# Logspace Computability and Regressive Machines

S. Mazzanti

Università Iuav di Venezia

ICTCS 2014 - Perugia 17-19/9/2014

# Outline

# Introduction

- The set **L** of languages decidable in logarithmic space is the set of languages recognized by:
  - read-only while programs [Jones, 1999];
  - the functions belonging to the closure with respect to substitution and simultaneous recursion on notation of the constant functions and the projection functions [Kristiansen, 2005].

# Results

This work:

- introduces the class **E** of number theoretic functions generated by the constant functions, the projection functions, the predecessor function, the substitution operator, and the recursion on notation operator;
- introduces *regressive machines*, i.e. register machines which have the division by 2 and the predecessor as basic operations;
- shows that **E** is the class of functions computable by regressive machines and that the sharply bounded functions of **E** coincide with the sharply bounded logspace computable functions.

# Advantages

- Function algebra **E** is defined without making use of ramified (safe) or bounded recursion schemes.
- Even if the present work is concerned with number theoretic functions, it can be considered an improvement of the characterization of **L** given in [Kristiansen, 2005] because:
  - recursion on notation is used instead of simultaneous recursion.
  - not only the $0-1$ valued logspace computable functions, but also the sharply bounded logspace computable functions are characterized.[1]
- Regressive machines are a simple computation model for **L**.

[1] Sharply bounded logspace functions were characterized using safe recursion in [Bellantoni thesis].

# Definitions

- Let $f, g, h$ be functions of finite arity on the set $\mathbb{N} = \{0, 1, \ldots\}$ of natural numbers.
  - $f$ is a *polynomial growth function* iff there is a polynomial $p$ such that $|f(\mathbf{x})| \leq p(|\mathbf{x}|)$ for any $\mathbf{x}$.[2]
  - $f$ is *sharply bounded* iff there is a polynomial $p$ such that $f(\mathbf{x}) \leq p(|\mathbf{x}|)$ for any $\mathbf{x}$,
  - $f$ is *regressive* iff there is some constant $k$ such that $f(\mathbf{x}) \leq \max(\mathbf{x}, k)$ for any $\mathbf{x}$.
- For any $f$, we set $bit_f(\mathbf{x}, i) = bit(f(\mathbf{x}), i)$ and $len_f(\mathbf{x}) = |f(\mathbf{x})|$.
- The characteristic function $ch_P$ of a predicate $P$ returns 1 if $P(\mathbf{x})$ is true, 0 otherwise.

[2] $|x_1, \ldots, x_n| = |x_1|, \ldots, |x_n|$ and $|x| = \lceil \log_2(x+1) \rceil$ is the number of bits of the binary representation of $x$.

# Some basic functions

- the *constant* functions $C_n : x \longmapsto n$ ;
- the *binary successor* functions $s_i : x \longmapsto 2x + i \quad (i \in \{0, 1\})$
- the *bit* function $bit : x, y \longmapsto rem(\lfloor x/2^y \rfloor, 2)$;
- the *length* function $len : x \longmapsto |x| = \lceil \log_2(x + 1) \rceil$ ;
- the *conditional* function $cond : 0, y, z \mapsto y;\ \ x + 1, y, z \mapsto z$ ;
- the *smash* function $smash : x, y \longmapsto x\#y = 2^{|x| \cdot |y|}$ ;
- the *most significant part* function $MSP : x, y \longmapsto \lfloor x/2^y \rfloor$;
- the *log most significant part* function $msp : x, y \longmapsto \left\lfloor x/2^{|y|} \right\rfloor$;
- the *predecessor* function $P : x + 1 \longmapsto x;\ \ 0 \longmapsto 0$;
- the *projection* functions $I^a[i] : x_1, \ldots, x_a \longmapsto x_i$.

