $\mathbf{FTC}^0$ and $\mathbf{FAC}^0[\mathbf{2}]$.

In this paper, we push this investigation further by characterizing classes allowing various forms of counting, which have never been captured via algebras (or first-order bounded arithmetic) before. Concretely we introduce *natural* and *uniform* function algebras, for $\mathbf{FAC}^0[\mathbf{n}]$ (being $n \in \mathbb{N}$), the class of functions computable by polynomial-size and constant-depth circuits with counting modulo n gates. The key ingredient is an ODE-based recursion schema, whose evaluation essentially requires modulo counting. Despite its apparent simplicity, this change in perspective allows for a significant advance over the existing literature since, to the best of our knowledge, so far implicit (recursion-theoretic) characterizations have been provided only for specific classes, namely $\mathbf{FAC}^0[\mathbf{2}]$ and $\mathbf{FAC}^0[\mathbf{6}]$ [15]. Additionally, inspired by these schemas, we introduce new bounded arithmetic theories to syntactically capture these same classes. Again, this goes beyond what is currently known (except for $n = 2$ and $n = 6$). More generally, the investigation presented here aims to clarify the connection between features of ODE schemas and resources in circuit computation. This mirrors insights obtained in descriptive complexity, which provides a correspondence between resources in circuits or parallel machines and expressive power of logical features (like the number of variables) [24, 25], but extends to broader forms of computation, for example including (modulo) counting, which have never been fully addressed in other frameworks. As an overall goal, this work aims to provide an original perspective for investigating core questions in (circuit) complexity, such as class separations and lower bounds.

Structure of the paper. In Sec. 2, we introduce notational conventions, basic notions and methodologies needed for the rest of the paper. In Sec. 3, we present new, ODE-based function algebras capturing poly-size and constant-depth circuit classes, including modulo 2 (Sec. 3.1), more general modulo (arbitrary) n (Sec. 3.2), and majority gates (Sec. 3.3). A formal comparison with the few existing characterizations is also presented. Moving to proof theory, in Sec. 4, we show the technical advantages of relying on these function algebras to deal with a known first-order bounded arithmetic for $\mathbf{FAC}^0$ (Sec. 4.1) and present alternative original theories inspired by the ODE-based schemas to capture provably total functions in $\mathbf{FAC}^0[\mathbf{n}]$ (Sec. 4.2 and 4.3). We conclude by pointing to future directions of research (Sec. 5). Due to space constraints, most proofs are omitted but some are available in the Appendix.

2 Preliminaries

▶ **Notation 1.** *We will adopt the following notational conventions:* $s(x) = x+1$ *for successor;* $s_i(x) = 2x + i$ $(i \in \{0, 1\})$ *for binary successors;* $x\#y = 2^{\ell(y)}x + y$ *for the smash function;* $\div n$ *for integer division by* $n \in \mathbb{N}^*$*, i.e.,* $x \div n = \lfloor \frac{x}{n} \rfloor$*, with* $x \in \mathbb{Z}$*;* $\pi^k_i(x_1, \ldots, x_k) = x_i$ *for projection;* $\ell(x)$ *for the length function which, given an input* x*, returns its length written in binary, i.e.,* $\ell(x) = \lceil \log_2(x+1) \rceil$ *and s.t., for convenience,* $\ell(0) = 0$*;* $\mathsf{BIT}(i, x) = \mathsf{mod2}\big(\lfloor \frac{x}{2^i} \rfloor\big)$*, where* $\mathsf{mod2}(y) = x - 2 \times \lfloor \frac{x}{2} \rfloor$*, for the bit function returning the value of the* i^{th} *bit in the binary representation of* x*;* $\mathsf{sg} : \mathbb{Z} \to \mathbb{Z}$ *for the sign function s.t.* $\mathsf{sg}(x) = 1$ *if* $x > 0$ *and* $\mathsf{sg}(x) = 0$ *otherwise;* $\mathsf{cosg}(x) = 1 - \mathsf{sg}(x)$*. For* $x \in \mathbb{N}^+$*,* $\alpha(x) = 2^u - 1$ *denotes the greatest integer whose length is* u*.*

2.1 A gentle introduction to discrete ODEs (in complexity)

Difference calculus, the discrete analogue of differential calculus, is an old field of mathematics [21, 26]. Recall that the *discrete derivative of* $f(x)$ is defined as $\Delta f(x) = f(x+1) - f(x)$ and that ODEs are expressions of the form: $\frac{\partial f(x,\mathbf{y})}{\partial x} = h\big(x, \mathbf{y}, f(x,\mathbf{y})\big)$ where $\frac{\partial f(x,\mathbf{y})}{\partial x}$ stands for the derivative of $f(x,\mathbf{y})$ considered as a function of x for a fixed value of $\mathbf{y}$. When some initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ is added, this form an Initial Value Problem (IVP).

As anticipated, this framework is shown to offer natural ways to capture primitive recursion and function classes. Specifically, in order to deal with complexity, this picture must be endowed with new notions, starting with that of derivation *along a function*:

▶ **Definition 2** (λ-ODE). *Let* $f, \lambda : \mathbb{N}^p \to \mathbb{Z}$ *and* $h : \mathbb{N}^{p+1} \to \mathbb{Z}$. *Then,* $\frac{\partial f(x,\mathbf{y})}{\partial \lambda} = \frac{\partial f(x,\mathbf{y})}{\partial \lambda(x,\mathbf{y})} = h\big(x, \mathbf{y}, f(x,\mathbf{y})\big)$ *is a synonym of* $f(x+1,\mathbf{y}) = f(x,\mathbf{y}) + \big(\lambda(x+1,\mathbf{y}) - \lambda(x,\mathbf{y})\big) \times h\big(x, \mathbf{y}, f(x,\mathbf{y})\big)$ $= f(x,\mathbf{y}) + \frac{\partial \lambda(x,\mathbf{y})}{\partial x} \times h\big(x, \mathbf{y}, f(x,\mathbf{y})\big)$. *For* $\lambda = \ell$, *we call it* length-ODE *or* ℓ-ODE.

Intuitively, one of the key properties of the λ-ODE schema is its dependence on the number of distinct values of λ, i.e., the value of f changes only when the value of λ does. In this way, the growth rate of λ determines the number of steps needed to compute the value of f. Computation of solutions of λ-ODEs has been fully described in [10]. Here, we focus on the special case of $\lambda = \ell$, which is particularly interesting for our characterizations since the value of $\ell(x)$ changes, increasing by 1, only when x goes from $2^t - 1$ to 2^t (for $t \geq 1$). Setting $h\big(\alpha(-1), \mathbf{y}, \cdot\big) = f(0,\mathbf{y})$, it is shown by induction that $f(x,\mathbf{y}) = \sum_{u=-1}^{\ell(x)-1} h\big(\alpha(u), \mathbf{y}, f(\alpha(u),\mathbf{y})\big)$.

Obviously, the choice of the function h also influences the complexity of the computation. Here, we will mostly deal with (limited) sg-polynomial expressions, where a *(resp. limited)* sg*-polynomial expression* is an expression over the signature $\{+, -, \div 2, \times\}$ (resp. $\{+, -, \div 2\}$) on the set of variables/terms $X = \{x_1, \ldots, x_h\}$ and integer constants (see [3]). The degree of a sg-polynomial expression $\deg(P)$ is inductively defined so that $\deg(P) = 0$ for P being a constant, a variable or a signed expression of the form $\mathsf{sg}(Q)$, $\deg(\mathbf{x}, Q \div 2) = \deg(\mathbf{x}, Q)$, $\deg(\mathbf{x}, Q * R)$ $= \max\{\deg(\mathbf{x}, Q), \deg(\mathbf{x}, R)\}$, for $* \in \{+, -\}$, and $\deg(\mathbf{x}, Q \times R) = \deg(\mathbf{x}, Q) + \deg(\mathbf{x}, R)$. A sg-polynomial expression P is said to be *essentially constant in a set of variables* $\mathbf{x}$ when $\deg(\mathbf{x}, P) = 0$ and *essentially linear in* $\mathbf{x}$, when $\deg(\mathbf{x}, P) = 1$. The concept of *linearity* allows us to control the growth of functions defined by ODEs.

▶ **Definition 3** (Linear λ-ODE). *Given* $g : \mathbb{N}^p \to \mathbb{N}$, $h, \lambda : \mathbb{N}^{p+1} \to \mathbb{Z}$ *and a* sg*-polynomial expression* P, *the function* $f : \mathbb{N}^{p+1} \to \mathbb{Z}$ *is* linear λ-ODE *definable from* g, h *and* P *if it is the solution of the IVP with initial value* $f(0,\mathbf{y}) = g(\mathbf{y})$ *and s.t.* $\frac{\partial f(x,\mathbf{y})}{\partial \lambda} = P\big(x, \mathbf{y}, f(x,\mathbf{y}), h(x,\mathbf{y})\big)$ *where* P *is essentially linear in* $f(x,\mathbf{y})$. *For* $\lambda = \ell$, *such schema is called* linear length ODE.

Stated otherwise, if P is *essentially linear in* $f(x,\mathbf{y})$, there exist A and B, which are essentially constants in $f(x,\mathbf{y})$, s.t. $f(0,\mathbf{y}) = g(\mathbf{y})$ and $\frac{\partial f(x,\mathbf{y})}{\partial \ell} = A(x,\mathbf{y},h(x,\mathbf{y}),f(x,\mathbf{y})) \times f(x,\mathbf{y}) + B(x,\mathbf{y},h(x,\mathbf{y}),f(x,\mathbf{y}))$. Then, for all x and $\mathbf{y}$, $f(x,\mathbf{y}) = \sum_{u=-1}^{\ell(x)-1} \Big(\prod_{t=u+1}^{\ell(x)-1} \Big(1 + A\Big(\alpha(t), \mathbf{y}, h(\alpha(t),\mathbf{y}), f(\alpha(t),\mathbf{y})\Big)\Big)\Big) \times B\big(\alpha(u), \mathbf{y}, h(\alpha(u),\mathbf{y}), f(\alpha(u),\mathbf{y})\big)$, with the convention that $\prod_x^{x-1} \kappa(x) = 1$ and $B\big(\alpha(-1), \mathbf{y}, \cdot, \cdot\big) = f(0,\mathbf{y})$. One of the main results of [9] 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.

