Towards a unified perspective in circuit complexity via discrete ODEs

(joint works with A. Durand, J. Kontinen, R. Li)

Melissa Antonelli
Universität Tübingen

IRN $\langle$L|I$\rangle$

International Research Network: Logic and Interaction

Lyon, 23-24 June 2026

ROADMAP

- INTRODUCTION
- FROM BOUNDED RECURSION TO DISCRETE ODES
- SMALL CIRCUIT CLASSES (AND CORRESPONDING ALGEBRAS)
- A NEW FRAMEWORK IN CIRCUIT COMPLEXITY
- FUTURE PERSPECTIVES

ROADMAP

- INTRODUCTION
- FROM BOUNDED RECURSION TO DISCRETE ODEs
- SMALL CIRCUIT CLASSES (AND CORRESPONDING ALGEBRAS)
- A NEW FRAMEWORK IN CIRCUIT COMPLEXITY
- FUTURE PERSPECTIVES

ROADMAP

- INTRODUCTION
- FROM BOUNDED RECURSION TO DISCRETE ODES
- SMALL CIRCUIT CLASSES (AND CORRESPONDING ALGEBRAS)
- A NEW FRAMEWORK IN CIRCUIT COMPLEXITY
- FUTURE PERSPECTIVES

ROADMAP

- INTRODUCTION
- FROM BOUNDED RECURSION TO DISCRETE ODES
- SMALL CIRCUIT CLASSES (AND CORRESPONDING ALGEBRAS)
- A NEW FRAMEWORK IN CIRCUIT COMPLEXITY
- FUTURE PERSPECTIVES

ROADMAP

- INTRODUCTION
- FROM BOUNDED RECURSION TO DISCRETE ODEs
- SMALL CIRCUIT CLASSES (AND CORRESPONDING ALGEBRAS)
- A NEW FRAMEWORK IN CIRCUIT COMPLEXITY
- FUTURE PERSPECTIVES

ROADMAP

- INTRODUCTION
- FROM BOUNDED RECURSION TO DISCRETE ODES
- SMALL CIRCUIT CLASSES (AND CORRESPONDING ALGEBRAS)
- A NEW FRAMEWORK IN CIRCUIT COMPLEXITY
- FUTURE PERSPECTIVES

INTRODUCTION

(IMPLICIT) COMPUTATIONAL COMPLEXITY

(IMPLICIT) COMPUTATIONAL COMPLEXITY

J. Hartmanis & R. Stearns, *On the Computational Complexity of Algorithms*, Trans. Amer. Math. Soc., 117, pp. 285-306, 1965.

IMPLICIT COMPUTATIONAL COMPLEXITY

- recursion theory

restricting the primitive recursion schema

IMPLICIT COMPUTATIONAL COMPLEXITY

- recursion theory
- descriptive complexity

considering expressibility in logical systems

IMPLICIT COMPUTATIONAL COMPLEXITY

- recursion theory
- descriptive complexity
- proof theory

relying on the correspondence between proofs and programs

IMPLICIT COMPUTATIONAL COMPLEXITY

- **recursion theory**
- descriptive complexity
- proof theory

A NEW FRAMEWORK IN COMPLEXITY

A NEW FRAMEWORK IN COMPLEXITY

THE CONTEXT : IMPLICIT COMPLEXITY

$$\mathbf{FP} = \mathcal{A}_{Cob}$$

functions computable in polynomial time

based on bounded recursion on notation [Cobham64]

A NEW FRAMEWORK IN COMPLEXITY

THE CONTEXT : IMPLICIT COMPLEXITY

$$\textbf{FP} = \mathbb{LDL}$$

functions computable in polynomial time

based on the "linear-length ODE" schema [BournezDurand19]

A NEW FRAMEWORK IN COMPLEXITY

THE GOAL

(Implicit) characterizations of small circuit classes based on discrete ODE-schemas.

$\mathbf{FAC}^k$, $\mathbf{FTC}^k$, $\mathbf{FNC}^k$

A NEW FRAMEWORK TO COMPLEXITY

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FNC}^k, \mathbf{FAC}^k, \mathbf{FTC}^k = ?$$

FROM BOUNDED RECURSION TO DISCRETE ODES

Starting point

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FNC}^k, \mathbf{FAC}^k, \mathbf{FTC}^k = ?$$

Starting point

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FNC}^k, \mathbf{FAC}^k, \mathbf{FTC}^k = ?$$

STARTING POINT

O. Bournez and A. Durand, *Recursion Schemes, Discrete Differential Equations and Characterization of Polynomial Time Computations*, MFCS, 23, pp. 1-14, 2019.
O. Bournez and A. Durand, *A Characterization of Functions over the Integers Computable in Polynomial Time Using Discrete Ordinary Differential Equations*, Computational Complexity, 32(2), 2023.

Starting point

O. Bournez and A. Durand, *Recursion Schemes, Discrete Differential Equations and Characterization of Polynomial Time Computations*, MFCS, 23, pp. 1-14, 2019.
O. Bournez and A. Durand, *A Characterization of Functions over the Integers Computable in Polynomial Time Using Discrete Ordinary Differential Equations*, Computational Complexity, 32(2), 2023.

Discrete ordinary differential equations...

- natural tool for algorithm design
- (implicit) characterization for complexity classes

STARTING POINT

O. Bournez and A. Durand, *Recursion Schemes, Discrete Differential Equations and Characterization of Polynomial Time Computations*, MFCS, 23, pp. 1-14, 2019.
O. Bournez and A. Durand, *A Characterization of Functions over the Integers Computable in Polynomial Time Using Discrete Ordinary Differential Equations*, Computational Complexity, 32(2), 2023.

Discrete ordinary differential equations...

- natural tool for algorithm design
- (implicit) characterization for complexity classes (in particular for **FP**).

CAPTURING CLASSES VIA RECURSION SCHEMAS

CAPTURING CLASSES VIA RECURSION SCHEMAS

PRIMITIVE RECURSION (PR)

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

$$\mathbf{PR} = [0, \pi_i^p, s; \circ, \mathrm{PR}]$$

Capturing complexity via limited recursion schemas

Primitive recursion (PR)

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

restrictions on recursion to capture complexity classes

$$\mathbf{PR} = [0, \pi_i^p, s; \circ, \mathrm{PR}]$$

Capturing complexity via limited recursion schemas

Bounded recursion on notation (BRN)

$i \in \{0, 1\}$

$$
\begin{aligned}
f(0, \mathbf{y}) &= g(\mathbf{y}) \\
f(s_i(x), \mathbf{y}) &= h_i(x, \mathbf{y}, f(x, \mathbf{y})) \\
f(x, \mathbf{y}) &\leq k(x, \mathbf{y}) \quad \text{for all } x, \mathbf{y}
\end{aligned}
$$

- explicit bound
- binary successor function

Capturing complexity via limited recursion schemas

Bounded recursion on notation (BRN)

$i \in \{0, 1\}$

does not grow too fast

$$f(0, \mathbf{y}) = g(\mathbf{y})$$
$$f(s_i(x), \mathbf{y}) = h_i(x, \mathbf{y}, f(x, \mathbf{y}))$$
$$f(x, \mathbf{y}) \leq k(x, \mathbf{y}) \quad \text{for all } x, \mathbf{y}$$

- explicit bound
- binary successor function

Capturing complexity via limited recursion schemas

Bounded recursion on notation (BRN)

$i \in \{0, 1\}$

$$f(0, \mathbf{y}) = g(\mathbf{y})$$
$$f(s_i(x), \mathbf{y}) = h_i(x, \mathbf{y}, f(x, \mathbf{y}))$$
$$f(x, \mathbf{y}) \leq k(x, \mathbf{y}) \quad \text{for all } x, \mathbf{y}$$

$s_i(x) = 2x + i$

- explicit bound
- binary successor function

control the number of recursion steps

Capturing complexity via limited recursion schemas

Bounded recursion on notation (BRN)

$i \in \{0, 1\}$

$$
\begin{aligned}
f(0, \mathbf{y}) &= g(\mathbf{y}) \\
f(s_i(x), \mathbf{y}) &= h_i(x, \mathbf{y}, f(x, \mathbf{y})) \\
f(x, \mathbf{y}) &\leq k(x, \mathbf{y}) \quad \text{for all } x, \mathbf{y}
\end{aligned}
$$

- explicit bound
- binary successor function

$x\#y = 2^{\ell(x)\times\ell(y)}$

$$\mathbf{FP} = [0, \pi_i^p, s_0, s_1, \#; \circ, \mathrm{BRN}]$$

FROM BOUNDED RECURSION TO ODE

Discrete ODE-based characterization of **FP** :

- no explicit bound in the recursion
- no specific role or type for variables

From bounded recursion to ODE

$\Delta f(x) = f(x+1) - f(x)$

Discrete ODE-based characterization of **FP** :

- no explicit bound in the recursion
- no specific role or type for variables

From bounded recursion to ODE

Discrete ODE-based characterization of **FP** :

- no explicit bound in the recursion
- no specific role or type for variables

as instead based on

- derivation along function
- linearity

FROM BOUNDED RECURSION TO ODE

Discrete ODE-based characterization of **FP** :

- no explicit bound in the recursion
- no specific role or type for variables

to control the number of computation steps

as instead based on

- derivation along function : $\mathcal{L}$-ODE
- linearity

FROM BOUNDED RECURSION TO ODE

Discrete ODE-based characterization of **FP** :

- no explicit bound in the recursion
- no specific role or type for variables

as instead based on

- derivation along function : $\mathcal{L}$-ODE

  to control the object size
- linearity : (essentially) linear ODEs

CAPTURING FP : LINEAR $\mathcal{L}$-ODE

DEFINITION (LINEAR $\mathcal{L}$-ODE). Function f is linear $\mathcal{L}$-ODE definable from u, g and h if it is the solution of IVP :

$$f(0, \mathbf{y}) = g(\mathbf{y})$$
$$\frac{\partial f(x, \mathbf{y})}{\partial \mathcal{L}} = u\big(x, \mathbf{y}, f(x, \mathbf{y}), h(x, \mathbf{y})\big),$$

where u is ESSENTIALLY LINEAR in $f(x, \mathbf{y})$.