# Recursion on notation

The *recursion on notation* operator $RN(g, h_0, h_1)$ transforms function $g : \mathbb{N}^a \to \mathbb{N}$ and functions $h_0, h_1 : \mathbb{N}^{a+1} \to \mathbb{N}$ into the function $f : \mathbb{N}^{a+1} \to \mathbb{N}$ such that

$$\begin{aligned} f(0, \mathbf{y}) &= g(\mathbf{y})\,, \\ f(s_i(x), \mathbf{y}) &= h_i(x, \mathbf{y}, f(x, \mathbf{y})) \end{aligned}$$

where $i \in \{0, 1\}$ and $x > 0$ when $i = 0$ .

# Function algebras and class $\mathbf{E}(\mathbf{F})$

Let

$$\mathrm{clos}(f_1, \ldots, f_n, \mathbf{F_1}, \ldots, \mathbf{F_m}; op_1, \ldots, op_b)$$

be the inductive closure of $\{f_1, \ldots, f_n\} \cup \mathbf{F_1} \cup \ldots \cup \mathbf{F_m} \cup \mathbf{I}$ with respect to operators $op_1, \ldots, op_b$ where $\mathbf{I}$ is the set of the projection functions $I^a[i] : x_1, \ldots, x_a \mapsto x_i$ with $1 \leq i \leq a$. We define

$$\mathbf{E}(\mathbf{F}) = \mathrm{clos}(P, \{C_n\}_n, \mathbf{F}; SUBST, RN)$$

and set

$$\mathbf{E} = \mathbf{E}(\emptyset), \mathbf{E}(f_1, \ldots, f_a) = \mathbf{E}(\{f_1, \ldots, f_a\}).$$

# Properties of class E

- $div_2, rem_2, cond, msp, MSP, bit, \max, \min \in \mathbf{E}$
- $\mathbf{E}$ contains the (characteristic functions of the) Boolean closure of $x < y, x \leq y, x > y, x \geq y, x = y, x \neq y$
- $\mathbf{E}$ contains sharply bounded versions of the arithmetic operations $add, sub, mult, rem, div$ (e.g. $add_p(\mathbf{x}, y, z) = y + z$ for $y, z \leq p(|\mathbf{x}|)$)
- (*Polynomial iteration*) For any $g \in \mathbf{E}$ and any polynomial $p$ there is a function $f_p \in \mathbf{E}$ such that

$$f_p(x_1, \ldots, x_k, y, z) = f(2^{p(|x_1|,\ldots,|x_k|)} - 1, y, z)$$

where

$$\begin{aligned} f(0, y, z) &= z, \\ f(s_i(x), y, z) &= g(y, f(x, y, z)). \end{aligned}$$

# Properties of class **E** - 2

- **E** is closed w.r.t. the *sharply bounded maximization* operator $MAX(g,h)$ transforming function $g : \mathbb{N}^{a+1} \to \mathbb{N}$ and s.b. function $h : \mathbb{N}^a \to \mathbb{N}$ into the function $f : \mathbb{N}^a \to \mathbb{N}$ such that

$$f(\mathbf{x}) = \max\{i \leq h(\mathbf{x}) | g(\mathbf{x}, i) \neq 0\}$$

if $\{i \leq h(\mathbf{x}) | g(\mathbf{x}, i) \neq 0\} \neq \emptyset$, otherwise $f(\mathbf{x}) = 0$.

(CP 1) $len_g \in \mathbf{E}(bit_g)$ for any polynomial growth function $g$;

(CP 2) $ch_{g_1 < g_2} \in \mathbf{E}(bit_{g_1}, bit_{g_2})$ for any pol. growth functions $g_1, g_2$;

(CP 3) $g \in \mathbf{E}(bit_g)$ for any sharply bounded function $g$.