2.2 Function algebras for small circuit classes

Circuit complexity through the lens of bounded recursion. We assume that the reader is familiar with the standard notions of small circuit classes (see [30]). Here we briefly recall main recursion theoretic characterizations offered in the literature. To the best of our knowledge, when considering small circuit classes including gates counting modulo $n \in \mathbb{N}$, only a few implicit characterizations have been provided so far: function algebras and first-order bounded arithmetics by Clote and Takeuti [15], a second-order family of theories by Cook and Nguyen [18] and Johannsen's theory for $\mathbf{FAC}^0[\mathbf{2}]$. Since we will often compare our results with those in [13, 15], we briefly recall them.

▶ **Definition 4** (CRN, k-BRN). *Let $i \in \{0,1\}$, $g : \mathbb{N}^p \to \mathbb{N}$ and $h_i : \mathbb{N}^{p+2} \to \mathbb{N}$. A function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is said to be defined from g, h_0 and h_1 by*

- Concatenation Recursion on Notation (CRN) *if for all x and $\mathbf{y}$: $f(0,\mathbf{y}) = g(\mathbf{y})$ and $f(s_i(x),\mathbf{y}) = s_{h_i(x,\mathbf{y})}(f(x,\mathbf{y}))$, where h_i takes values in $\{0,1\}$*
- k-Bounded Recursion on Notation (k-BRN) *if for all x and $\mathbf{y}$: $f(0,\mathbf{y}) = g(\mathbf{y})$ and $f(s_i(x),\mathbf{y}) = h_i(x,\mathbf{y},f(x,\mathbf{y}))$, with $f(x,\mathbf{y}) \leq k$ for a constant k.*

In [13], it is proved that the function algebra $\mathcal{A}_0 = [0, \pi_i^p, s_0, s_1, \ell, \mathrm{BIT}, \#; \circ, \mathrm{CRN}]$ captures $\mathbf{FAC}^0$. Enriching $\mathcal{A}_0$ with 1-BRN gives a characterization for $\mathbf{FAC}^0[\mathbf{2}]$, with 2-BRN captures $\mathbf{FAC}^0[\mathbf{6}]$ and for k-BRN, with $k \geq 4$, characterizes $\mathbf{FNC}^1$ [15, Th. 2.2, 2.5]. Stated otherwise, by restricting the range of the schema characterizing $\mathbf{FNC}^1$, i.e. k-BRN, two specific counting classes are captured but no generalization to $\mathbf{FAC}^0[\mathbf{n}]$ for $n \notin \{2,6\}$ can be expected. A function algebra for $\mathbf{FTC}^0$ is there defined by simply endowing $\mathcal{A}_0$ with multiplication. Also, first-order bounded theories are introduced to syntactically characterize all these classes (see Sec. 4).

Circuit complexity through the lens of ODEs. We present an overview of the characterizations of small circuit classes (without modulo counting gates) obtained relying on (ℓ-)ODE schemas. Although different classes have been considered in [3, 4], investigation of those that include counting gates has been developed only partially. In particular, for counting modulo 2, an algebra was defined in terms of non-strict schemas, which, as we shall see, captures constant-depth computation somewhat indirectly.

▶ **Notation 5** (Strict and simple schemas). *An equation or, by extension, an ODE schema is said to be* strict *when it does not include any call to f in A and B (not even under the scope of the sign function). For the sake of clarity, we will use $k(x,\mathbf{y})$ (resp., $K(x,\mathbf{y},h(x,\mathbf{y}),f(x,\mathbf{y}))$) for functions (resp. expressions) with restricted range, typically in $\{0,1\}$.*

If not stated otherwise, the expressions used in schemas characterizing $\mathbf{FAC}^0$ and $\mathbf{FAC}^0[\mathbf{n}]$ are *limited* sg-polynomial expressions (multiplication is not computable in these classes).

▶ **Notation 6.** *Let $\mathcal{B}_0$ denote the set of basic functions $0, 1, \mathsf{sg}, \ell, +, -, \div 2, \#, \mathsf{BIT}, \pi_i^p$.*

The function algebra capturing $\mathbf{FAC}^0$ is obtained by the strict schema below [3, 4].

▶ **Definition 7** (ℓ-ODE_1 schema). *Given $g : \mathbb{N}^p \to \mathbb{N}$ and $k : \mathbb{N}^{p+1} \to \{0,1\}$, the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is obtained by ℓ-ODE_1 from g and k, if it is the solution of the IVP with initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ and s.t. $\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$.*

Computation through ℓ-ODE_1 intuitively corresponds to iteratively left-shifting (the binary representation) of a number, each time possibly adding 1 in the last position. Due to it, the algebra $\mathbb{ACDL} = [\mathcal{B}_0; \circ, \ell\text{-}\mathrm{ODE}_1]$ is introduced and proved to correspond to $\mathbf{FAC}^0$.

▶ Remark 8 (Bounded quantification and minimization). Sharply bounded quantification and minimization can be naturally rewritten in $\mathbb{ACDL}$. Let $R \subseteq \mathbb{N}^{p+1}$ and h_R be its characteristic function. For all x and $\mathbf{y}$, it holds that $(\exists z \leq \ell(x))R(z, \mathbf{y}) = \mathsf{sg}(f_\exists(x, \mathbf{y}))$, where $f_\exists$ is defined as the solution of the IVP with initial value $f_\exists(0, \mathbf{y}) = h_{R_0}(\mathbf{y})$ (being $h_{R_0}(\mathbf{y}) = 0$ if $R(0, \mathbf{y}) = 0$ and $h_{R_0}(\mathbf{y}) = 1$ if $R(0, \mathbf{y}) = 1$) and such that:

$$\frac{\partial f_\exists(x, \mathbf{y})}{\partial \ell} = f_\exists(x, \mathbf{y}) + h_R(\ell(x) + 1, \mathbf{y}).$$

Clearly, this is an instance of ℓ-ODE$_1$. Intuitively, $f_\exists(x, \mathbf{y}) \neq 0$ when, for some z smaller than $\ell(x)$, $R(z, \mathbf{y})$ is satisfied (i.e., $h_R(z, \mathbf{y}) = 1$): if such instance exists, our bounded search ends with a positive answer.

In order to express universally sharply bounded quantification in $\mathbb{ACDL}$ we consider $(\forall z \leq \ell(x))R(z, \mathbf{y}) = \mathsf{cosg}(f_\forall(x, \mathbf{y}))$ for f such that:

$$\begin{aligned} f(0, \mathbf{y}) &= \mathsf{cosg}(h_{R_0}(\mathbf{y})) \\ \frac{\partial f_\forall(x, \mathbf{y})}{\partial \ell} &= f_\forall(x, \mathbf{y}) + \mathsf{cosg}(h_R(\ell(x) + 1), \mathbf{y}) \end{aligned}$$

Finally, the sharply bounded μ-operator, $\mu x.R(x, \mathbf{y})$, returning the smallest $z \leq \ell(x)$ such that $R(z, \mathbf{y})$ holds, can be expressed in $\mathbb{ACDL}$ by considering $\mathsf{mu}_R(x, \mathbf{y}) = \ell(x) - \ell(f_\mu(x, \mathbf{y}))$, where f_μ is the solution of the IVP with initial value $f_\mu(0, \mathbf{y}) = 0$ and such that:

$$\frac{\partial f_\mu(x, \mathbf{y})}{\partial \ell} = f_\mu(x, \mathbf{y}) + \exists i \leq \ell(x + 1)\big(h_R(i, \mathbf{y}) = 1\big).$$

Since, as seen, existentially sharply bounded search can be expressed in $\mathbb{ACDL}$, this is an instance of ℓ-ODE$_1$.

To capture computation with counting gates or that goes beyond constant depth, one has to introduce more general schemas.

▶ **Definition 9.** *Given $g : \mathbb{N}^p \to \mathbb{N}$ and $h : \mathbb{N}^{p+1} \to \mathbb{N}$, the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ defined as the solution of the IVP with initial value $f(0, \mathbf{y}) = g(\mathbf{y})$ and such that*

- $\frac{\partial f(x, \mathbf{y})}{\partial \ell} = A(x, \mathbf{y}, h(x, \mathbf{y})) \times f(x, \mathbf{y}) + B(x, \mathbf{y}, h(x, \mathbf{y}))$ *where $A, B \geq 1$ are* sg*-polynomial expressions, is said to be defined by* ℓ-pODE*;*
- $\frac{\partial f(x, \mathbf{y})}{\partial \ell} = -f(x, \mathbf{y}) + K\big(x, \mathbf{y}, h(x, \mathbf{y}), f(x, \mathbf{y})\big)$ *where $K \in \{0, 1\}$ is a limited* sg*-polynomial expression and f occurs in K only under the scope of the sign function, is said to be defined by* ℓ-b$_0$ODE*;*
- $\frac{\partial f(x, \mathbf{y})}{\partial \ell} = -f(x, \mathbf{y}) + B\big(x, \mathbf{y}, h(x, \mathbf{y}), f(x, \mathbf{y})\big)$ *, where $B \geq 0$ is a* sg*-polynomial expression, and f occurs in B only in expressions of the form* $\mathsf{sg}(f - c)$*, for a constant c, is said to be defined by* ℓ-bODE*.*

Notably, ℓ-pODE is the strict counterpart of the ℓ-ODE schema introduced in [10] to capture **FP**. In contrast, ℓ-b$_0$ODE and ℓ-bODE are non-strict schemas, allowing for (restricted) calls to f. By simply endowing $\mathbb{ACDL}$ with these schemas, it is proved that [3, 4]:

$$\begin{aligned} \mathbf{FAC}^0[\mathbf{2}] &= [\mathcal{B}_0; \circ, \ell\text{-ODE}_1, \ell\text{-b}_0\text{ODE}] \\ \mathbf{FTC}^0 &= [\mathcal{B}_0; \circ, \ell\text{-pODE}] \\ \mathbf{FNC}^1 &= [\mathcal{B}_0; \circ, \ell\text{-pODE}, \ell\text{-bODE}]. \end{aligned}$$

3 Characterizing small circuit classes with modulo counting gates

We extend the investigation started in [3, 4] by providing ODE-based function algebra characterizations for all $\mathbf{FAC}^0[\mathbf{n}]$ classes. Indeed, although ℓ-b$_0$ODE, which intuitively corresponds to 1-BRN (i.e. to $\mathbf{FAC}^0[\mathbf{2}]$), offers the advantage of being related to the (obviously, non-strict) schema characterizing $\mathbf{FNC}^1$, it is not naturally generalizable to capture any $\mathbf{FAC}^0[\mathbf{n}]$. Instead we introduce simple schemas that naturally mirror modulo counting, allowing us to obtain uniform characterizations for all these classes. Consequently, our proofs are entirely self-contained and offer a marked simplification over existing ones (for $\mathbf{FAC}^0[\mathbf{2}]$ and $\mathbf{FAC}^0[\mathbf{6}]$) [15]. In Sec. 3.1, we focus on $\mathbf{FAC}^0[\mathbf{2}]$ as a case study, which, in Sec. 3.2, is generalized by introducing a family of homogeneous schemas offering the very first characterizations for all $\mathbf{FAC}^0[\mathbf{n}]$. As a byproduct, $\mathbf{FTC}^0$ is captured in a new way (Sec. 3.3).