CAPTURING FP : LINEAR $\mathcal{L}$-ODE

DEFINITION (LINEAR $\mathcal{L}$-ODE). Function f is linear $\mathcal{L}$-ODE definable from u, g and h if it is the solution of IVP :

The discrete derivative of $f(x)$ is $f(x+1) - f(x)$

$$f(0, \mathbf{y}) = g(\mathbf{y})$$

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

where u is ESSENTIALLY LINEAR in $f(x, \mathbf{y})$.

CAPTURING FP : LINEAR $\mathcal{L}$-ODE

DEFINITION (LINEAR $\mathcal{L}$-ODE). Function f is linear $\mathcal{L}$-ODE definable from u, g and h if it is the solution of IVP :

$$f(0,\mathbf{y}) = g(\mathbf{y})$$

$$\frac{\partial f(x,\mathbf{y})}{\partial \mathcal{L}} = u(x,\mathbf{y}, f(x,\mathbf{y}), h(x,\mathbf{y})),$$

where u is ES

$$f(x+1,\mathbf{y}) = f(x,\mathbf{y}) + (\mathcal{L}(x+1,\mathbf{y}) - \mathcal{L}(x,\mathbf{y})) \times u(x,\mathbf{y}, f(x,\mathbf{y}), h(x,\mathbf{y}))$$

Capturing FP : linear $\mathcal{L}$-ODE

Definition (Linear $\mathcal{L}$-ODE). Function f is linear $\mathcal{L}$-ODE definable from u, g and h if it is the solution of IVP :

$$f(0, \mathbf{y}) = g(\mathbf{y})$$

$$\frac{\partial f(x, \mathbf{y})}{\partial \mathcal{L}} = u(x, \mathbf{y}, f(x, \mathbf{y}), h(x, \mathbf{y})),$$

where u is ESS

The value of $f(x, \mathbf{y})$ changes only when the value of $\mathcal{L}(x, \mathbf{y})$ does.

CAPTURING FP : LINEAR $\mathcal{L}$-ODE

DEFINITION (LINEAR $\mathcal{L}$-ODE). Function f is linear $\mathcal{L}$-ODE definable from u, g and h if it is the solution of IVP :

$$f(0, \mathbf{y}) = g(\mathbf{y})$$
$$\frac{\partial f(x, \mathbf{y})}{\partial \mathcal{L}} = u\big(x, \mathbf{y}, f(x, \mathbf{y}), h(x, \mathbf{y})\big),$$

where u is ESSENTIALLY LINEAR in $f(x, \mathbf{y})$.

CAPTURING FP : LINEAR $\mathcal{L}$-ODE

DEFIN[...] d h if it
is the

A sg-polynomial expression is an expression over the signature $\{+, -, \times, \mathrm{sg}\}$.
A sg-polynomial expression P is ESSENTIALLY LINEAR IN x if $\mathbf{deg}(x, P) = 1$, i.e. there exist sg-expressions Q_1, Q_2 such that $P = Q_1 \times x + Q_2$ and $\mathbf{deg}(x, Q_1) = \mathbf{deg}(x, Q_2) = 0$.
The degree $\mathbf{deg}(x, P)$ of a term $x \in X$ in P :

- $\mathbf{deg}(x, x) = 1$ and, for $x' \in X \cup \mathbb{Z}$ s.t. $x' \neq x$, $\mathbf{deg}(x, x') = 0$
- $\mathbf{deg}(x, P + Q) = \mathbf{deg}(x, P - Q) = \max\{\mathbf{deg}(x, P), \mathbf{deg}(x, Q)\}$
- $\mathbf{deg}(x, P \times Q) = \mathbf{deg}(x, P) + \mathbf{deg}(x, Q)$
- $\mathbf{deg}(x, \mathrm{sg}(P)) = 0$.

$\partial\mathcal{L}$

where u is ESSENTIALLY LINEAR in $f(x, \mathbf{y})$.

CAPTURING FP : LINEAR $\mathcal{L}$-ODE

A sg-polynomial expression is an expression over the signature $\{+, -, \times, \mathrm{sg}\}$.
A sg-polynomial expression P is ESSENTIALLY LINEAR IN x if $\mathbf{deg}(x, P) = 1$, i.e. there exist sg-expressions Q_1, Q_2 such that $P = Q_1 \times x + Q_2$ and $\mathbf{deg}(x, Q_1) = \mathbf{deg}(x, Q_2) = 0$.
The degree $\mathbf{deg}(x, P)$ of a term $x \in X$ in P :

- $\mathbf{deg}(x, x) = 1$ and, for $x' \in X \cup \mathbb{Z}$ s.t. $x' \neq x$, $\mathbf{deg}(x, x') = 0$
- $\mathbf{deg}(x, P + Q) = \mathbf{deg}(x, P - Q) = \max\{\mathbf{deg}(x, P), \mathbf{deg}(x, Q)\}$
- $\mathbf{deg}(x, P \times Q) = \mathbf{deg}(x, P) + \mathbf{deg}(x, Q)$
- $\mathbf{deg}(x, \mathrm{sg}(P)) = 0$.

DEFIN... d h if it is the ...

$\partial\mathcal{L}$

where u is ESSENTIALLY LINEAR in $f(x, \mathbf{y})$.

Capturing FP : linear $\mathcal{L}$-ODE

Definition (Linear $\mathcal{L}$-ODE). Function f is linear $\mathcal{L}$-ODE definable from u, g and h if it is the solution of IVP :

$$\begin{aligned} f(0,\mathbf{y}) &= g(\mathbf{y}) \\ \frac{\partial f(x,\mathbf{y})}{\partial \mathcal{L}} &= A(x,\mathbf{y},f,h) \times f(x,\mathbf{y}) + B(x,\mathbf{y},f,h) \end{aligned}$$

where $\mathbf{u}$ occurs in A and B under the scope of sg.

CAPTURING FP : LINEAR ℓ-ODE

DEFINITION (LINEAR LENGTH-ODE, ℓ-ODE). Function f is linear length-ODE definable from u, g and h, if it is the solution of IVP :

$$
\begin{aligned}
f(0, \mathbf{y}) &= g(\mathbf{y}) \\
\frac{\partial f(x, \mathbf{y})}{\partial \ell} &= u\big(x, \mathbf{y}, f(x, \mathbf{y}), h(x, \mathbf{y})\big),
\end{aligned}
$$

where u is ESSE

$\ell(x)$ returns the length of x written in binary

CAPTURING FP : LINEAR ℓ-ODE

DEFINITION (LINEAR LENGTH-ODE, ℓ-ODE). Function f is linear length-ODE definable from u, g and h, if it is the solution of IVP :

$$f(0, \mathbf{y}) = g(\mathbf{y})$$

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

where u is ESSENTIAL…

The solution of this system is

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

where $\alpha(z) = 2^z - 1$, $\prod_x^{x-1} \kappa(x) = 1$ and $B(\alpha(-1), \cdot) = f(0, \mathbf{y})$.

CAPTURING FP : $\mathbb{LDL}$

LINEAR ℓ-ODE

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

- derivation along to the length function ℓ
- linearity

$$\mathbf{FP} = [0,1,\ell,\mathrm{sg},+,-,\times,\pi^p_i;\circ,\ell\text{-ODE}]$$

CAPTURING FP : $\mathbb{LDL}$

LINEAR ℓ-ODE

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

- derivation along to the length function ℓ
- linearity

$\mathrm{sg}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$

$$\mathbf{FP} = [0, 1, \ell, \mathrm{sg}, +, -, \times, \pi_i^p; \circ, \ell\text{-ODE}]$$

$\ell(0) = 0$ and $\ell(x) = \lceil \log_2(x+1) \rceil$

composition

CAPTURING FP : $\mathbb{LDL}$

LINEAR ℓ-ODE

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

to control the number of computation steps

- derivation along to the length function ℓ
- linearity

to control the object size

$$\mathbf{FP} \supseteq [\mathbf{0}, \mathbf{1}, \ell, \mathrm{sg}, +, -, \times, \pi_i^p; \circ, \ell\text{-ODE}]$$

CAPTURING FP : $\mathbb{LDL}$

LINEAR ℓ-ODE

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

- derivation along to the length function ℓ
- linearity

$$\mathbf{FP} \subseteq [0, 1, \ell, \mathrm{sg}, +, -, \times, \pi_i^p; \circ, \ell\text{-ODE}]$$

"encoding" computation of a poly-time register machine via ℓ-ODE

ON SMALL CIRCUIT CLASSES (AND CORRESPONDING ALGEBRAS)

Capturing small circuit classes

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FNC}^k, \mathbf{FAC}^k, \mathbf{FTC}^k = ?$$

Capturing small circuit classes

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FNC}^k, \mathbf{FAC}^k, \mathbf{FTC}^k = ?$$

ON PARALLEL COMPUTATION : INTUITION

In parallel computation multiple operations can be executed simultaneously

ON PARALLEL COMPUTATION : INTUITION

In parallel computation multiple operations can be executed simultaneously

A parallel algorithm is efficient when :

ON PARALLEL COMPUTATION : INTUITION

In parallel computation multiple operations can be executed simultaneously

A parallel algorithm is efficient when :

i.e. polynomial in the length of the input

- the number of processors is not too high

On parallel computation : intuition

In parallel computation multiple operations can be executed simultaneously

A parallel algorithm is efficient when :

- the number of processors is not too high
- speed-up with respect to its (efficient) sequential counterpart.