# Regressive Machines

*Regressive machines* are register machines that operate on a finite number of variables (the registers) $\mathtt{X}_1, \ldots, \mathtt{X}_b$. Programs are built up according to the following grammar:

$$\mathtt{P} ::= \mathtt{X_i} := \mathsf{e} | \mathtt{pred}(\mathtt{X_i}) | \mathtt{half}(\mathtt{X_i}) | \mathtt{P}_1 ; \mathtt{P}_2 | \mathtt{loop}\, \mathtt{X_i}\, \mathsf{do}\, \mathsf{P}\, \mathsf{end}.$$

Expression $\mathsf{e}$ can be any natural number constant, any register, or the least significant bit $\mathtt{lsb}(\mathtt{X_j})$ of $\mathtt{X_j}$.
Instructions $\mathtt{pred}(\mathtt{X_i})$ and $\mathtt{half}(\mathtt{X_i})$ compute the predecessor and (the quotient of) the division by 2 of $\mathtt{X_i}$, respectively.
The program $\mathtt{loop}\, \mathtt{X_i}\, \mathsf{do}\, \mathsf{P}\, \mathsf{end}$ executes $|x|$ times program $\mathtt{P}$, where $x$ is the value of $\mathtt{X_i}$.

# Regressive Machines - 2

- Regressive machines compute regressive functions and operate in polynomial time.
- A program $\mathtt{P}$ with $b$ registers computes a function $f : \mathbb{N}^a \to \mathbb{N}$ w.r.t. inputs $\mathtt{X_1}, \ldots, \mathtt{X_a}$ and output $\mathtt{X_j}$ iff for any $n_1, \ldots, n_a$ the value $f(n_1, \ldots, n_a)$ is returned in register $\mathtt{X_j}$ when $\mathtt{P}$ is executed with $\mathtt{X_i}$ having initial value $n_i$ for $1 \leq i \leq a$ and all the other registers are initialized to zero.

# The Main Theorem ...

- Let **SB** be the set of sharply bounded functions,
- let **RM** be the set of functions computable by regressive machines,
- let **FL** be the set of logspace computable functions.

**Theorem ( Main Theorem )**

$$\mathbf{FL} \cap \mathbf{SB} \subseteq \mathbf{E} \subseteq \mathbf{RM} \subseteq \mathbf{FL} \cap \mathbf{E}.$$

# … and its consequences

**Corollary (E is the class of functions computable by regressive machines, $\mathbf{E} \subseteq \mathbf{FL}$)**

$\mathbf{E} = \mathbf{RM} \subseteq \mathbf{FL}$.

**Corollary (The s.b. functions in E are the s.b. logspace functions)**

$\mathbf{FL} \cap \mathbf{SB} = \mathbf{E} \cap \mathbf{SB}$.

**Corollary (New characterization of L)**

*The characteristic functions of logspace predicates coincide with the $\{0,1\}$-valued functions in* $\mathbf{E}$.

# Clote-Takeuti's characterization of **FL**

Clote and Takeuti ([Clo-Ta 1995]) have shown that

$$\mathbf{FL} = clos(C_0, s_0, s_1, len, bit, smash; SUBST, CRN, SBRN)$$

where:

- $CRN(g, h_0, h_1)$ is the *concatenation recursion on notation* of $g, h_0, h_1$ ($h_0, h_1$ are $0-1$ valued functions), i.e. the function

$$\begin{array}{rcl} f(0, \mathbf{y}) & = & g(\mathbf{y}), \\ f(s_i(x), \mathbf{y}) & = & s_{h_i(x,\mathbf{y})}(f(x, \mathbf{y})); \end{array}$$

- $SBRN(g, h_0, h_1, l)$ is the *sharply bounded recursion on notation* of $g, h_0, h_1, l$, i.e. the function $f$ s.t.

$$f = RN(g, h_0, h_1)$$

provided that $f(x, \mathbf{y}) \leq |l(x, \mathbf{y})|$ .