3.1 A new characterization for $\mathbf{FAC}^0[2]$ by strict schemas

We start by introducing a strict schema defined as a special, limited case of ℓ-ODE.

▶ **Definition 10** (Schema ℓ-2ODE). *Let $g : \mathbb{N}^p \to \{0,1\}$ and $k : \mathbb{N}^{p+1} \to \{0,1\}$. Then $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is defined by ℓ-2ODE from g and k if it is the solution of the IVP s.t. $f(0,\mathbf{y}) = g(\mathbf{y})$ and $\frac{\partial f(x,\mathbf{y})}{\partial \ell} = -2 \times k(x,\mathbf{y}) \times f(x,\mathbf{y}) + k(x,\mathbf{y})$.*

Clearly, if $k(x,\mathbf{y}) = 0$, $\frac{\partial f(x,\mathbf{y})}{\partial \ell} = 0$, and if $k(x,y) = 1$, $\frac{\partial f(x,\mathbf{y})}{\partial \ell} = -2f(x,\mathbf{y}) + 1$. Hence, $f(x,\mathbf{y})$ takes values in $\{0,1\}$, that, when k is positive, switch from 0 to 1, and vice versa. As desired the computation through ℓ-2ODE is in $\mathbf{FAC}^0[\mathbf{2}]$.

▶ **Proposition 11.** *If f is defined by ℓ-2ODE from functions in $\mathbf{FAC}^0[\mathbf{2}]$, f is in $\mathbf{FAC}^0[\mathbf{2}]$.*

Proof sketch. By Def. 10, $f(x,\mathbf{y}) = \sum_{u=-1}^{\ell(x)-1} k(\alpha(u),\mathbf{y}) \times (-1)^{\#\{t \in [u+1,\ell(x)-1] : k(\alpha(t),\mathbf{y})=1\}}$. All terms in which $k(\alpha(u),\mathbf{y}) = 0$ vanish from the sum. Let $\{u_1,\dots,u_a\} = \{t \in [-1,\ell(x)-1] : k(\alpha(t),\mathbf{y}) = 1\}$. Then, $f(x,\mathbf{y}) = \sum_{i=0}^{a}(-1)^{a-i}$, which is 0 if a is odd and 1 otherwise. ◀

Remarkably, this schema also offers a natural tool to compute parity, for example, by checking whether the number of 1s in the binary representation of its input is even or odd. This function is especially crucial, as it provides one of the few known separation results between small circuit classes (namely, separating $\mathbf{FAC}^0$ and $\mathbf{FAC}^0[\mathbf{2}]$).

▶ Remark 12 (Counting modulo 2). Computation performed by Mod 2 gates can be simulated via ℓ-2ODE by considering the special case of $g(x,y) = 0$ and $k(x,\mathbf{y}) = \mathsf{BIT}(\ell(x),y)$, i.e. by considering the solution of the IVP below:

$$\mathsf{cmod2}(0,y) = 0$$
$$\frac{\partial \mathsf{cmod2}(x,y)}{\partial \ell} = -2\mathsf{BIT}(\ell(x),y) \times \mathsf{cmod2}(x,y) + \mathsf{BIT}(\ell(x),y).$$

Then, the count of input bits modulo 2 is obtained by computing $\mathsf{cmod2}(x,x)$.

▶ **Example 13.** Let $n = 2$, so that $z = z_1 z_0$. Then,

$$\mathsf{cmod2}(0,z) = 0$$
$$\mathsf{cmod2}(1,z) = \mathsf{cmod2}(0,z) - 2\mathsf{BIT}(0,z) \times \mathsf{cmod2}(0,z) + \mathsf{BIT}(0,z) = \mathsf{BIT}(0,z) = z_0$$
$$\mathsf{cmod2}(2,z) = \mathsf{cmod2}(1,z) - 2\mathsf{BIT}(1,z) \times \mathsf{cmod2}(1,z) + \mathsf{BIT}(1,z) = z_0 - 2z_0z_1 + z_1$$
$$= \begin{cases} 0 & \text{if } z_0 = z_1 \\ 1 & \text{otherwise.} \end{cases}$$

Therefore, to obtain full characterization of $\mathbf{FAC}^0[\mathbf{2}]$ we simply endow $\mathbb{ACDL}$ with the ℓ-2ODE schema, capturing Mod 2 gates.

▶ **Theorem 14.** $\mathbf{FAC}^0[\mathbf{2}] = [\mathcal{B}_0; \circ, \ell\text{-ODE}_1, \ell\text{-2ODE}]$.

Proof Sketch. ($\supseteq$) As in [3] for functions and schemas in $\mathbb{ACDL}$, plus Prop. 11 for ℓ-2ODE. ($\subseteq$) By capturing computation by circuits defining $\mathbf{FAC}^0[\mathbf{2}]$ via ℓ-2ODE (Remark 12). ◀

The relationship between strict ℓ-2ODE and non-strict ℓ-b$_0$ODE is clarified by the remark below, which, due to closure of ℓ-b$_0$ODE under $\mathbf{FAC}^0[\mathbf{2}]$ computation [4, Th. 19], offers an alternative proof of Prop. 11.

▶ Remark 15 (ℓ-2ODE vs. ℓ-b$_0$ODE). The schema ℓ-2ODE can be seen as a special case of ℓ-b$_0$ODE s.t. $K(x, \mathbf{y}, h_k(x, \mathbf{y}), f(x, \mathbf{y})) = \mathsf{cosg}(f(x, \mathbf{y})) \times h_k(x, \mathbf{y}) + \mathsf{sg}(f(x, \mathbf{y})) \times \mathsf{cosg}(h_k(x, \mathbf{y}))$.

More general connections can be established between the 1-BRN, ℓ-b$_0$ODE and ℓ-2ODE schemas (see Remark 35 in App. A.1.1 for the former two and giving a third, indirect, proof of Th. 14). However, using ℓ-2ODE offers the advantage of a very straightforward proof that does not rely on results in group theory or convoluted combinatorial arguments. Additionally, its close link with modulo counting resources is what enables us to scale the characterization across all classes; unlike k-BRN which jumps to $\mathbf{FNC}^1$ already for $k = 4$.

We conclude our investigation of $\mathbf{FAC}^0[\mathbf{2}]$ by remarking that similar completeness results holds in the non-uniform setting too. In this context, functions and relations that described the circuits for all n (the so-called direct connection language [30]) are given as basic functions and are added to $\mathcal{B}_0$.

▶ **Proposition 16.** *If $f : \mathbb{N} \to \mathbb{N}$ is computable by a family $\mathcal{C} = (C_n)_{n \geq 0}$ of polynomial size and constant depth circuits including Mod 2 gates, then it is in $[\mathcal{B}_0, \mathbf{circ}_{\mathcal{C}}; \circ, \ell\text{-ODE}_1, \ell\text{-2ODE}]$, where $\mathbf{circ}_{\mathcal{C}} = \{C, L_0^{in}, L_0^{\neg}, L_e\}$, denotes the set of characteristic functions associated to predicate and relations describing the underlying graph of a circuit and its level of gates, resp.*

3.2 A family of algebras for $\mathbf{FAC}^0[\mathbf{n}]$

Nicely, the framework of Sec. 3.1 uniformly scales to all natural numbers $n \in \mathbb{N}$, when considering the family of schemas ℓ-nODE below.

▶ **Definition 17** (ℓ-nODE schema). *Let $n \in \mathbb{N}^{>1}$, $g : \mathbb{N}^p \to \{0, \ldots, n-1\}$ and $k : \mathbb{N}^{p+1} \to \{0, 1\}$, then $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is defined by ℓ-nODE from g and k if it is the solution of the IVP with initial value $f(0, \mathbf{y}) = g(\mathbf{y})$ and s.t.:*

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = -n \times k(x, \mathbf{y}) \times \left\lfloor \frac{f(x, \mathbf{y})}{n-1} \right\rfloor + k(x, \mathbf{y}).$$

Note that the integer division symbol is used here with a slight abuse of notation since, for any x and $\mathbf{y}$, $f(x, \mathbf{y})$ takes a value smaller than n, so that division is actually only checking whether $f(x, \mathbf{y}) = n-1$. Hence, the kind of division involved here can be computed by circuits of constant depth. Indeed, if $k(x, \mathbf{y}) = 0$, then $f(x, \mathbf{y}) = f(\alpha(\ell(x)-1), \mathbf{y})$, while if $k(x, \mathbf{y}) = 1$, then $f(x, \mathbf{y}) = f\big(\alpha(\ell(x)-1), \mathbf{y}\big) - n \times \left\lfloor \frac{f(\alpha(\ell(x)-1), \mathbf{y})}{n-1} \right\rfloor + 1$, i.e. $f(x, \mathbf{y}) = f\big(\alpha(\ell(x)-1), \mathbf{y}\big) + 1$ if $f\big(\alpha(\ell(x)-1), \mathbf{y}\big) < n-1$ and $f(x, \mathbf{y}) = f\big(\alpha(\ell(x)-1), \mathbf{y}\big) - n = 0$ if $f(\alpha(\ell(x)-1), \mathbf{y}) = 2$. From this the closure property follows (see App. A.1.1).