SMALL CIRCUIT CLASSES : $\mathbf{FAC}^i$

DEFINITION (CLASS $\mathbf{FAC}^i$). $\mathbf{AC}^i$ is the class of languages recognized by a **Dlogtime**-uniform family of unbounded Boolean circuit of polynomial size and $O((\log n)^i)$ depth. We denote by $\mathbf{FAC}^i$ the corresponding function class.

circuits over (standard) unbounded fan-in basis $\{\neg, (\wedge^n)_{n\in\mathbb{N}}, (\vee^n)_{n\in\mathbb{N}}\}$

SMALL CIRCUIT CLASSES : $\mathbf{FAC}^0$

DEFINITION (CLASS $\mathbf{FAC}^0$). $\mathbf{AC}^0$ is the class of languages recognized by a **Dlogtime**-uniform family of unbounded Boolean circuit of polynomial size and constant depth. We denote by $\mathbf{FAC}^0$ the corresponding function class.

SMALL CIRCUIT CLASSES : $\mathbf{FNC}^i$

DEFINITION (CLASS $\mathbf{FNC}^i$). $\mathbf{NC}^i$ is the class of languages recognized by a **Dlogtime**-uniform family of bounded Boolean circuit of polynomial size and $O((\log n)^i)$ depth. We denote by $\mathbf{FNC}^i$ the corresponding function class.

circuits over (standard) bounded fan-in basis $\{\neg, \wedge, \vee\}$

SMALL CIRCUIT CLASSES

DEFINITION (CLASSES $\mathbf{FAC}^i$ AND $\mathbf{FNC}^i$). $\mathbf{AC}^i$ (resp., $\mathbf{NC}^i$) is the class of languages recognized by a **Dlogtime**-uniform family of unbounded (resp., bounded) Boolean circuit of polynomial size and $O((\log n)^i)$ depth. We denote by $\mathbf{FAC}^i$ the corresponding function class.

DEFINITION (CLASSES $\mathbf{FACC}[\mathbf{p}]$ AND $\mathbf{FTC}^0$) For $p \in \mathbb{N}$, $\mathbf{ACC}[\mathbf{p}]$ (resp., $\mathbf{TC}^0$) is the class of languages recognized by a uniform unbounded Boolean circuit family *including* MOD *p gates* (resp., MAJ *gates*) of polynomial size and constant depth. We denote by $\mathbf{FACC}[\mathbf{p}]$ (resp., $\mathbf{FTC}^0$) the corresponding function class.

SMALL CIRCUIT CLASSES

DEFINITION (CLASSES **FAC**^{i} AND **FNC**^{i}). **AC**^{i} (resp., **NC**^{i}) is the class of languages recognized by a **Dlogtime**-uniform family of unbounded (resp., bounded) Boolean circuit of polynomial size and $O((\log n)^i)$ depth. We denote by **FAC**^{i} the corresponding function class.

DEFINITION (CLASSES **FACC[p]** AND **FTC**^{0}) For $p \in \mathbb{N}$, **ACC[p]** (resp., **TC**^{0}) is the class of languages recognized by a uniform unbounded Boolean circuit family *including* MOD *p gates* (resp., MAJ *gates*) of polynomial size and constant depth. We denote by **FACC[p]** (resp., **FTC**^{0}) the corresponding function class.

MOD p outputs the sum of its inputs modulo p

SMALL CIRCUIT CLASSES

DEFINITION (CLASSES $\mathbf{FAC}^i$ AND $\mathbf{FNC}^i$). $\mathbf{AC}^i$ (resp., $\mathbf{NC}^i$) is the class of languages recognized by a **Dlogtime**-uniform family of unbounded (resp., bounded) Boolean circuit of polynomial size and $O((\log n)^i)$ depth. We denote by $\mathbf{FAC}^i$ the corresponding function class.

DEFINITION (CLASSES **FACC[p]** AND $\mathbf{FTC}^0$) For $p \in \mathbb{N}$, **ACC[p]** (resp., $\mathbf{TC}^0$) is the class of languages recognized by a uniform unbounded Boolean circuit family *including* MOD *p gates* (resp., MAJ *gates*) of polynomial size and constant depth. We denote by **FACC[p]** (resp., $\mathbf{FTC}^0$) the corresponding function class.

MAJ outputs 1 when the majority of its inputs are 1's

SMALL CIRCUIT CLASSES

DEFINITION (CLASSES $\mathbf{FAC}^i$ AND $\mathbf{FNC}^i$). $\mathbf{AC}^i$ (resp., $\mathbf{NC}^i$) is the class of languages recognized by a **Dlogtime**-uniform family of unbounded (resp., bounded) Boolean circuit of polynomial size and $O((\log n)^i)$ depth. We denote by $\mathbf{FAC}^i$ the corresponding function class.

DEFINITION (CLASSES **FACC[p]** AND $\mathbf{FTC}^0$) For $p \in \mathbb{N}$, **ACC[p]** (resp., $\mathbf{TC}^0$) is the class of languages recognized by a uniform unbounded Boolean circuit family *including* MOD *p gates* (resp., MAJ *gates*) of polynomial size and constant depth. We denote by **FACC[p]** (resp., $\mathbf{FTC}^0$) the corresponding function class.

$\mathbf{FAC}^0 \subsetneq \mathbf{FACC[2]}$

Small circuit classes

DEFINITION (CLASSES $\mathbf{FAC}^i$ AND $\mathbf{FNC}^i$). $\mathbf{AC}^i$ (resp., $\mathbf{NC}^i$) is the class of languages recognized by a **Dlogtime**-uniform family of unbounded (resp., bounded) Boolean circuit of polynomial size and $O((\log n)^i)$ depth. We denote by $\mathbf{FAC}^i$ the corresponding function class.

DEFINITION (CLASSES $\mathbf{FACC[p]}$ AND $\mathbf{FTC}^0$) For $p \in \mathbb{N}$, $\mathbf{ACC[p]}$ (resp., $\mathbf{TC}^0$) is the class of languages recognized by a uniform unbounded Boolean circuit family *including* MOD *p gates* (resp., MAJ *gates*) of polynomial size and constant depth. We denote by $\mathbf{FACC[p]}$ (resp., $\mathbf{FTC}^0$) the corresponding function class.

$\mathbf{FACC[p]} \subsetneq \mathbf{FTC}^0$

SMALL CIRCUIT CLASSES

DEFINITION (CLASSES $\mathbf{FAC}^i$ AND $\mathbf{FNC}^i$). $\mathbf{AC}^i$ (resp., $\mathbf{NC}^i$) is the class of languages recognized by a **Dlogtime**-uniform family of unbounded (resp., bounded) Boolean circuit of polynomial size and $O((\log n)^i)$ depth. We denote by $\mathbf{FAC}^i$ the corresponding function class.

DEFINITION (CLASSES $\mathbf{FACC}[\mathbf{p}]$ AND $\mathbf{FTC}^0$) For $p \in \mathbb{N}$, $\mathbf{ACC}[\mathbf{p}]$ (resp., $\mathbf{TC}^0$) is the class of languages recognized by a uniform unbounded Boolean circuit family *including* MOD *p gates* (resp., MAJ *gates*) of polynomial size and constant depth. We denote by $\mathbf{FACC}[\mathbf{p}]$ (resp., $\mathbf{FTC}^0$) the corresponding function class.

ADD, LOGITADD

ITADD, BCOUNT, MULT, SORT

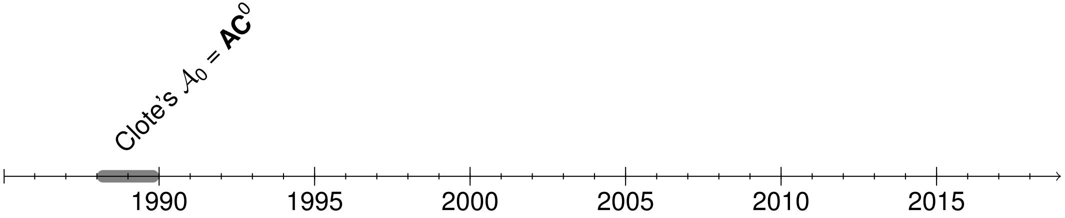
$$\mathbf{FAC}^0 \subsetneq \mathbf{FACC}[2] \subsetneq \mathbf{FTC}^0 \subseteq \mathbf{FNC}^1 \subseteq \mathbf{FAC}^1 \subseteq \mathbf{FNC}^2 \dots$$

PARITY

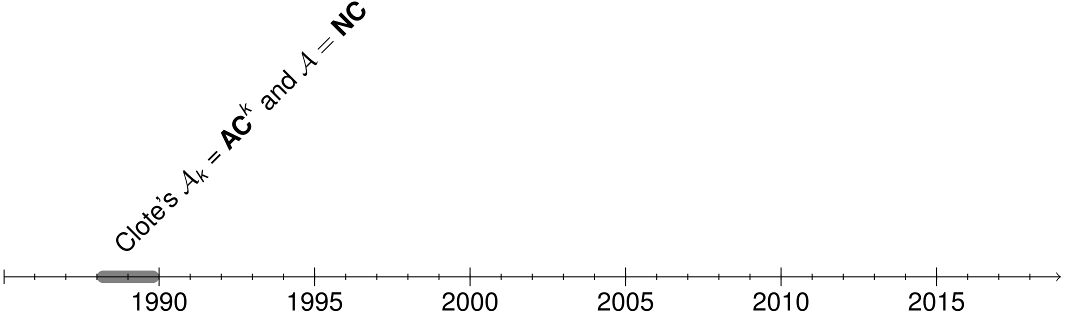
Capturing small circuit classes

$$\mathbf{FP} = \mathcal{A}_{Cob} \quad \rightsquigarrow \quad \mathbf{FAC}^0, \mathbf{FAC}^k, \mathbf{FNC} = \mathcal{A}_0, \mathcal{A}_k, \mathcal{A}$$