# Step 1: $\mathbf{FL} \cap \mathbf{SB} \subseteq \mathbf{E}$

## Theorem

*For any $f \in \mathbf{FL}$ and any polynomial growth functions $g_1, \ldots, g_a$,*

$$bit_{f(g_1,\ldots,g_a)} \in \mathbf{E}(bit_{g_1}, \ldots, bit_{g_a}).$$

## Proof.

The proof is carried out by induction on the characterization of **FL** given by Clote and Takeuti.

Induction Basis ($f = bit$)

$$bit_{bit(g_1,g_2)}(\mathbf{x}, i) = \begin{cases} bit_{g_1}(\mathbf{x}, g_2(\mathbf{x})) & \text{if } (i = 0) \wedge (g_2(\mathbf{x}) < |g_1(\mathbf{x})|) \\ 0 & \text{otherwise} \end{cases}$$

Induction Step: *SUBST* (trivial); *CRN*, *SBRN* (difficult) □

# Step 1: $\mathbf{FL} \cap \mathbf{SB} \subseteq \mathbf{E}$ (end)

**Corollary**

*$bit_f \in \mathbf{E}$ for any $f \in \mathbf{FL}$.*

**Proof.**

$bit_f = bit_{f(I^a[1],\dots,I^a[a])}$ where $a$ is the arity of $f$. □

Then, $\mathbf{FL} \cap \mathbf{SB} \subseteq \mathbf{E}$ because for any $g \in \mathbf{FL} \cap \mathbf{SB}$ we have

$$g \in \mathbf{E}(bit_g) = \mathbf{E}$$

by the corollary above and CP 3.

# Step 2: $\mathbf{E} \subseteq \mathbf{RM}$

- We show by induction on $\mathbf{E}$ that for any $f \in \mathbf{E}$ there is a regressive machine computing $f$.
- The induction basis is trivial, as well as the induction step concerning function substitution.
- The case of recursion on notation is shown by using the `LOOP` construct.

skip proof

## Step 2: $\mathbf{E} \subseteq \mathbf{RM}$

We show the case of recursion on notation. By ind. hyp. assume that there are $\mathtt{P}, \mathtt{Q}_0$ and $\mathtt{Q}_1$ s.t.

$$\{\mathtt{V}_1 = y_1, \ldots \mathtt{V}_a = y_a\}\mathtt{P}\{\mathtt{Z} = g(\mathbf{y})\},$$

$$\{\mathtt{U} = x, \mathtt{V}_1 = y_1, \ldots \mathtt{V}_a = y_a, \mathtt{W} = z\}\mathtt{Q}_\mathtt{i}\{\mathtt{Z}_\mathtt{i} = h_i(x, \mathbf{y}, z)\}(i = 0, 1).$$

Let

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

# Step 2: $\mathbf{E} \subseteq \mathbf{RM}$

Then, the program

```
P;
W:=Z;X0,X1,X2:=X;
loop X do
    half(X0); loop X0 do half(X1); end;
    R:=lsb(X1);half(X1);U:=X1;
    if (R=0) then Q0; W:=Z0 else Q1; W:=Z1;
    X1=X2;
end
```

computes $f$ with respect to inputs $\mathtt{X}, \mathtt{V}_1, \ldots, \mathtt{V}_a$ and output W.

# Step 3: $\mathbf{RM} \subseteq \mathbf{FL} \cap \mathbf{E}$ (preliminary discussion)

- In order to show that $\mathbf{RM} \subseteq \mathbf{FL} \cap \mathbf{E}$, we need to simulate the computations of regressive machines by storing the registers' contents with at most $O(\log(\max(|\mathbf{x}|)))$ bits where $\mathbf{x}$ is the sequence of input values (in other words, we can store values bounded by $p(|\mathbf{x}|)$ for some polynomial $p$).
- Since regressive machines compute regressive functions, registers are bounded by $\max(\mathbf{x}, c))$. So, if we encoded a memory state as usual, the encoding would exceed the logarithmic bound on memory space and we could not compute **RM** functions in logarithmic space.
- To overcome the memory space bound, we introduce *counter machines* and show that they simulate regressive machines using only a logarithmic amount of memory space.