▶ **Proposition 18.** *Let $n \in \mathbb{N}$. If f is defined by ℓ-nODE from functions in $\mathbf{FAC}^0[\mathbf{n}]$, then f is in $\mathbf{FAC}^0[\mathbf{n}]$ as well.*

This bound on f values also enforces that the typical $\mathbf{FTC}^0$ function BCount [30] (counting the number of 1's of the binary representation of its input) cannot be defined by ℓ-nODE.

▶ **Theorem 19.** *For any* $n \in \mathbb{N}^{>1}$, $\mathbf{FAC}^0[\mathbf{n}] = [\mathcal{B}_0; \circ, \ell\text{-ODE}_1, \ell\text{-}n\text{ODE}]$.

Proof Sketch. ($\supseteq$) Following [3], plus Prop. 18 for ℓ-nODE. ($\subseteq$) Generalizing results for constant-depth circuits computation (via $\mathbb{ACDL}$) with Mod n gates, which can be rewritten in our class as functions to count modulo n, namely, $\mathsf{cmodn}(x, x)$, where cmodn is defined as an instance of ℓ-nODE being the solution of the IVP s.t. $g(\mathbf{y}) = 0$ and $k(x, \mathbf{y}) = \mathsf{BIT}(\ell(x), \mathbf{y})$. ◀

▶ Remark 20. Clearly, ℓ-2ODE is nothing but a special case of ℓ-nODE for $n = 2$ so that Prop. 11 can be alternatively established following the intuition of Th. 19.

As expected, characterizations in the non-uniform setting are established without substantial modifications with respect to Prop. 16.

Before concluding Sec. 3.2, we remark that an equivalent characterization for these classes can be obtained not only by considering an alternative family of strict schemas s.t. $\frac{\partial f(x,\mathbf{y})}{\partial \ell} = -n \times \left\lfloor \frac{f(x,\mathbf{y})+k(x,\mathbf{y})}{n} \right\rfloor + k(x, \mathbf{y})$, but also their *non-strict* counterpart, obtained by allowing only special calls to f.

▶ Remark 21 (Non-strict counterpart). Let $n \in \mathbb{N}^{>2}$, $g : \mathbb{N}^p \to \{0, \ldots, n-1\}$ and $k : \mathbb{N}^{p+1} \to \mathbb{N}$. The function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ defined as the solution of the IVP s.t. $f(0, \mathbf{y}) = g(\mathbf{y})$ and $\frac{\partial f(x,\mathbf{y})}{\partial \ell} = -n \times \mathsf{sg}(f(x, \mathbf{y}) - c) \times k(x, \mathbf{y}) + k(x, \mathbf{y})$ for $c = n - 2$ is exactly the function computed by ℓ-nODE from the same g and k.

The systematic investigation of the relationship between strict and non-strict schemas, which has just been initiated here, deserves further attention but is left for future work.

3.3 Reaching $\mathbf{FTC}^0$, again

The family of schemas in Def. 17 ℓ-nODE have been so defined that $n \in \mathbb{N}^{>2}$. Indeed, a schema analogous to ℓ-nODE but with $n = 0$ is stronger than any instance of ℓ-nODE as, together with $\mathbb{ACDL}$, it is enough to capture $\mathbf{FTC}^0$.

▶ **Definition 22** (ℓ-0ODE schema). *Let* $g : \mathbb{N}^p \to \{0, 1\}$ *and* $k : \mathbb{N}^{p+1} \to \{0, 1\}$, *then* $f : \mathbb{N}^{p+1} \to \mathbb{N}$ *is defined by* ℓ-0ODE *from* g *and* k *if it is the solution of the IVP with initial value* $f(0, \mathbf{y}) = g(\mathbf{y})$ *and s.t.* $\frac{\partial f(x,\mathbf{y})}{\partial \ell} = k(x, \mathbf{y})$.

Computation performed by this schema can be easily obtained by iterated (bit-)sum of the values of k, which is in $\mathbf{FTC}^0$ [30], so that the desired closure property follows. Conversely, it is easy to see that ℓ-0ODE can compute BCount, considering $k(x, \mathbf{y}) = \mathsf{BIT}(\ell(x), \mathbf{y})$ and computing $f(x, x)$. The result below follows immediately (see App. A.2).

▶ **Theorem 23.** $\mathbf{FTC}^0 = [\mathcal{B}_0; \circ, \ell\text{-ODE}_1, \ell\text{-0ODE}]$

This theorem, together with the fact that ℓ-0ODE can simulate not only bit-counting but also counting modulo n for any $n \in \mathbb{N}$, gives us an extremely simple proof that $\mathbf{FAC}^0[\mathbf{n}]$ is included in $\mathbf{FTC}^0$ in a totally machine-independent framework.

▶ **Proposition 24.** *For any* $n \in \mathbb{N}^{>1}$, $[\mathcal{B}_0; \circ, \ell\text{-ODE}_1, \ell\text{-}n\text{ODE}] \subseteq [\mathcal{B}_0; \circ, \ell\text{-ODE}_1, \ell\text{-0ODE}]$.

Proof Sketch. Let $\mathsf{bcount}(x, y)$ be defined by ℓ-0ODE from g and k s.t. $g(\mathbf{y}) = 0$ and $k(x, \mathbf{y}) = \mathsf{BIT}(x, y)$. Clearly, $\mathsf{bc}(x) = \mathsf{bcount}(x, x)$ counts the number of 1's in the binary representation of x. We conclude by defining $\mathsf{mod_n}(x) = \mathsf{bc}(x) - \left\lfloor \frac{\mathsf{bc}(x)}{n} \right\rfloor \times n$, which clearly returns the sum modulo n of the 1's in the binary representation of the x, and whose construction only involves functions and schemas in the algebra of Th. 23. ◀

▶ **Corollary 25.** *For any* $n \in \mathbb{N}$, $\mathbf{FAC}^0[\mathbf{n}] \subseteq \mathbf{FACC^0} \subseteq \mathbf{FTC}^0$.

4 From ODE-based algebras to bounded arithmetic

Inspired by the ODE-design of schemas presented in the previous section, we introduce proof-theoretical characterizations for constant-depth classes, including modulo counting. In Sec. 4.1, we recall the definition of the first-order theory TAC^0, introduced in [15] to capture $\mathbf{FAC}^0$, and show that passing through $\mathbb{ACDL}$ simplifies the proof. In Sec. 4.2, we present new purely syntactical characterizations for all classes $\mathbf{FAC}^0[\mathbf{n}]$, by introducing a family of simple "counting-modulo n" axioms. Finally, in Sec. 4.3, we propose an alternative approach, this time considering a new rule.

4.1 Bounded theory for $\mathbf{FAC}^0$, revised

In [15], it is shown that the functions in $\mathbf{FAC}^0$ are precisely those which are esb-definable in the first-order theory TAC^0 (Def. 28). While the original proof passes through Clote's $\mathcal{A}_0$ [13], here we consider $\mathbb{ACDL}$ as our auxiliary algebra, yielding a more direct characterization proof. The main difference lies in the treatment of cases involving Bit Comprehension, which intuitively corresponds to the computation performed by ℓ-ODE$_1$, rather than by CRN.

A formula is said to be *essentially sharply bounded in a theory* T, if it is in the smallest family including a set of atomic formulas, closed under Boolean connectives and sharply bounded quantification and s.t., if $A(\mathbf{a}, x)$ and $B(\mathbf{a}, x)$ are in the class and $T \vdash \exists! x \leq s(\mathbf{a}).A(\mathbf{a}, x)$, then also $\exists x \leq s(\mathbf{a})(A(\mathbf{a}, x) \wedge B(\mathbf{a}, x))$ and $\forall x \leq s(\mathbf{a})(A(\mathbf{a}, x) \to B(\mathbf{a}, x))$ are.

▶ **Definition 26** (esb-definability). *A function is said to be* esb-definable *in a theory T if there is an esb-formula A and a term t s.t. (i) $T \vdash \forall \mathbf{x} \exists y \leq t(\mathbf{x}).A(\mathbf{x}, y)$ (ii) $T \vdash \forall \mathbf{x} \forall y \forall z (A(\mathbf{x}, y) \wedge A(\mathbf{x}, z) \to y = z)$ and (iii) $\forall \mathbf{x} A(\mathbf{x}, f(\mathbf{x}))$ is satisfied.*

This is a restricted form of Σ^b_k-definability [11], where Σ^b_k-formulas are replaced by esb ones. The language of TAC^0 is made of 0, 1, $|\cdot|$, $\div 2$, $+$, $-$, $\#$, $\leq, =$, PAD, MSP, where (informally) $|x|$ is the length symbol, $\text{PAD}(x, y)$ to $2^{\ell(y)} \times x$ and $\text{MSP}(x, y)$ to $\lfloor \frac{x}{2^y} \rfloor$.

▶ **Notation 27.** *We will use the following standard abbreviations for logical and arithmetic symbols: for any formula A and B, $A \leftrightarrow B$ is a shorthand for $(A \to B) \wedge (B \to A)$, $a \neq b$ for $\neg(a = b)$, $2 \times a = a + a$, $\mathit{MOD2}(a) = a - 2 \times (a \div 2)$ and $\mathit{BIT}(i, a) = \mathit{MOD2}(\mathit{MSP}(a, i))$.*

▶ **Definition 28** (Theory TAC^0). *The theory TAC^0 is defined by axioms for all basic functions and predicates in its language, plus the following peculiar ones:*

- *Bit Extensionality axiom* (BE): $|a| = |b|, \forall i < |a| \big(\mathit{BIT}(i, a) = \mathit{BIT}(i, b)\big) \to a = b$
- *Bit Comprehension axiom scheme* (BC): $\exists y < 2^{|s(\mathbf{a})|} \forall i < |s(\mathbf{a})| \big(\mathit{BIT}(i, y) = 1 \leftrightarrow A(\mathbf{a}, i)\big)$, *where $A(\mathbf{a}, i)$ is esb*

and the inference rule esb-LIND*:*

$$\frac{A(a), \Gamma \Rightarrow \Delta, A(a+1)}{A(0), \Gamma \Rightarrow \Delta, A(|t|)} \text{ esb-LIND}$$

where $A(a)$ is esb and a satisfies the eigenvariable condition.

The characterization proof for $\mathbf{FAC}^0$ in terms of esb-definability in TAC^0 is structurally similar to that in [15] but, as anticipated, it relies on $\mathbb{ACDL}$ rather than Clote's $\mathcal{A}_0$ [13].