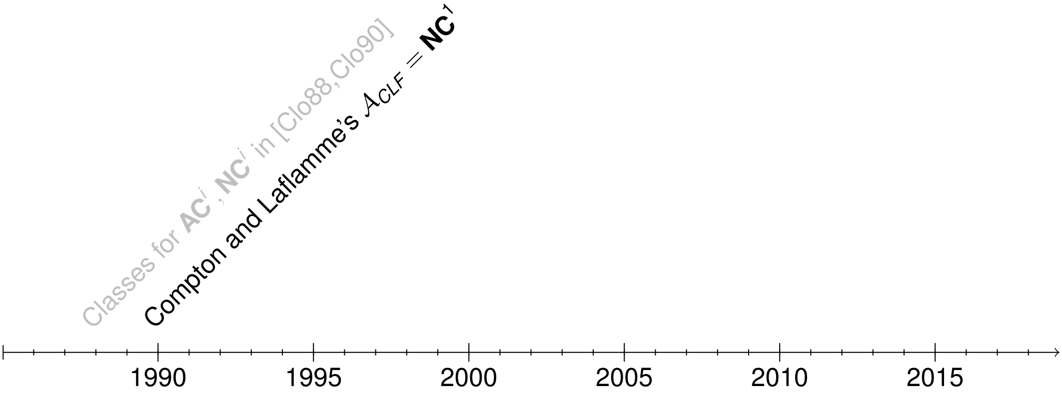
State of the Art



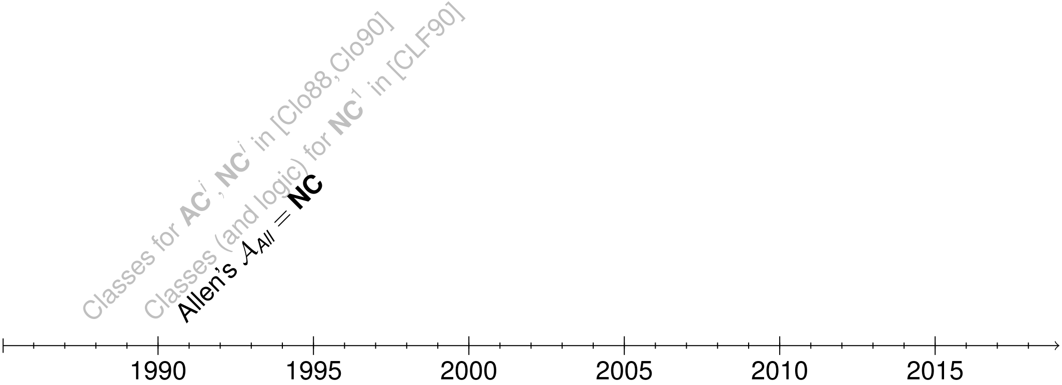
STATE OF THE ART



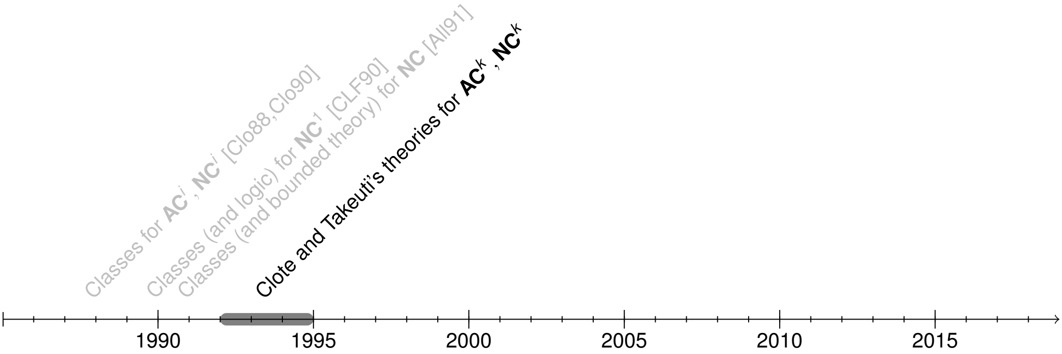
State of the Art



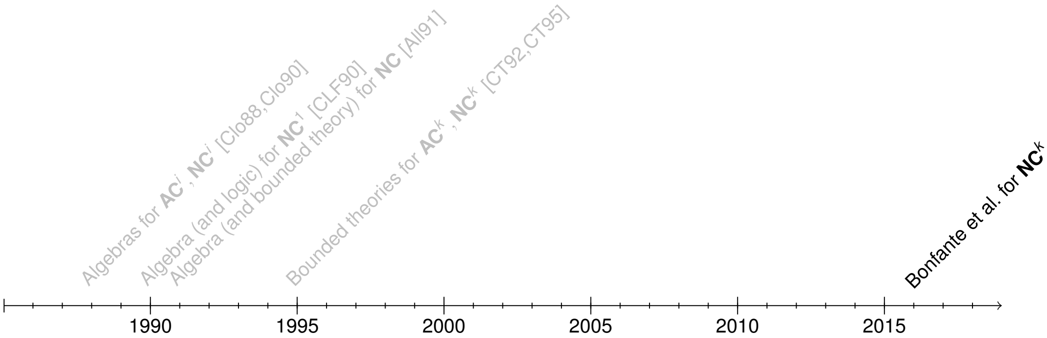
STATE OF THE ART



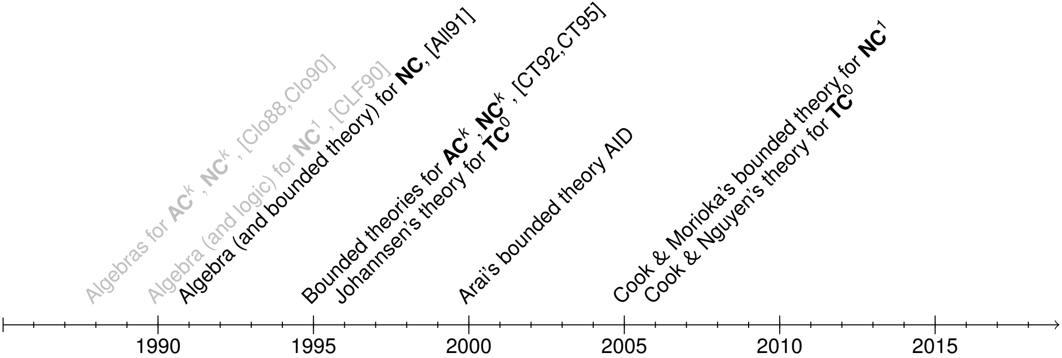
STATE OF THE ART



State of the Art



State of the Art



CAPTURING SMALL CIRCUITS VIA LIMITED RECURSION SCHEMAS

CAPTURING SMALL CIRCUITS VIA LIMITED RECURSION SCHEMAS

CONCATENATION RECURSION ON NOTATION (CRN)

$i \in \{0,1\}$

$$f(0,\mathbf{y}) = g(\mathbf{y})$$
$$f(s_i(x),\mathbf{y}) = s_{h_i}(f(x,\mathbf{y}))$$

For $h_i(x,\mathbf{y}) \leq 1$

Capturing small circuits via limited recursion schemas

Concatenation recursion on notation (CRN)

$i \in \{0, 1\}$

$$f(0, \mathbf{y}) = g(\mathbf{y})$$
$$f(s_i(x), \mathbf{y}) = s_{h_i}(f(x, \mathbf{y}))$$

For $h_i(x, \mathbf{y}) \leq 1$

Clote 1990

$$\mathbf{FAC}^0 = [0, \pi_i^p, s_0, s_1, \mathrm{BIT}, \ell, \#; \circ, \mathrm{CRN}]$$

$s_i(x) = 2x + i$

bit(x,i) yields the i^{th} bit in the binary representation of x

A NEW FRAMEWORK IN CIRCUIT COMPLEXITY

Capturing circuit classes via ODEs

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FNC}^k, \mathbf{FAC}^k, \mathbf{FTC}^k = ?$$

WHY CAPTURING CIRCUIT CLASSES VIA ODEs ?

This framework offers :

- FLEXIBILITY : multiple features we can modify to capture different resources in circuit computation (e.g., constraints on linearity, the function we're deriving along, etc.).
- UNIFORMITY : minimal modification in the basic set of functions and on constraints over discrete ODEs (e.g., over linearity) to jump from one class to the other.

WHY CAPTURING CIRCUIT CLASSES VIA ODES ?

This framework offers :

- FLEXIBILITY : multiple features we can modify to capture different resources in circuit computation (e.g., constraints on linearity, the function we're deriving along, etc.).
- UNIFORMITY : minimal modification in the basic set of functions and on constraints over discrete ODEs (e.g., over linearity) to jump from one class to the other.

Reveal a connection between (constraints on) discrete ODEs computation and circuit resources.

WHY CAPTURING CIRCUIT CLASSES VIA ODES ?

This framework offers :

- FLEXIBILITY : multiple features we can modify to capture different resources in circuit computation (e.g., constraints on linearity, the function we're deriving along, etc.).
- UNIFORMITY : minimal modification in the basic set of functions and on constraints over discrete ODEs (e.g., over linearity) to jump from one class to the other.

Reveal a connection between (constraints on) discrete ODEs computation and circuit resources.

Due to this perspective, we obtain :

- Simplified characterizations for multiple classes
- The very first characterizations for some classes (e.g., **FACC[p]** for $p \notin \{2, 6\}$)
- A uniform picture (possibly scaling to the continuous setting)
- Possible connections with induction schema and bounded arithmetic

CASE STUDY : CAPTURING $\mathbf{FAC}^0$

CASE STUDY : CAPTURING FAC^0

THE KEY : A NEW SCHEMA

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = f(x, \mathbf{y}) + k(x, \mathbf{y})$$

k takes values in $\{0, 1\}$

to control the number of computation steps

- derivation along the length function ℓ

CASE STUDY : CAPTURING FAC^0

THE KEY : A NEW SCHEMA

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = f(x, \mathbf{y}) + k(x, \mathbf{y})$$

k takes values in $\{0, 1\}$

- derivation along the length function ℓ
- restricted form of linearity

to control the object size

CASE STUDY : CAPTURING FAC^0

THE KEY : A NEW SCHEMA

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = f(x, \mathbf{y}) + k(x, \mathbf{y})$$

k takes values in $\{0, 1\}$

- derivation along the length function ℓ
- restricted form of linearity

Intuitively, at each step, we perform a left shift of one unit, optionally adding a bit.

CASE STUDY : CAPTURING FAC^0

THE KEY : A NEW SCHEMA

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = f(x, \mathbf{y}) + k(x, \mathbf{y})$$

k takes values in $\{0, 1\}$

- derivation along the length function ℓ
- restricted form of linearity

Intuitively, at each step, we perform a left shift of one unit, optionally adding a bit.

Case study : capturing FAC^0

The key : a new schema

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = f(x, \mathbf{y}) + k(x, \mathbf{y})$$

k takes values in $\{0, 1\}$

- derivation along the length function ℓ
- restricted form of linearity

Intuitively, at each step, we perform a left shift of one unit, optionally adding a bit.

Case study : capturing FAC^0

The key : a new schema

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

k takes values in $\{0,1\}$

- derivation along the length function ℓ
- restricted form of linearity

Intuitively, at each step, we perform a left shift of one unit, optionally adding a bit.

Example (Function $2^{\ell(x)}$). The function $x \mapsto 2^{\ell(x)}$ can be rewritten as the solution of the IVP below :

$$f(0) = 1$$
$$\frac{\partial f(x)}{\partial \ell} = f(x)$$

CASE STUDY : CAPTURING FAC^0

THE KEY : A NEW SCHEMA

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = f(x, \mathbf{y}) + k(x, \mathbf{y})$$

k takes values in $\{0, 1\}$

- derivation along the length function ℓ
- restricted form of linearity

Intuitively, at each step, we perform a left shift of one unit, optionally adding a bit.

The solution of such system is of the form $f(x) = \prod_{u=0}^{\ell(x)} 2 = 2^{\ell(x)}$

EXAMPLE (FUNCTION $2^{\ell(x)}$). The function $x \mapsto 2^{\ell(x)}$ can be rewritten as the solution of the IVP below :

$$f(0) = 1$$
$$\frac{\partial f(x)}{\partial \ell} = f(x)$$

CASE STUDY : CAPTURING FAC^0

THE KEY : A NEW SCHEMA

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

k takes values in $\{0,1\}$

- derivation along the length function ℓ
- restricted form of linearity

Intuitively, at each step, we perform a left shift of one unit, optionally adding a bit.

EXAMPLE (FUNCTION $2^{\ell(x)}$). The function $x \mapsto 2^{\ell(x)}$ can be r[…]s the solution of the IVP below :

$g(\mathbf{y}) = 1$

$$f(0) = 1$$

$$\frac{\partial f(x)}{\partial \ell} = f(x)$$

$k(x,\mathbf{y}) = 0$

CASE STUDY : CAPTURING FAC^0

THE KEY : A NEW SCHEMA (ℓ-ODE$_1$)

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = f(x, \mathbf{y}) + k(x, \mathbf{y})$$

- derivation along to the length function ℓ
- restricted form of linearity

Intuitively, at each step, we perform a left shift of one unit, possibly adding a bit.

REMARK (SHARPLY BOUNDED QUANTIFICATION). Sharply bounded quantification is captured using the function sg and ℓ-ODE$_1$.

CASE STUDY : CAPTURING $\mathbf{FAC}^0$

THE KEY : A NEW SCHEMA (ℓ-ODE$_1$)

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

EXAMPLE (SHARPLY BOUNDED EXISTENTIAL QUANTIFICATION). Let $R \subseteq \mathbb{N}^{p+1}$ and h_R be its characteristic function. Then, for all x and $\mathbf{y}$,

$$(\exists z \leq \ell(x))R(z,\mathbf{y}) = \mathrm{sg}(f(x,\mathbf{y}))$$

where f is such that

$$\begin{aligned} f(0,\mathbf{y}) &= h_R(0,\mathbf{y}) \\ \frac{f(x,\mathbf{y})}{\partial \ell} &= f(x,\mathbf{y}) + h_R(\ell(x+1),\mathbf{y}). \end{aligned}$$

CASE STUDY : CAPTURING $\mathbf{FAC}^0$

THE KEY : A NEW SCHEMA (ℓ-ODE$_1$)

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

EXAMPLE (SHARPLY BOUNDED EXISTENTIAL QUANTIFICATION). Let $R \subseteq \mathbb{N}^{p+1}$ and h_R be its characteristic function. Then, for all x and $\mathbf{y}$,

$$(\exists z \leq \ell(x))R(z,\mathbf{y}) = \mathrm{sg}(f(x,\mathbf{y}))$$

where f is such that

Intuitively, $f(x,\mathbf{y}) \neq 0$ when, for some $z \leq x$, $R(z,\mathbf{y})$ is satisfied, i.e. $h_R(z,\mathbf{y}) = 1$.

$$f(0,\mathbf{y}) = h_R(0,\mathbf{y})$$

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

CASE STUDY : CAPTURING $\mathbf{FAC}^0$

THE KEY : A NEW SCHEMA (ℓ-ODE$_1$)

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = f(x, \mathbf{y}) + k(x, \mathbf{y})$$

EXAMPLE (SHARPLY BOUNDED EXISTENTIAL QUANTIFICATION). Let $R \subseteq \mathbb{N}^{p+1}$ and h_R be its characteristic function. Then, for all x and $\mathbf{y}$,

$$(\exists z \leq \ell(x))R(z, \mathbf{y}) = \mathrm{sg}(f(x, \mathbf{y}))$$

where f is such that

instance of ℓ-ODE$_1$

$$f(0, \mathbf{y}) = h_R(0, \mathbf{y})$$
$$\frac{f(x, \mathbf{y})}{\partial \ell} = f(x, \mathbf{y}) + h_R(\ell(x+1), \mathbf{y}).$$

CASE STUDY : CAPTURING FAC^0

THE KEY IDEA : NEW SCHEMAS

ℓ-ODE$_1$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

ℓ-ODE$_3$

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

$\lceil \frac{z}{2} \rceil$ is a shorthand for $z - (z \div 2)$

CASE STUDY : CAPTURING $\mathbf{FAC}^0$

THE KEY IDEA : NEW SCHEMAS

ℓ-ODE$_1$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

ℓ-ODE$_3$

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

THE FIRST RESULT [MFCS24]

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FAC}^0 = \mathbb{ACDL}$$

$[0, 1, \ell, \mathrm{sg}, +, -, \div 2, \#, \pi^p_i; \circ, \ell\text{-ODE}_1, \ell\text{-ODE}_3]$

CASE STUDY : CAPTURING $\mathbf{FAC}^0$

THE KEY IDEA : NEW SCHEMAS

ℓ-ODE$_1$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

ℓ-ODE$_3$

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

THE FIRST RESULT [MFCS24]

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FAC}^0 = \mathbb{ACDL}$$

$[0, 1, \ell, \mathrm{sg}, +, -, \times, \pi_i^p; \circ, \ell\text{-ODE}]$

$[0, 1, \ell, \mathrm{sg}, +, -, \div 2, \#, \pi_i^p; \circ, \ell\text{-ODE}_1, \ell\text{-ODE}_3]$

CASE STUDY : CAPTURING $\mathbf{FAC}^0$

THE KEY IDEA : NEW SCHEMAS

ℓ-ODE$_1$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

ℓ-ODE$_3$

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

THE FIRST RESULT [MFCS24]

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FAC}^0 = \mathbb{ACDL}$$

($\supseteq$) Basic functions are computable in $\mathbf{FAC}^0$ and closure under composition and ℓ-ODE$_1$

CASE STUDY : CAPTURING $\mathbf{FAC}^0$

THE KEY IDEA : NEW SCHEMAS

ℓ-ODE$_1$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

ℓ-ODE$_3$

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

THE FIRST RESULT [MFCS24]

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FAC}^0 = \mathbb{ACDL}$$

($\subseteq$) Arithmetization (mimicking functions and schemas of Clote's $\mathcal{A}_0$)

CASE STUDY : CAPTURING FAC^0

THE KEY IDEA : NEW SCHEMAS

ℓ-ODE$_1$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

ℓ-ODE$_3$

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

THE FIRST RESULT [MFCS24]

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FAC}^0 = \mathcal{A}_0 \subseteq \mathbb{ACDL}$$

$[0, s_0, s_1, \mathrm{BIT}, \ell, \#; \circ, \mathrm{CRN}]$

CASE STUDY : CAPTURING $\mathbf{FAC}^0$

THE KEY IDEA : NEW SCHEMAS

ℓ-ODE$_1$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

ℓ-ODE$_3$

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

THE FIRST RESULT [MFCS24]

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FAC}^0 = \mathcal{A}_0 \subseteq \mathbb{ACDL}$$

$[0, s_0, s_1, \mathrm{BIT}, \ell, \#; \circ, \mathrm{CRN}]$

CASE STUDY : CAPTURING $\mathbf{FAC}^0$

THE KEY IDEA : NEW SCHEMAS

ℓ-ODE$_1$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y})$$

ℓ-ODE$_3$

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

THE FIRST RESULT [MFCS24]

$$\mathbf{FP} = \mathbb{LDL} \quad \rightsquigarrow \quad \mathbf{FAC}^0 = \mathcal{A}_0 \subseteq \mathbb{ACDL}$$

$[0, s_0, s_1, \mathrm{BIT}, \ell, \#; \circ, \mathrm{CRN}]$

CAPTURING SMALL CIRCUIT CLASSES VIA ODES

THE KEY : MANY NEW SCHEMAS

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = A(x, \mathbf{y}, f, h) \times f(x, \mathbf{y}) + B(x, \mathbf{y}, f, h)$$

- derivation along the length function ℓ
- various constraints on linearity

THE RESULTS [MFCS24, MFCS25, ICALP26]

Uniform ODE characterizations of classes between $\mathbf{FAC}^0$ and $\mathbf{FNC}^1$ (and with counting gates).