▶ **Theorem 29.** *A function f is in $\mathbf{FAC}^0$ iff it is esb-definable in TAC^0*

Proof Sketch. ($\subseteq$) By induction on the structure of functions in $\mathbb{ACDL}$ (via [3]). All base and inductive cases except for ℓ-ODE$_1$ are standard. Let f be defined by ℓ-ODE$_1$ from g and k in $\mathbb{ACDL}$ with codomain $\{0, 1\}$. W.l.o.g. let $g(\mathbf{y}) = 0$. By Df. 3, $f(x, \mathbf{y}) = \sum_{i=0}^{\ell(x)-1} 2^{\ell(x)-i} \times$

$k(x, \mathbf{y})$, that is $\mathsf{BIT}(j, f(x, \mathbf{y})) = k(\alpha(\ell(x)-(j+1)), \mathbf{y})$, for any $j \in \{0, \dots, \ell(x)-1\}$. By IH, k is esb-definable and, by (BC), $f(m, \mathbf{n})$ is uniquely defined by y, $\exists y \leq 2^{|m|} \forall i \leq |m|(\text{BIT}(i, y) = 1 \leftrightarrow k(2^{|m|-(i+1)} - 1, \mathbf{n}) = 1)$. ($\supseteq$) By simultaneous induction on the free cut free derivation height. The most interesting case is that of derivation of height 0 defined by an initial sequent (BC). W.l.o.g. let A be sharply bounded (by IH). It can be shown that, for any sharply bounded formula, there is a corresponding function in $\mathbb{ACDL}$ which is 1 when the formula holds and 0 otherwise. Let k_A be the function corresponding to $A(\mathbf{a}, i)$. Then, the desired function is constructed by defining f_a as the solution of the IVP s.t. $f_a(0, \mathbf{y}) = 0$ and $\frac{\partial f_a(x)}{\partial \ell} = f_a(x) + k_A(\ell(x), \mathbf{a})$. We conclude by considering $f_A(\mathbf{a}) = f_a(2^{\ell(s(\mathbf{a}))}, \mathbf{a})$, in $\mathbb{ACDL}$, s.t. $f_A(\mathbf{a}) \leq 2^{|s(\mathbf{a})|}$ and $\forall i \leq |s(\mathbf{a})|\big(\text{BIT}(i, f_A(\mathbf{a})) = 1 \leftrightarrow k_A(\ell(x) - (i+1), \mathbf{a})\big)$ are satisfied. ◀

4.2 Bounded theories for $\mathbf{FAC}^0[\mathrm{n}]$ by extending the language

In this Section, we present a family of homogeneous first-order bounded theories, $\mathsf{TAC}\mathrm{n}^0$, capturing all modulo-counting classes $\mathbf{FAC}^0[\mathbf{n}]$ and the characteristic axioms of which are crucially inspired by the ODE schemas introduced in Sec. 3. The language of $\mathsf{TAC}\mathrm{n}^0$ is obtained by endowing that of TAC^0 by the unary symbol CMOD-N.

▶ **Definition 30** (Theory $\mathsf{TAC}\mathrm{n}^0$). *The theory* $\mathsf{TAC}\mathrm{n}^0$ *is defined by all axioms and schemas of* TAC^0 *(Df. 28), plus the following axioms:*

N1 $\textsc{cmod-n}(x) = b$, *for* $x = b$ *and* $b \in \{0, 1\}$

N2 $\textsc{cmod-n}(x) = \textsc{cmod-n}\big(\lfloor \frac{x}{2} \rfloor\big) - n \times \big\lfloor \frac{\textsc{cmod-n}(\lfloor \frac{x}{2} \rfloor)}{n-1} \big\rfloor \times \big(x - 2 \times \lfloor \frac{x}{2} \rfloor\big) + \big(x - 2 \times \lfloor \frac{x}{2} \rfloor\big)$.

Intuitively, the symbol CMOD-N is the syntactical counterpart of ℓ-nODE (or of the function cmodn; see App. A.1.1), which counts modulo n *the number of 1's in the binary representation of the input*. Notice that in the definition above the symbol for integer division by $n-1$, $\lfloor \frac{z}{n-1} \rfloor$ is used with a slight abuse of notation, but it is actually a shorthand for $z \leq n-1$. Similarly, the multiplication symbol is not actually in the language of $\mathsf{TAC}\mathrm{n}^0$, but we are here using it to denote bit-multiplication, i.e., multiplication of a number by 0 or 1.

▶ **Theorem 31.** *For* $n \in \mathbb{N}^{>1}$, *a function* f *is in* $\mathbf{FAC}^0[\mathbf{n}]$ *iff it is esb-definable in* $\mathsf{TAC}\mathrm{n}^0$.

Proof Sketch. ($\subseteq$) By induction on the structure of functions in the ODE-based algebra presented in Th. 19. Basic functions and inductive steps, except for ℓ-nODE are treated as in Th. 29. Let f be defined by ℓ-nODE from g and k in $\mathbf{FAC}^0[\mathbf{n}]$. By Df. 3, this function counts modulo n the bit-sum $t(x, \mathbf{y}) = \sum_{u=-1}^{\ell(x)-1} k(\alpha(u), \mathbf{y}) \times 2^{(\ell(x)-u+1)}$, with the convention that $k(\alpha(-1), \mathbf{y}) = g(\mathbf{y})$ (definable by ℓ-ODE$_1$ from g and k, in the given algebra by IH). We conclude by considering the constant bound $n-1$ and the formula $\exists y \leq n-1.\big(\textsc{cmod-n}(t(m, \mathbf{n})) = y\big)$ esb-defining $f(m, \mathbf{n})$. ($\supseteq$) By induction on the corresponding free cut free proof. Since, by Df. 30, only axioms N1 and N2 have been added to TAC^0, we can consider (basic) cases involving CMOD-N only. Other cases are as in Th. 29. In particular, for the only non trivial case N2, existence and uniqueness easily follow from application of characteristic axioms and (the esb-PIND counterpart of) esb-LIND, while for (iii) we consider $\mathsf{cmn}(x) = \mathsf{cmodn}(x, x)$, where cmodn is defined as in Th. 19. As desired this function is s.t., for any $x \in \mathbb{N}$, if $x = 0$, $\mathsf{cmn}(x) = 0$, if $x = 1$, $\mathsf{cmn}(x) = 1$, and if $x = 2z + b$ (with $b \in \{0, 1\}$), $\mathsf{cmn}(x) = \mathsf{cmn}(z) - n \times (\mathsf{cmn}(z) \geq n-1) \times \mathsf{BIT}(x, 0) + \mathsf{BIT}(x, 0)$. We conclude by noticing that, since cmn is in the algebra presented in Th. 19, by the same theorem, is in $\mathbf{FAC}^0[\mathbf{n}]$. ◀

4.3 Bounded theories for $\mathbf{FAC^0[n]}$ by extending the rules

In this Section, we propose alternative bounded arithmetic obtained by extending the rule system of the theory without enlarging the language. The resulting system, $\mathsf{TAC^0}[n]$, not only captures $\mathbf{FAC^0[n]}$ using the same language as $\mathsf{TAC^0}$, so to possibly allow a model-theoretic study and comparison between corresponding complexity classes. For instance, the separation of $\mathbf{FAC^0}$ and $\mathbf{FAC^0[n]}$ translates to the existence of a model of $\mathsf{TAC^0}$ that is not a model of $\mathsf{TAC^0}[n]$. In addition, its propositional translation may introduce a new way to formulate propositional proof systems with counting gates.

Again, the new system is strongly inspired by ℓ-nODE. However, this approach more faithfully reflects the meaning of ODE in the context of arithmetic. Since ODE schemes state that the totality of difference implies the totality of functions, they are essentially induction schemes in disguise. In particular, the totality of $\mathsf{cmod}n$ (the function that outputs the number of bits modulo n) implies the totality of functions defined by it. Informally, it holds

$$\frac{f(x,\mathbf{y}) \text{ is defined from } k \text{ by } \ell\text{-}n\text{ODE} \qquad k(x,\mathbf{y}) \text{ is total} \qquad f(0,\mathbf{y}) \text{ is defined}}{f(x,\mathbf{y}) \text{ is total}}$$

which intuitively translates into the following inference rule.

▶ **Definition 32.** *Let $n \in \mathbb{N}$ and B_k be some esb-formula. Then, n-PIND(B) is the following inference rule:*

$$\frac{\Gamma,\, b < n\ B(\lfloor \tfrac{a}{2} \rfloor, b)\ \Rightarrow\ \Delta,\, \exists z < n\ B(a,z) \wedge A(a,b,z) \qquad \Gamma\ \Rightarrow\ \Delta,\, \exists y < 2\ B_k(a,y)}{\Gamma,\, b < n\ B(0,b)\ \Rightarrow\ \Delta,\, \exists y\, B(t,y)}$$

where A is the formula s.t. $A(x,y,z) \iff \exists z' < 2\ z = y - n \times z' \times \left\lfloor \frac{y}{n-1} \right\rfloor + z' \wedge B_k(x,z')$. Note that the terms a, b must satisfy the eigenvariable condition in the premise.

▶ **Definition 33.** $\mathsf{TAC^0}[n] = \mathsf{TAC^0} + \{n\text{-}PIND(B) : B \text{ is esb-formula}\}$.

It remains to show that $\mathsf{TAC^0}[n]$ captures exactly $\mathbf{FAC^0[n]}$. The soundness proof is relatively standard. The completeness proof is more involved: it requires encoding the history of computation à la Buss.