Capturing small circuit classes via ODEs

The key : many new schemas

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = A(x,\mathbf{y},f,h) \times f(x,\mathbf{y}) + B(x,\mathbf{y},f,h)$$

- derivation along the length function ℓ
- various constraints on linearity

The results [MFCS24, MFCS25, ICALP26]

Uniform ODE characterizations of classes between $\mathbf{FAC}^0$ and $\mathbf{FNC}^1$ (and with counting gates).

Capturing small circuit classes via ODEs

The key : many new schemas

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = A(x,\mathbf{y},f,h)\times f(x,\mathbf{y}) + B(x,\mathbf{y},f,h)$$

- derivation along the length function ℓ
- various constraints on linearity

Some of these classes were never characterized by recursion schemas before; namely, all classes counting modulo n (**FACC**[**n**]) except for **FACC**[**2**] and **FACC**[**6**].

The results [MFCS24, MFC

Uniform ODE characterizations of classes between **FAC**^{0} and **FNC**^{1} (and with counting gates).

THE GLOBAL PICTURE (SO FAR) : FP

BOURNEZ & DURAND 2019

$$f(0,\mathbf{y}) = g(\mathbf{y})$$
$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = A(x,\mathbf{y},h,f) \times f(x,\mathbf{y}) + B(x,\mathbf{y},h,f)$$

full linear schema deriving along ℓ captures **FP**

THE GLOBAL PICTURE (SO FAR) : FTC^0

MFCS 2025

$$f(0,\mathbf{y}) = g(\mathbf{y})$$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = A(x,\mathbf{y},h) \times f(x,\mathbf{y}) + B(x,\mathbf{y},h)$$

strict linear schema deriving along ℓ captures **FTC^0**

THE GLOBAL PICTURE (SO FAR) : $\mathbf{FTC}^0$

MFCS 2025

$$f(0, \mathbf{y}) = g(\mathbf{y})$$
$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = A(x, \mathbf{y}, h) \times f(x, \mathbf{y}) + B(x, \mathbf{y}, h)$$

REMARK (BINARY COUNTING). BCOUNT can be expressed by the IVP with initial value $f(0, \mathbf{y}) = g(\mathbf{y})$ and such that :

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = k(x, \mathbf{y})$$

For $g(\mathbf{y}) = \mathrm{bit}(0, y)$ and $k(x, \mathbf{y}) = \mathrm{bit}(\ell(x), y)$, $f(x, x) = \#\{u : \mathrm{bit}(u, x) = 1\}$, which as desired outputs the sum of the bits of a given input.

THE GLOBAL PICTURE (SO FAR) : FTC^0

MFCS 2025

$$f(0,\mathbf{y}) = g(\mathbf{y})$$
$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = A(x,\mathbf{y},h) \times f(x,\mathbf{y}) + B(x,\mathbf{y},h)$$

REMARK (BINARY COUNTING). BCOUNT can be expressed by the IVP with initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ and such that :

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = k(x,\mathbf{y})$$

$A = 0$ and $B = k(x,\mathbf{y}) \leq \{0,1\}$

For $g(\mathbf{y}) = \mathrm{bit}(0,y)$ and $k(x,\mathbf{y}) = \mathrm{bit}(\ell(x),y)$, f … $\}$, which as desired outputs the sum of the bits of a given input.

THE GLOBAL PICTURE (SO FAR) : FTC^0

MFCS 2025

$$f(0, \mathbf{y}) = g(\mathbf{y})$$
$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = A(x, \mathbf{y}, h) \times f(x, \mathbf{y}) + B(x, \mathbf{y}, h)$$

The solution of this system is : $f(x, \mathbf{y}) = \sum_{u=-1}^{\ell(x)-1} k(\alpha(u), \mathbf{y})$, where $\alpha(z) = 2^z - 1$ and $k(\alpha(-1), \cdot) = g(y)$.

REMARK (BINARY COUNTING). BCOUN $y) = g(\mathbf{y})$ and such that :

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = k(x, \mathbf{y})$$

For $g(\mathbf{y}) = \mathrm{bit}(0, y)$ and $k(x, \mathbf{y}) = \mathrm{bit}(\ell(x), y)$, $f(x, x) = \#\{u : \mathrm{bit}(u, x) = 1\}$, which as desired outputs the sum of the bits of a given input.

The global picture (so far) : FACC[n]

ICALP 2026

$k(x, \mathbf{y})$ takes values in $\{0, 1\}$

$$f(0, \mathbf{y}) = g(\mathbf{y})$$

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

strict schema, with special restricted “linearity” captures corresponding **FACC[n]**

THE GLOBAL PICTURE (SO FAR) : FACC[n]

for any $n \notin \{2, 6\}$, these are the very first recursion-theoretic characterizations of these classes

$$f(0, \mathbf{y}) = g(\mathbf{y})$$

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

THE GLOBAL PICTURE (SO FAR) : FACC[2]

ICALP 2026

$g(\mathbf{y}), k(x, \mathbf{y})$ take values in $\{0, 1\}$

$$f(0, \mathbf{y}) = g(\mathbf{y})$$

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = -2 \times k(x, \mathbf{y}) \times f(x, \mathbf{y}) + k(x, \mathbf{y})$$

REMARK (COUNTING MODULO 2). Computation performed by MOD 2 gates can be simulated via ℓ-2ODE by considering $g(x, y) = 0$ and $k = \mathrm{bit}(\ell(x), y)$, i.e., the solution of the IVP s.t. cmod2(0,y)=0 and

$$\frac{\partial \mathrm{cmod2}(x, y)}{\partial \ell} = -2\mathrm{bit}(\ell(x), y) \times \mathrm{cmod2}(x, y) + \mathrm{bit}(\ell(x), y).$$

The count of input bits modulo 2 is obtained by computing cmod2(x, x).

Let us consider the input z of length $\ell(z) = 2$ and whose binary representation is $z_1 z_0$. Then,

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

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = -2 \times k(x, \mathbf{y}) \times f(x, \mathbf{y}) + k(x, \mathbf{y})$$

REMARK (COUNTING MODULO 2). Computation performed by MOD 2 gates can be simulated via ℓ-2ODE by considering $g(x, y) = 0$ and $k = \mathrm{bit}(\ell(x), y)$, i.e., the solution of the IVP s.t. cmod2(0,y)=0 and

$$\frac{\partial \mathrm{cmod2}(x, y)}{\partial \ell} = -2\mathrm{bit}(\ell(x), y) \times \mathrm{cmod2}(x, y) + \mathrm{bit}(\ell(x), y).$$

The count of input bits modulo 2 is obtained by computing cmod2(x, x).

THE GLOBAL PICTURE (SO FAR) : $\mathbf{FAC}^0$

MFCS 2025

$$f(0,\mathbf{y}) = g(\mathbf{y})$$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y}) \quad k(x,\mathbf{y}) \leq 1$$

strict schema with restricted linearity s.t. $A = 1$ and $B \leq 1$) captures $\mathbf{FAC}^0$

THE GLOBAL PICTURE (SO FAR) : FNC[1]

MFCS 2025

$$f(0, \mathbf{y}) = g(\mathbf{y})$$

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = -f(x, \mathbf{y}) + B(x, \mathbf{y}, f, h) \quad B \geq 0$$

non-strict (with simple calls) schema with restricted linearity, s.t. $A = -1$ and $B \geq 0$

THE GLOBAL PICTURE (SO FAR) : FACC[2]

MFCS 2025

$$f(0,\mathbf{y}) = g(\mathbf{y})$$

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = -f(x,\mathbf{y}) + K(x,\mathbf{y},f,h) \quad K \leq 1$$

non-strict schema with restricted linearity s.t. $A = -1$ and $K \leq 1$

The global picture (so far)

	strict schemas along ℓ	non-strict schemas along ℓ	schemas along ℓ_2
FAC^0	$(k(x)-1)\times f(x)$ $f(x)+k(x)$	$(K(x,f)-1)\times f(x)$	$B(x)$
FACC[2]	$-k(x)\times f(x)+k(x)\times k'(x)$ $-2k(x)\times f(x)+k(x)$	$-f(x)+K(x,f)$	
FACC[n]	$-n\times k(x)\times \left\lfloor \frac{f(x)}{n-1} \right\rfloor + k(x)$		
FTC^0	$k(x)$ $A(x)\times f(x)+B(x)$	$K(x,f)$ $K(x,f)\times f(x)$	$B(x,f)$
FNC^1		$f(x)+K(x,f)$ $-f(x)+B(x,f)$	$A(x,f)\times f(x)+B(x,f)$
FAC^1		$B(x,f)$	

SUMMING UP

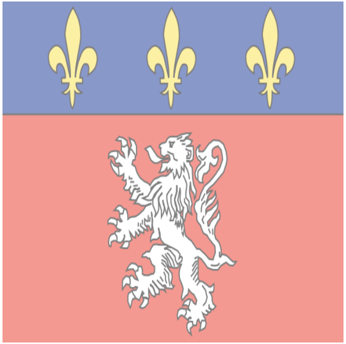
- new framework to study circuit complexity
- simple and uniform characterizations for small circuit classes ranging from $\mathbf{FAC}^0$ to $\mathbf{FNC}^1$
- first characterizations for classes including counting modulo *n* (and corresponding bounded arithmetic)

RESEARCH PERSPECTIVES

RESEARCH PERSPECTIVES

- ODE-based characterizations for $\mathbf{FAC}^i$ and $\mathbf{FNC}^i$ and for complexity hierarchies
- constraints on discrete ODE vs. circuit resources : (further) investigation into the computational power of schemas obtained deriving along ℓ_2
- ODE-based characterizations for circuit over the reals and arithmetic circuits
- ODE-based characterizations for counting classes (and for monotone ones)
- separation results via discrete ODEs
- logical and proof-theoretical counterparts

Thank you for your attention! Questions?



ADDITIONAL MATERIAL

From bounded recursion to ODEs

Definition (Discrete Derivative)

The *discrete derivative* of $\mathbf{f}(x)$ is defined as

$$\mathbf{f}'(x) = \mathbf{f}(x+1) - \mathbf{f}(x).$$

Definition (Discrete ODE)

System of equations of the form,

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

From bounded recursion to ODEs

Definition (Discrete Derivative)

The *discrete derivative* of $\mathbf{f}(x)$ is defined as

$$\mathbf{f}'(x) = \mathbf{f}(x+1) - \mathbf{f}(x).$$

Definition (Discrete ODE)

$$\frac{\partial \mathbf{f}(x,\mathbf{y})}{\partial x} = \mathbf{f}(x+1,\mathbf{y}) - \mathbf{f}(x,\mathbf{y})$$

System of equations of the form,

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

DISCRETE LINEAR ODE

Lemma (Solution of Linear ODE)

For matrices $\mathbf{A}$ and vectors $\mathbf{B}, \mathbf{G}$, the solution of

$$\begin{aligned} \mathbf{f}(0, \mathbf{y}) &= \mathbf{G}(\mathbf{y}) && \text{(initial condition)} \\ \mathbf{f}'(x, \mathbf{y}) &= \mathbf{A}(x, \mathbf{y}) \times \mathbf{f}(x, \mathbf{y}) + \mathbf{B}(x, \mathbf{y}) \end{aligned}$$

is

$$\mathbf{f}(x, \mathbf{y}) = \sum_{u=-1}^{x-1} \left(\prod_{t=u+1}^{x-1} \left(1 + \mathbf{A}(t, \mathbf{y})\right)\right) \times \mathbf{B}(u, \mathbf{y})$$

with conventions $\prod_{x}^{x-1} \kappa(x) = 1$ and $\mathbf{B}(-1, \mathbf{y}) = \mathbf{G}(\mathbf{y})$.

DISCRETE LINEAR ODE

Lemma (Solution of Linear ODE)

For matrices $\mathbf{A}$ and vectors $\mathbf{B}, \mathbf{G}$, the solution of

$$\mathbf{f}(0, \mathbf{y}) = \mathbf{G}(\mathbf{y}) \qquad \text{(initial condition)}$$
$$\mathbf{f}'(x, \mathbf{y}) = \mathbf{A}(x, \mathbf{y}) \times \mathbf{f}(x, \mathbf{y}) + \mathbf{B}(x, \mathbf{y})$$

is

$$\mathbf{f}(x, \mathbf{y}) = \sum_{u=-1}^{x-1} \left(\prod_{t=u+1}^{x-1} \left(1 + \mathbf{A}(t, \mathbf{y})\right)\right) \times \mathbf{B}(u, \mathbf{y})$$

algorithm unrolling the computation of $\mathbf{f}(x+1, \mathbf{y})$ from $\mathbf{f}(x, \mathbf{y}), \mathbf{f}(x-1, \mathbf{y}), \dots, \mathbf{f}(0, \mathbf{y})$

CHARACTERIZING $\mathbf{FTC}^0$

Lemma

If g and h are in $\mathbf{FTC^0}$ *and f is defined by ℓ-pODE from then, then f is in* $\mathbf{FTC^0}$.

Proof Sketch.

By definition of ℓ-ODE,

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

with the convention that $\prod_{x}^{x-1} \kappa(x) = 1$ and $B(-1,\mathbf{y}) = g(\mathbf{y})$. Computing this expression can be done combining iterated addition and multiplication, which are in $\mathbf{FTC^0}$. □

CHARACTERIZING $\mathbf{FTC}^0$

EXAMPLE (BINARY COUNTING AND ITERATED ADDITION) BCOUNT can be expressed by the IVP with initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ and such that :

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = k(x,\mathbf{y})$$

For $g(y) = \mathrm{bit}(0,y)$ and $k(x,y) = \mathrm{bit}(\ell(x),y)$, $f(x,x) = \#\{u : \mathrm{bit}(u,x) = 1\}$, which as desired outputs the sum of the bits of a given input.
Similarly, ITADD is obtained considering an IVP such that :

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = B(x,\mathbf{y},h(x,\mathbf{y})).$$

CHARACTERIZING FTC^0

EXAMPLE (BINARY COUNTING AND ITERATED ADDITION) BCOUNT can be expressed by the IVP with initial value $f(0, \mathbf{y}) = g(\mathbf{y})$ and such that :

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = k(x, \mathbf{y})$$

$A = 0$ and $B = k(x, \mathbf{y}) \in \{0, 1\}$

For $g(y) = \mathrm{bit}(0, y)$ and $k(x, y) =$ … $\{u : \mathrm{bit}(u, x) = 1\}$, which as desired outputs the sum of the bits of a given input.
Similarly, ITADD is obtained considering an IVP such that :

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = B(x, \mathbf{y}, h(x, \mathbf{y})).$$

CHARACTERIZING FTC^0

EXAMPLE (BINARY COUNTING AND ITERATED ADDITION) BCOUNT can be expressed by the IVP with initial value $f(0, \mathbf{y}) = g(\mathbf{y})$ and such that :

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = k(x, \mathbf{y})$$

For $g(y) = \text{bit}(0, y)$ and k … $(u, x) = 1\}$, which as desired outputs the su…

The solution of this system is : $f(x, \mathbf{y}) = \sum_{u=-1}^{\ell(x)-1} k(\alpha(u), \mathbf{y})$, where $\alpha(z) = 2^z - 1$ and $k(\alpha(-1), \cdot) = g(y)$.

Similarly, ITADD is obtained considering an IVP such that :

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = B(x, \mathbf{y}, h(x, \mathbf{y})).$$

CHARACTERIZING FTC^0

EXAMPLE (BINARY COUNTING AND ITERATED ADDITION) BCOUNT can be expressed by the IVP with initial value $f(0, \mathbf{y}) = g(\mathbf{y})$ and such that :

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = k(x, \mathbf{y})$$

For $g(y) = \text{bit}(0, y)$ and $k(x, y) = \text{bit}(\ell(x), y)$, $f(x, x) = \#\{u : \text{bit}(u, x) = 1\}$, which as desired outputs the sum of the bits of a given input.

Similarly, ITADD is obtained con

We consider BCOUNT$(x) = f(x, x)$, where f is defined by ℓ-pODE so that $f(x, y) = \sum_{u=0}^{\ell(x)} \text{bit}(u, y)$.

$$\frac{\partial f(x, \mathbf{y})}{\partial \ell} = B(x, \mathbf{y}, h(x, \mathbf{y})).$$

CHARACTERIZING $\mathbf{FTC}^0$

Theorem (MFCS 2025)

$$\mathbf{FTC}^0 = [\mathbf{0}, \mathbf{1}, \mathrm{sg}, \ell, \mathrm{bit}, +, -, \div, \pi_i^p, \#; \circ, \ell\text{-pODE}]$$

CHARACTERIZING $\mathbf{FTC}^0$

Theorem (MFCS 2025)

$$\mathbf{FTC}^0 = [\mathbf{0}, \mathbf{1}, \mathrm{sg}, \ell, \mathrm{bit}, +, -, \div, \pi_i^p, \#; \circ, \ell\text{-pODE}]$$

- $\supseteq$ Closure of ℓ-pODE.
- $\subseteq$ Relying on the previous Example and on the fact that ℓ-pODE encapsulates ℓ-ODE$_1$ computation.