▶ **Theorem 34.** *The class of esb-definable formulas in $\mathsf{TAC^0}[n]$ is exactly $\mathbf{FAC^0[n]}$.*

5 Conclusion and perspective

The paper introduces *new* and *uniform* machine-independent characterizations for constant-depth small circuit classes including modulo counting for any constant, in both the recursion- and proof-theoretic settings. The key novelty of our approach lies in its reliance on very simple (i.e., strict) recursion schemas defined by discrete ODEs. This apparently simple shift in perspective, initiated in [9] for **FP** and scalable from the discrete to the continuous realm, has already proven to be highly effective in capturing (small circuit) classes, possibly including (unrestricted) counting gates [3, 4]. In the ODE framework it becomes natural to "parametrize" the recursion schema in ways that are not natural (hence were not considered) in the classical recursion-theoretic context. This is precisely what allows us to obtain the first implicit characterizations for constant-depth circuit computation that extend to modulo counting. By simply equipping algebras with properly-defined (still strict) schemas, the correspondence between strict schemas and $\mathbf{FAC^0}$ (see also [3, 4]) extends to $\mathbf{FAC^0[n]}$ classes. This was not the case for characterizations presented so far in the literature; as mentioned,

$\mathbf{FAC}^0[\mathbf{2}]$ and $\mathbf{FAC}^0[\mathbf{6}]$ were captured by restricting k-BRN, the schema defining $\mathbf{FNC}^1$ (for $k \geq 4$), in a way that did not allow for a generalization to other modulo counting classes. Furthermore, by mirroring the behavior of the new schemas, we present bounded arithmetic theories capturing the corresponding circuit classes. As an additional positive feature, our axioms are conceptually simpler than the corresponding rules by [15] and the naturalness of the approach leads to completely self-contained and technically straightforward proofs, which were not the case for (partial) characterizations obtained in the less flexible frameworks of [15, 4].

Overall, this work is conceived as a first step towards a more general study of (circuit) complexity from the viewpoint of ODEs and several research directions are still open. First, it would be profitable to better understand connections between strict and non-strict schemas (the latter, allowing one to call more freely for preceding values in their definition), to delineate the frontier between computation in constant and (bounded) logarithmic-depth in this machine-independent framework. In this vein, the role of the change of variable as a method of reduction should be investigated much further. Other natural and intriguing questions concern ℓ-ODE schemas allowing for exact division or over the reals. In the longer term, it would be interesting to investigate separation between classes or bounds, possibly in terms of syntactical constraints over ODE schemas (or bounded theories) and relying on tools from the difference calculus. Regarding bounded arithmetic, this work is the first attempt to link ODE-based characterizations and proof-theoretical approaches. Although here we can rely on multiple simplifications due to the limited small circuit settings (i.e., the fact that we are mostly dealing with "small sequences", whose encoding is especially simple), our results seem to point to promising directions to "translate" ODE-based algebras into corresponding rule systems (and vice versa). At first, it would be natural to see whether this approach can be used to define ODE-designed deduction rules for $\mathbb{LDL}$, the algebra defined in [9] for **FP**. Follow-up, more general studies might help to develop an original machine-independent approach to relate recursion- and proof-theoretical views of complexity.

References

1 B. Allen. Arithmetizing uniform NC. *Ann. Pure Appl. Log.*, 53(1):1–50, 1991. doi:10.1016/0168-0072(91)90057-S.

2 E. Allender and K.W. Wagner. Counting hierarchies: polynomial time and constant. *Bull. EATCS*, 40:182–194, 1990.

3 M. Antonelli, A. Durand, and J. Kontinen. A New Characterization of FAC^0 via Discrete Ordinary Differential Equations. In *49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)*, volume 306 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.MFCS.2024.10.

4 M. Antonelli, A. Durand, and J. Kontinen. Characterizing Small Circuit Classes from FAC^0 to FAC^1 via Discrete Ordinary Differential Equations. In *50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025)*, volume 345 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.MFCS.2025.10.

5 T. Barlag, V. Holzapfel, L. Strieker, J. Virtema, and H. Vollmer. Graph neural networks and arithmetic circuits. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

6 M. Blanc and O. Bournez. A characterization of polynomial time computable functions from the integers to the reals using discrete ordinary differential equations. In Jérôme Durand-Lose and György Vaszil, editors, *Machines, Computations, and Universality - 9th International Conference, MCU 2022, Debrecen, Hungary, August 31 - September 2, 2022, Proceedings*, volume 13419 of *Lecture Notes in Computer Science*, pages 58–74. Springer, 2022. doi:10.1007/978-3-031-13502-6_4.

7 M. Blanc and O. Bournez. A characterisation of functions computable in polynomial time and space over the reals with discrete ordinary differential equations: Simulation of turing machines with analytic discrete odes. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, Bordeaux, France, August 28 - September 1, 2023*, volume 272 of *LIPIcs*, pages 21:1–21:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.MFCS.2023.21.

8 M. Blanc and O. Bournez. The complexity of computing in continuous time: Space complexity is precision. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, Tallinn, Estonia, July 8-12, 2024*, volume 297 of *LIPIcs*, pages 129:1–129:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ICALP.2024.129.

9 O. Bournez and A. Durand. Recursion schemes, discrete differential equations and characterization of polynomial time computations. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, Aachen, Germany, August 26-30, 2019*, volume 138 of *LIPIcs*, pages 23:1–23:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.MFCS.2019.23.

10 O. Bournez and A. Durand. A characterization of functions over the integers computable in polynomial time using discrete ordinary differential equations. *Comput. Complex.*, 32(2):7, 2023. doi:10.1007/S00037-023-00240-1.

11 S. Buss. *Bounded Arithmetic*. PhD thesis, Department of Mathematics, 1986.

12 P.G. Clote. A sequential characterization of the parallel complexity class NC. Technical report, Boston College, 1988.

13 P.G. Clote. *Sequential, machine-independent characterizations of the parallel complexity classes AlogTIME, AC^k, NC^k and NC*, pages 49–69. Progress in Computer Science and Applied Logic. Birkhäuser, Boston, MA, 1990.

14 P.G. Clote. On polynomial size Frege proofs of certain combinatorial principles. In P. Clote and Krjícek J., editors, *Arithmetic, Proof Theory, and Computational Complexity*, pages 166–184. Clarendon Press, 1993.

15 P.G. Clote and G. Takeuti. First order bounded arithmetic and small complexity classes. In P.G. Clote and J.B. Remmel, editors, *Feasible Mathematics II*, pages 154–218. Springer, 1995.

16 A. Cobham. The intrinsic computational difficulty of functions. In *Logic, Methodology and Phylosophy of Science: Proc. 1964 International Congress*, pages 24–30. North-Holland Publishing Company, 1965.

17 K.J. Compton and C. Laflamme. An algebra and a logic for NC^1. In *Proceedings of the Third Annual Symposium on Logic in Computer Science (LICS '88), Edinburgh, Scotland, UK, July 5-8, 1988*, pages 12–21. IEEE Computer Society, 1988. doi:10.1109/LICS.1988.5096.

18 S. Cook and P. Nguyen. Theories for TC^0 and other small circuit classes. *Log. Meth. Comput. Sci.*, 2(1–40), 2006.

19 A. Durand, A. Haak, and H. Vollmer. Model-theoretic characterization of boolean and arithmetic circuit classes of small depth. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 354–363. ACM, 2018. doi:10.1145/3209108.3209179.

20 M. L. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. In *22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981*, pages 260–270. IEEE Computer Society, 1981. doi:10.1109/SFCS.1981.35.

21 A.O. Gelfand. *Calcul des différences finies*. Dunod, 1963.

22 L. Hella, J. Kontinen, and K. Luosto. Regular representations of uniform TC^0. *ACM Trans. Comput. Log.*, 26(4):21:1–21:46, 2025. doi:10.1145/3750044.

23 William Hesse. Division is in uniform TC^0. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *Automata, Languages and Programming*, pages 104–114, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

24 N. Immerman. Languages that capture complexity classes. *SIAM J. Comput.*, 16:760–778, 1987. doi:10.1137/0216051.

25 N. Immerman. Expressibility and parallel complexity. *SIAM J. Comput.*, 18(3):625–638, 1989. doi:10.1137/0218043.

26 C. Jordan and K. Jordán. *Calculus of finite differences*, volume 33. American Mathematical Soc., 1965.

27 W. Maass, G. Schnitger, and E.D. Sontag. On the computational power of sigmoid versus boolean threshold circuits. In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, October 1-4, 1991*, pages 767–776. IEEE Computer Society, 1991. doi:10.1109/SFCS.1991.185447.

28 I. Parberry. *Circuit Complexity and Neural Networks*. MIT Press, 1994.

29 R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 77–82. ACM, 1987. doi:10.1145/28395.28404.

30 H. Vollmer. *Introduction to Circuit Complexity: A Uniform Approach*. Springer, 1999.

A Proofs for Section 3 (Characterizing small circuit classes with modulo counting gates)

A.1 Proofs from Section 3.1

Proof of Prop. 11. By Def. 10,

$$f(x,\mathbf{y}) = \sum_{u=-1}^{\ell(x)-1} \left(\prod_{t=u+1}^{\ell(x)-1} \Big(1 - 2k(\alpha(t),\mathbf{y})\Big) \right) \times k(\alpha(u),\mathbf{y})$$

for $k \leq 1$ and with the convention that $k(\alpha(-1),\mathbf{y}) = f(0,\mathbf{y})$. This is computable in $\mathbf{FAC}^0[\mathbf{2}]$. Indeed, it holds that:

$$f(x,\mathbf{y}) = \sum_{u=-1}^{\ell(x)-1} k(\alpha(u),\mathbf{y}) \times (-1)^{\sharp\{t\in[u+1,\ell(x)-1]:k(\alpha(t),\mathbf{y})=1\}}.$$

All terms above in which $k(\alpha(u),\mathbf{y}) = 0$ vanish from the sum (recall that $k(\alpha(-1),\mathbf{y}) = g(\mathbf{y})$). Let $\{u_1,\ldots,u_a\} = \{t \in [-1,\ell(x)-1] : k(\alpha(t),\mathbf{y}) = 1\}$. Then,

$$f(x,\mathbf{y}) = \sum_{i=0}^{a}(-1)^{a-i} = \sum_{i=0}^{a}(-1)^{i} = \begin{cases} 0 & \text{if } a \text{ is odd} \\ 1 & \text{if } a \text{ is even.} \end{cases}$$

In terms of (constant-depth) circuit, to compute $f(x,\mathbf{y})$ we first compute all values $g(\mathbf{y})$ and $k(\alpha(u),\mathbf{y})$ for $u \in \{0,\ldots,\ell(x)-1\}$. Then, we count the parity of those who are equal to 1 by a Mod 2 gate. Since g and k are in $\mathbf{FAC}^0[\mathbf{2}]$ by hypothesis, the whole computation is also in $\mathbf{FAC}^0[\mathbf{2}]$. ◀