On Clote's $\mathcal{N}_0'$

Definition (k Bounded Recursion on Notation, k-BRN)

For $h_i(x, \mathbf{y}) \leq k$ and $i \in \{0, 1\}$,

$$f(0, \mathbf{y}) = g(\mathbf{y})$$
$$f(\mathbf{s_i}(x), \mathbf{y}) = h_i(x, \mathbf{y}, f(x, \mathbf{y})).$$

Definition (Function Algebra $\mathcal{N}_0'$)

$$\mathcal{N}_0' = [\mathbf{0}, \pi_i^p, \mathbf{s}_0, \mathbf{s}_1, \mathbf{BIT}, \ell, \#; \circ, \mathrm{CRN}, 4\text{-BRN}]$$

On Clote's $\mathcal{N}_0'$

Definition (k Bounded Recursion on Notation, k-BRN)

For $h_i(x, \mathbf{y}) \leq k$ and $i \in \{0, 1\}$,

$$f(0, \mathbf{y}) = g(\mathbf{y})$$
$$f(\mathbf{s_i}(x), \mathbf{y}) = h_i(x, \mathbf{y}, f(x, \mathbf{y})).$$

Theorem (Clote, 1993)

$$\mathbf{FNC^1} = \mathcal{N}_0'$$

CHARACTERIZING $\mathbf{FNC}^1$

CHARACTERIZING FNC[1]

Definition (Schema ℓ-bODE)

The function f is defined by ℓ-bODE from g and h if it is the solution of the IVP :

$$
\begin{aligned}
f(0, \mathbf{y}) &= g(\mathbf{y}) \\
\frac{\partial f(x, \mathbf{y})}{\partial \ell} &= -f(x, \mathbf{y}) + B(x, \mathbf{y}, h(x, \mathbf{y}), f(x, \mathbf{y}))
\end{aligned}
$$

where $B \geq 0$ is a sg-polynomial expression and f only occurs in it in expressions of the form $\mathrm{sg}(f(x, \mathbf{y}) - c)$, for any constant c.

CHARACTERIZING FNC[1]

Definition (Schema ℓ-b$_0$ODE)

The function f is defined by ℓ-b$_0$ODE from g and h if it is the solution of the IVP :

$$
\begin{aligned}
f(0, \mathbf{y}) &= g(\mathbf{y}) \\
\frac{\partial f(x, \mathbf{y})}{\partial \ell} &= -f(x, \mathbf{y}) + K(x, \mathbf{y}, h(x, \mathbf{y}), f(x, \mathbf{y}))
\end{aligned}
$$

CHARACTERIZING FNC[1]

Definition (Schema ℓ-b$_0$ODE)

The function f is defined by ℓ-b$_0$ODE from g and h if it is the solution of the IVP :

$$f(0,\mathbf{y}) = g(\mathbf{y})$$
$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = -f(x,\mathbf{y}) + K(x,\mathbf{y},h(x,\mathbf{y}),f(x,\mathbf{y}))$$

$A = -1$

CHARACTERIZING FNC[1]

Definition (Schema ℓ-bODE)

The function f is defined by ℓ-bODE from g and h if it is the solution of the IVP :

$$
\begin{aligned}
f(0,\mathbf{y}) &= g(\mathbf{y}) \\
\frac{\partial f(x,\mathbf{y})}{\partial \ell} &= -f(x,\mathbf{y}) + B(x,\mathbf{y},h(x,\mathbf{y}),f(x,\mathbf{y}))
\end{aligned}
$$

where $B \geq 0$ is a sg-polynomial expression and f only occurs in B in expressions of the form $\mathrm{sg}(f(x,\mathbf{y}) - c)$, for any constant c.

CHARACTERIZING $\mathbf{FNC}^1$

$$[\mathbf{0}, \mathbf{1}, \ell, \mathrm{sg}, \mathbf{BIT}, +, -, \div 2, \#, \pi_i^p; \circ, \ell\text{-pODE}, \ell\text{-bODE}]$$

CHARACTERIZING $\mathbf{FNC}^1$

Theorem (MFCS 2025)

$$\mathbf{FNC}^1 = [0, 1, \ell, \mathrm{sg}, \mathrm{bit}, +, -, \div 2, \#, \pi_i^p; \circ, \ell\text{-pODE}, \ell\text{-bODE}]$$

CHARACTERIZING FNC^1

Theorem (MFCS 2025)

$$\mathbf{FNC}^1 = [0, 1, \ell, \mathrm{sg}, \mathrm{bit}, +, -, \div 2, \#, \pi^p_i; \circ, \ell\text{-pODE}, \ell\text{-bODE}]$$

- $\supseteq$ Closure property of ℓ-pODE and ℓ-bODE.
- $\subseteq$ Indirect proof passing through Clote's $\mathcal{N}'_0$ and rewriting the defining 4-BRN schema via ℓ-bODE.

Characterizing FNC^1

Theorem (MFCS 2025)

$$\mathbf{FNC^1} = [0, 1, \ell, \mathrm{sg}, \mathrm{bit}, +, -, \div 2, \#, \pi_i^p; \circ, \ell\text{-pODE}, \ell\text{-bODE}]$$

- $\supseteq$ Closure property of ℓ-pODE and ℓ-bODE.
- $\subseteq$ Indirect proof passing through Clote's $\mathcal{N}_0'$ and rewriting the defining 4-BRN schema via ℓ-bODE.

$[0, \pi_i^p, s_0, s_1, \ell, \mathrm{bit}, \#; \circ, \mathrm{CRN}, \text{4-BRN}]$

CHARACTERIZING FNC^1

Theorem (MFCS 2025)

$$\mathbf{FNC^1} = [0, 1, \ell, \mathrm{sg}, \mathrm{bit}, +, -, \div 2, \#, \pi_i^p; \circ, \ell\text{-pODE}, \ell\text{-bODE}]$$

- $\supseteq$ Closure property of ℓ-pODE and ℓ-bODE.
- $\subseteq$ Indirect proof passing through Clote's $\mathcal{N}_0'$ and rewriting the defining 4-BRN schema via ℓ-bODE.

$[0, \pi_i^p, s_0, s_1, \ell, \mathrm{bit}, \#; \circ, \mathrm{CRN}, \text{4-BRN}]$

CHARACTERIZING FNC[1]

Remark (4-BRN via ℓ-bODE)

Given f defined by 4-BRN from g and h_i's, we consider the solution of the IVP below

$$
\begin{aligned}
\mathbf{4brn}(x, \mathbf{y}) &= g(\mathbf{y}) \\
\frac{\partial \mathbf{4brn}(x, \mathbf{y})}{\partial \ell} &= -\mathbf{4brn}(x, \mathbf{y}) + h_{z(x)}(\ell(x), \mathbf{y}, 0) \times \mathrm{cosg}(\mathbf{4brn}(x, \mathbf{y})) \\
&\quad + h_{z(x)}(\ell(x), \mathbf{y}, 1) \times \mathrm{sg}(\mathbf{4brn}(x, \mathbf{y})) \times \mathrm{cosg}(\mathbf{4brn}(x, \mathbf{y}) - 1) \\
&\quad + \dots \\
&\quad + h_{z(x)}(\ell(x), \mathbf{y}, 4) \times \mathrm{sg}(\mathbf{4brn}(x, \mathbf{y}) - 3) \times \mathrm{cosg}(\mathbf{4brn}(x, \mathbf{y}) - 4)
\end{aligned}
$$

with $z(x) = \mathbf{BIT}(\ell(y) - \ell(x) - 1, y)$. This is an instance of ℓ-bODE. Then

$$\mathbf{4brn}(x, x, \mathbf{z}).$$

CHARACTERIZING FNC^1

Remark (4-BRN via ℓ-bODE)

Given f defined by 4-BRN from g and h_i's, we consider the solution of the IVP below

$\mathbf{4brn}(x, \mathbf{y})$ only occurs B only in expressions of the form $\mathrm{sg}(\mathbf{4brn}(x, \mathbf{y}) - c)$

$$
\begin{aligned}
\mathbf{4brn}(x, \mathbf{y}) &= g(\mathbf{y}) \\
\frac{\partial \mathbf{4brn}(x, \mathbf{y})}{\partial \ell} &= -\mathbf{4brn}(x, \mathbf{y}) + h_{z(x)}(\ell(x), \mathbf{y}, 0) \times \mathrm{cosg}(\mathbf{4brn}(x, \mathbf{y})) \\
&\quad + h_{z(x)}(\ell(x), \mathbf{y}, 1) \times \mathrm{sg}(\mathbf{4brn}(x, \mathbf{y}) \times \mathrm{cosg}(\mathbf{4brn}(x, \mathbf{y}) - 1) \\
&\quad + \dots \\
&\quad + h_{z(x)}(\ell(x), \mathbf{y}, 4) \times \mathrm{sg}(\mathbf{4brn}(x, \mathbf{y}) - 3) \times \mathrm{cosg}(\mathbf{4brn}(x, \mathbf{y}) - 4)
\end{aligned}
$$

with $z(x) = \mathbf{BIT}(\ell(y) - \ell(x) - 1, y)$. This is an instance of ℓ-bODE. Then

$$\mathbf{4brn}(x, x, \mathbf{z}).$$

CHARACTERIZING FACC[2]

Theorem (MFCS 2025)

$$\mathbf{FACC[2]} = [0, 1, \mathrm{sg}, \ell, +, -, \div 2, \pi_i^p, \#; \circ, \ell\text{-ODE}_1, \ell\text{-ODE}_3, \ell\text{-b}_0\text{ODE}]$$

CHARACTERIZING FACC[2]

To count the number of 1's in z we consider the solution of the following system

$$\begin{aligned}\mathbf{par}(0,y) &= \mathrm{bit}(0,y)\\ \frac{\partial \mathbf{par}(x,y)}{\partial \ell} &= -\mathbf{par}(x,y) + (\mathrm{sg}(\mathbf{par}(x,y)\times \mathrm{cosg}(\mathrm{bit}(\ell(x)+1,y))\\ &\qquad + \mathrm{cosg}(\mathbf{par}(x,y))\times \mathrm{sg}(\mathrm{bit}(\ell(x)+1,y)))\end{aligned}$$

which is clearly an instance of ℓ-b$_0$ODE and compute $\mathbf{par}(z,z)$.

The

, ℓ-b$_0$ODE]

- $\supseteq$ Basic functions are computable [...] FACC[2] and closure under composition and ℓ-ODE$_1$, ℓ-ODE$_3$ and ℓ-b$_0$ODE.
- $\subseteq$ Follows from the fact that the parity function can be rewritten via ℓ-b$_0$ODE.