Proof of Theorem 14. ($\subseteq$) For all functions and schemas in $\mathbb{ACDL}$ the proof is analogous to the one presented in [3], while the closure of $\mathbf{FAC}^0[\mathbf{2}]$ under ℓ-2ODE has been established in Prop. 11. ($\supseteq$) Recall that computation through constant-depth, polynomial-size circuits is already captured by $\mathbb{ACDL}$. Moreover, by Remark 12, ℓ-2ODE allows to capture computation performed by Mod 2 gates, the only gates to be added to $\mathbf{FAC}^0$ circuits to obtain $\mathbf{FAC}^0[\mathbf{2}]$. This is enough to conclude our proof (see, e.g., [15, Th. 2.2]). ◀

Proof of Remark 15. The schema ℓ-2ODE can be seen as a special case of ℓ-b$_0$ODE such that $K(x, \mathbf{y}, h_k(x, \mathbf{y}), f(x, \mathbf{y})) = \mathsf{cosg}(f(x, \mathbf{y})) \times h_k(x, \mathbf{y}) + \mathsf{sg}(f(x, \mathbf{y})) \times \mathsf{cosg}(h_k(x, \mathbf{y}))$, i.e., a function f defined by ℓ-2ODE from g_k and h_k can be rewritten as an instance of ℓ-b$_0$ODE with initial value $f(0, \mathbf{y}) = g_k(\mathbf{y})$ and such that:

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = -f(x, \mathbf{y}) + \mathsf{cosg}(f(x, \mathbf{y})) \times h_k(x, \mathbf{y}) + \mathsf{sg}(f(x, \mathbf{y})) \times \mathsf{cosg}(h_k(x, \mathbf{y})).$$

Clearly, since it is easily provable that $f(x, \mathbf{y})$ returns only values in $\{0, 1\}$, this can be rephrased as follows:

$$\begin{aligned} f(x, \mathbf{y}) &= f(\ell(x) - 1) - f(\ell(x) - 1) + \begin{cases} \mathsf{cosg}(h_k(\alpha(x) - 1)) & \text{if } \mathsf{sg}(f(\alpha(\ell(x) - 1)) = 1 \\ h_k(\ell(x) - 1) & \text{if } \mathsf{cosg}(f(\ell(x) - 1)) = 1 \end{cases} \\ &= \begin{cases} \mathsf{cosg}(h_k(x, \mathbf{y})) & \text{if } f(\ell(x) - 1) = 1 \\ h_k(x, \mathbf{y}) & \text{if } f(\ell(x) - 1) = 0 \end{cases} \\ &= \begin{cases} 0 & \text{if } h_k(\ell(x) - 1) = f(\ell(x) - 1) \\ 1 & \text{if } h_k(\ell(x) - 1) \neq f(\ell(x) - 1) \end{cases} \\ &= f(\alpha(\ell(x) - 1), \mathbf{y}) - 2f(\alpha(\ell(x) - 1), \mathbf{y}) \times h_k(\ell(x) - 1) + h_k(\ell(x) - 1). \end{aligned}$$

where $f(\ell(x)-1)$ is a shorthand for $f(\alpha(\ell(x)-1), \mathbf{y})$ and similarly $h_k(\ell(x)-1)$ for $h_k(\alpha(\ell(x)-1), \mathbf{y})$. Due to completeness via ℓ-b$_0$ODE [4, Th. 19], this simple rewriting would offer an alternative proof of Prop. 11. ◀

The connection between 1-BRN and ℓ-2ODE is clarified by the following remark, showing that the former can be rewritten as an instance of the latter.

▶ Remark 35 (1-BRN vs. ℓ-2ODE). Recall that a function $f(x, \mathbf{y})$ is defined by 1-BRN from g and h if it such that $f(0, \mathbf{y}) = g(\mathbf{y})$ and $f(x, \mathbf{y}) = h\big(x, \mathbf{y}, f\big(\lfloor \frac{x}{2} \rfloor, \mathbf{y}\big)\big)$ where, for any $x, \mathbf{y}$, $f(x, \mathbf{y}) \leq 1$ and $i \in \{0, 1\}$. Since $f(x, \mathbf{y})$ takes only values 0 or 1, $f(x, \mathbf{y}) = f\big(\lfloor \frac{x}{2} \rfloor, \mathbf{y}\big) \times \big(h(x, \mathbf{y}, 1) - h(x, \mathbf{y}, 0)\big) + h(x, \mathbf{y}, 0)$. This can be rewritten as the solution of an IVP defined by ℓ-2ODE from the function g above and

$$k(x, \mathbf{y}) = \begin{cases} 0 & \text{if } h_0(x) \neq h_1(x) \text{ and } h_0(x) = 0 \\ & \text{or } h_0(x) = h_1(x), \mathsf{changes}(x) = i \text{ and } h_0(x) = i \\ 1 & \text{otherwise} \end{cases}$$

where $h_i(x)$ is a shorthand for $h(x, \mathbf{y}, i)$ and $\mathsf{changes}(x) = i$ abbreviates $\mu.z \in \{x - 1, \ldots, 0\}(h_0(z) = h_1(z)) = p \wedge \mathsf{cmod2}\big(\sum_{j=p}^{x-1} h_0(j) \neq h_0(j + 1)\big) = i$, intuitively returning the number of times such that $h_0(t)$ (and $h_1(t)$) changes with respect to the preceding value $h_0(t - 1)$ between x (for which $h_0(x) = h_1(x)$) and the closest value x' such that $h_0(x') \neq h_1(x')$.

In [4], it is shown that endowing $\mathcal{A}_0$ (i.e., $\mathbb{ACDL}$) with ℓ-b$_0$ODE is enough to capture $\mathbf{FAC}^0[\mathbf{2}]$. Since all the functions needed to rewrite 1-BRN using ℓ-2ODE are in the algebra introduced in Th. 14, Remark 35 provides a third, alternative (indirect [15]) proof of completeness for the desired characterization.

A.1.1 From $\mathbf{FAC^0[3]}$ to $\mathbf{FAC^0[n]}$

The schema characterizing $\mathbf{FAC^0[3]}$ is a special case of ℓ-ODE whose defining equation involves integer division, i.e. $\left\lfloor \frac{f(x,\mathbf{y})}{2} \right\rfloor$, $A = -3k(x,\mathbf{y}), B = k(x,\mathbf{y})$ and $k(x,\mathbf{y}) \leq 1$.[1]

▶ **Definition 36** (ℓ-3ODE schema). *Let $g : \mathbb{N}^p \to \{0,1,2\}$ and $k : \mathbb{N}^{p+1} \to \{0,1\}$, then $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is defined by ℓ-3ODE from g and k if it is the solution of the IVP with initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ and such that:*

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = -3 \times k(x,\mathbf{y}) \times \left\lfloor \frac{f(x,\mathbf{y})}{2} \right\rfloor + k(x,\mathbf{y}).$$

Clearly, either $k(\alpha(\ell(x)-1),\mathbf{y}) = 0$ and $f(x+1,\mathbf{y}) = f(\alpha(\ell(x)-1),\mathbf{y})$ or $k(\alpha(\ell(x)-1),\mathbf{y}) = 1$ and then,

$$\begin{aligned} f(x,\mathbf{y}) &= f(\alpha(\ell(x)-1),\mathbf{y}) - 3 \times \left\lfloor \frac{f(\alpha(\ell(x)-1),\mathbf{y})}{2} \right\rfloor + 1 \\ &= \begin{cases} f(\alpha(\ell(x)-1),\mathbf{y}) + 1 & \text{if } f(\alpha(\ell(x)-1),\mathbf{y}) < 2 \\ f(\alpha(\ell(x)-1),\mathbf{y}) - 2 = 0 & \text{if } f(\alpha(\ell(x)-1),\mathbf{y}) = 2. \end{cases} \end{aligned}$$

▶ **Proposition 37.** *If f is defined by ℓ-3ODE from functions in $\mathbf{FAC^0[3]}$, then f is in $\mathbf{FAC^0[3]}$ as well.*

Proof. Straightforward consequence of the observation above. Indeed, $f(x,\mathbf{y})$ can be computed by a circuit including Mod 3 gates in constant depth as follows:

- In the beginning, compute $g(\mathbf{y})$ and $k(\alpha(t),\mathbf{y})$'s for any $t \in \{0,\dots,x-1\}$, independently and in parallel. This can be done in $\mathbf{FAC^0[3]}$ by hypothesis.
- In one step, compute the sum of these values modulo 3 using one Mod 3 gate. ◀

That this schema is enough to capture $\mathbf{FAC^0[3]}$ is established following the techniques of Remark 12.

▶ Remark 38 (Counting modulo 3). Computation performed by Mod 3 gates can be simulated by ℓ-3ODE from k, for $k(x,\mathbf{y}) = \mathsf{BIT}(x,\mathbf{y})$; that is, by computing $\mathsf{cmod3}(x,x)$, where $\mathsf{cmod3}(x,\mathbf{y})$ is defined as the solution of the IVP with initial value $\mathsf{cmod3}(0,\mathbf{y}) = 0$ and such that $\frac{\partial \mathsf{cmod3}(x,\mathbf{y})}{\partial \ell} = -3 \times \left\lfloor \frac{\mathsf{cmod3}(x,\mathbf{y})}{2} \right\rfloor \times \mathsf{BIT}(\ell(x),\mathbf{y}) + \mathsf{BIT}(\ell(x),\mathbf{y})$. Clearly, for any $t \in \{0,\dots,x\}$ and $\mathbf{y}$, $\mathsf{cmod3}(\alpha(t),\mathbf{y}) \leq 2$. In particular, whenever $\mathsf{comd3}(t,\mathbf{y}) < 2$, the next value $t' \in \{0,\dots,x\}$ such that $\ell(t') = \ell(t)+1$ will be simply $\mathsf{cmod3}(t',\mathbf{y}) = \mathsf{cmod3}(t,\mathbf{y}) + \mathsf{BIT}(t,\mathbf{y})$, while when $\mathsf{cmod3}(t,\mathbf{y}) = 2$,

$$\mathsf{cmod3}(t',\mathbf{y}) = \begin{cases} \mathsf{cmod3}(\alpha(t),\mathbf{y}) = 2 & \text{if } \mathsf{BIT}(\alpha(t),\mathbf{y}) = 0 \\ \mathsf{cmod3}(\alpha(t),\mathbf{y}) - 2 = 0 & \text{if } \mathsf{BIT}(\alpha(t),\mathbf{y}) = 1 \end{cases}$$

Then, $\mathbf{FAC^0[3]}$ can be captured by simply endowing $\mathbb{ACDL}$ with ℓ-3ODE. Completeness in the uniform and non-uniform setting is established exactly as before.

▶ **Lemma 39.** $\mathbf{FAC^0[3]} = [\mathcal{B}_0; \circ, \ell\text{-ODE}_1, \ell\text{-3ODE}]$.

Proof of Lemma 39. ($\supseteq$) As in [3], for all functions and schemas except ℓ-3ODE, the closure property of which is proved in Proposition 37. ($\subseteq$) By [3] plus Remark 38, capturing computation through Mod 3 gates. ◀

[1] This schema can be seen as a generalization of ℓ-ODE$_3$, see [3].

▶ **Lemma 40.** *If $f : \mathbb{N} \to \mathbb{N}$ is computable by a family of polynomial size and constant depth circuits including* Mod *3 gates, $\mathcal{C} = (C_n)_{n\geq 0}$, then f is in the class $[\mathcal{B}_0, \mathbf{circ}_{\mathcal{C}}; \circ, \ell\text{-ODE}_1, \ell\text{-3ODE}]$.*

Proof of Lemma 40. Precisely as for Prop. 16 but constructing Eval so that $\mathsf{Eval}_{3e}(y, x)$ is equal to $\mathsf{cmod3}\Big(\sum_{i=0}^{\ell(x)^k}\{\mathsf{Eval}_{3e-3}(i, x) : C(x, i, t) = 1\}\Big)$. ◀

Proof of Prop. 18. By generalizing Prop. 11 and 37. In particular, if $k(\alpha(\ell(x)-1), \mathbf{y}) = 0$, $f(x, \mathbf{y}) = f(\alpha(\ell(x)-1), \mathbf{y})$. If $k(\alpha(\ell(x)-1), \mathbf{y}) = 1$,

$$\begin{aligned} f(x, \mathbf{y}) &= f(\alpha(\ell(x)-1), \mathbf{y}) - n \times \left\lfloor \frac{f(\alpha(\ell(x)-1), \mathbf{y})}{n} \right\rfloor + 1 \\ &= \begin{cases} f(\alpha(\ell(x)-1), \mathbf{y}) + 1 & \text{if } f(\alpha(\ell(x)-1), \mathbf{y}) < n-1 \\ f(\alpha(\ell(x)-1), \mathbf{y}) - n + 1 & \text{if } f(\alpha(\ell(x)-1), \mathbf{y}) = n-1 \end{cases} \\ &= \begin{cases} f(\alpha(\ell(x)-1), \mathbf{y}) + 1 & \text{if } f(\alpha(\ell(x)-1), \mathbf{y}) < n-1 \\ 0 & \text{if } f(\alpha(\ell(x)-1), \mathbf{y}) = n-1 \end{cases} \end{aligned}$$

Clearly, for any x and $\mathbf{y}$, $f(x, \mathbf{y}) \leq n-1$. Additionally, when $k(\alpha(\ell(x)-1), \mathbf{y}) = 1$, $f(x, \mathbf{y}) = (f(\alpha(\ell(x)-1), \mathbf{y})+1) \bmod n$. Hence, $f(x, \mathbf{y}) = \sharp\{t \in \{-1, \ldots, \ell(x)-1\} : k(\alpha(t), \mathbf{y}) \neq 0\}$ MOD n, where $k(\alpha(-1), \mathbf{y}) = g(\mathbf{y})$. The corresponding computation is doable using constant-depth circuits including Mod n gates:

- In the beginning compute $g(\mathbf{y})$ and, for any $t \in \{0, \ldots, \ell(x)-1\}$, $k(\alpha(t), \mathbf{y})$'s independently and in parallel. This can be done in $\mathbf{FAC}^0[\mathbf{n}]$ by hypothesis.
- In one step, compute their sum modulo n, by using a single Mod n gate. ◀

The following proof is simply a special case of the one below, showing that a alternative version of the proof of Prop. 11 can be provided.

Alternative proof of Prop. 11. If $k(\alpha(\ell(x)-1), \mathbf{y}) = 0$, $f(x, \mathbf{y}) = f(\alpha(\ell(x)-1), \mathbf{y})$. If $k(\alpha(\ell(x)-1), \mathbf{y}) = 1$, there are two possible cases:

$$\begin{aligned} f(x, \mathbf{y}) &= f(\alpha(\ell(x)-1), \mathbf{y}) - 2f(\alpha(\ell(x)-1), \mathbf{y}) + 1 \\ &= \begin{cases} 1 & \text{if } f(\alpha(\ell(x)-1), \mathbf{y}) = 0 \\ 0 & \text{if } f(\alpha(\ell(x)-1), \mathbf{y}) = 1. \end{cases} \end{aligned}$$

Clearly, for any x and $\mathbf{y}$, $f(x, \mathbf{y}) \leq 1$. Since for $k(t, \mathbf{y}) = 0$, we can step t, $f(x, \mathbf{y}) = \sharp\{t \in \{-1, \ldots, \ell(x)-1\} : k(\alpha(t), \mathbf{y}) \neq 0\} \bmod 2$, where $k(\alpha(-1), \mathbf{y}) = g(\mathbf{y})$. The corresponding computation is feasible in constant-depth for polynomial-sized unbounded fan-in circuits with Mod 2 gates:

- In the beginning compute $g(\mathbf{y})$ and, for any $t \in \{0, \ldots, \ell(x)-1\}$, $k(\alpha(t), \mathbf{y})$'s independently and in parallel. This can be done in $\mathbf{FAC}^0[\mathbf{2}]$ by hypothesis.
- In one step, compute their sum modulo 2, by using a single Mod 2 gate. ◀

Proof of Theorem 19. ($\supseteq$) For functions and schemas in $\mathbb{ACDL}$ the proof is as in [3], while for ℓ-nODE it is due to Prop. 18. ($\subseteq$) Computation through constant-depth, polynomial-size circuits is captured by $\mathbb{ACDL}$, while counting using Mod n gates can be obtained by a function $\mathsf{cmodn}(x, x)$ defined by ℓ-nODE from g and k, such that $g(\mathbf{y}) = 0$ and $k(x, \mathbf{y}) = \mathsf{BIT}(\ell(x), \mathbf{y})$, i.e., as the solution of the IVP with initial value $\mathsf{cmodn}(0, \mathbf{y}) = 0$ and such that $\frac{\partial \mathsf{cmodn}(x, \mathbf{y})}{\partial \ell} = -n \times \left\lfloor \frac{\mathsf{cmodn}(x, \mathbf{y})}{n-1} \right\rfloor \times \mathsf{BIT}(\ell(x), \mathbf{y}) + \mathsf{BIT}(\ell(x), \mathbf{y})$ (see App. A.1.1). ◀

Proof of Remark 20. The ℓ-2ODE is a special case of ℓ-nODE such that $n = 2$,

$$\begin{aligned}\frac{\partial f(x,\mathbf{y})}{\partial \ell} &= -2 \times k(x,\mathbf{y}) \times \lfloor f(x,\mathbf{y}) \rfloor + k(x,\mathbf{y}) \\ &= -2 \times k(x,\mathbf{y}) \times f(x,\mathbf{y}) + k(x,\mathbf{y}).\end{aligned}$$

Indeed, for any x and $\mathbf{y}$, $f(x,\mathbf{y})$ takes only values 0 or 1. ◀

A.2 Proofs from Section 3.3

▶ **Proposition 41.** *If f is defined by ℓ-0ODE from g and k computable in $\mathbf{FTC}^0$, then f is also computable in $\mathbf{FTC}^0$.*

Proof. By Df. 3, $f(x,\mathbf{y}) = \sum_{u=-1}^{\ell(x)-1} k(\alpha(u),\mathbf{y})$, with the convention that $k(\alpha(-1),\mathbf{y}) = g(\mathbf{y})$. This is nothing but an iterated addition, which is computable in $\mathbf{FTC}^0$ [30]. ◀

▶ Remark 42. Actually closure of ℓ-0ODE can be generalized so that $k : \mathbb{N}^{k+1} \to \mathbb{N}$ and the closure property is established in the same way.

Proof. The proof is precisely as for Prop. 41, but here really exploiting the power of iterated addition (and not simply iterated bit-addition). ◀

Proof of Th. 23. ($\supseteq$) Basic functions and schemas other than ℓ-0ODE are already in $\mathbb{ACDL}$, plus the closure property of ℓ-0ODE (Prop. 41).
($\subseteq$) Due to the equivalence between this ODE-based function algebra and $\mathbb{TCDL}$ (characterizing $\mathbf{FTC}^0$) [3]. ◀

Proof of Prop. 24. Let $\mathsf{bcount}(x,y)$ be defined as the solution of the IVP below:

$$\begin{aligned}f(0,y) &= 0 \\ \frac{\partial f(x,y)}{\partial \ell} &= \mathsf{BIT}(x,y).\end{aligned}$$

Clearly, this is an instance of ℓ-0ODE and, since BIT is in $\mathcal{B}_0$, it is definable in $[\mathcal{B}_0; \circ, \ell\text{-ODE}_1, \ell\text{-0ODE}]$. As, by Df. 3, the solution of this system is $f(x,\mathbf{y}) = \sum_{u=0}^{\ell(x)} \mathsf{BIT}(x,y)$, when computing $\mathsf{bc}(x) = \mathsf{bcount}(x,x)$ we are precisely computing BCOUNT. Then, the function $\mathsf{mod_n}$ computing the sum modulo n of the bits in the binary representation of its input is obtained by simply considering computing $\mathsf{bc}(x) - \lfloor \frac{\mathsf{bc}(x)}{n} \rfloor \times n$. All operations involved are in $[\mathcal{B}_0; \circ, \ell\text{-ODE}_1, \ell\text{-0ODE}]$ as, in particular, multiplication and integer division are known to be in $\mathbf{FTC}^0$ [23] and the given algebra has been proved complete with respect to this class (Th. 23). ◀

Due to Remark 42, the same characterization can be obtained considering the closure of $\mathcal{B}_0$ under $\circ$, ℓ-ODE$_1$ and *generalized* ℓ-0ODE.