# Counter Machines

- A counter machine operates on read-only input registers $\mathtt{Y}_1, \ldots, \mathtt{Y}_\mathtt{a}$ and read/write registers $\mathtt{Z}_1, \ldots, \mathtt{Z}_\mathtt{b}$ called *counters*.
- A counter machine program is defined as follows:

$$\mathtt{P} ::= \mathtt{Z_i} := \mathtt{e} | \mathtt{succ}(\mathtt{Z_i}) | \mathtt{half}(\mathtt{Z_i}) | \mathtt{P_1} ; \mathtt{P_2} \\ | \mathtt{if}\,(\mathtt{e_1} = \mathtt{n})\,\mathtt{then}\,\mathtt{P_1}\,\mathtt{else}\,\mathtt{P_2} | \mathtt{loop}\,\mathtt{E_i}\,\mathtt{do}\,\mathtt{P}\,\mathtt{end}$$

where $\mathtt{e}$ is a constant, a counter or $\mathtt{lsb}(\mathtt{E_j})$ and $\mathtt{e_1}$ is a counter, $\mathtt{bit}(\mathtt{Y_{Z_i}}, \mathtt{Z_j})$ or $\mathtt{lsb}(\mathtt{Z_i})$. $\mathtt{E_i}$ has value

$$e_i^b(\mathbf{y}, \mathbf{z}) = \begin{cases} z_{i+2} & \text{if } z_i = 0, \\ MSP(y_{z_i}, z_{i+1}) \dot{-} z_{i+2} & \text{otherwise} \end{cases} \quad (1 \leq i \leq b-2)$$

where $\mathbf{y} = y_1, \ldots, y_a$ are the input values and $\mathbf{z} = z_1, \ldots, z_b$ are the values of the counters.

# CMs simulate RMs

- Every regressive machine program $\mathtt{P}$ with $b$ registers is *simulated* by a counter machine program $\mathtt{Q}$ with $3b$ counters.
- The value of register $\mathtt{X_i}$ of program $\mathtt{P}$ is represented by counters $\mathtt{Z_{3i-2}}, \mathtt{Z_{3i-1}}, \mathtt{Z_{3i}}$ of $\mathtt{Q}$ so that $e_{3i-2}(\mathbf{y}, \mathbf{z}) = x_i$.
- If $\mathtt{X_i}$ has been set to a constant value, then $z_{3i-2} = 0$ and $z_{3i}$ is the value of $\mathtt{X_i}$.
- Otherwise, an input value has been assigned (or copied) to $\mathtt{X_i}$ and decrement or division instructions have been performed on it. In that case, the value of $\mathtt{X_i}$ is $MSP(y_{z_{3i-2}}, z_{3i-1}) \dot{-} z_{3i}$.

# Simulation - 2

At start, assume that $\mathtt{X_j} = n_j$. Then, we set $\mathtt{Y_j} = n_j$ and $\mathtt{Z_{3j-2}} = j, \mathtt{Z_{3j-1}} = 0, \mathtt{Z_{3j}} = 0$.

| OPERATION | $\mathtt{X_i}$ | $\mathtt{Y_i}$ | $\mathtt{Z_{3i-2}}$ | $\mathtt{Z_{3i-1}}$ | $\mathtt{Z_{3i}}$ |
|---|---|---|---|---|---|
| Init | $n_i$ | $n_i$ | $i$ | 0 | 0 |
| $\mathtt{X_i} := \mathtt{k}$ | $k$ | $n_i$ | 0 | 0 | $k$ |
| $\mathtt{X_i} := \mathtt{X_j}$ | $n_j$ | $n_i$ | $j$ | 0 | 0 |
| $\mathtt{pred(X_i)}$ | $n_j \dot{-} 1$ | $n_i$ | $j$ | 0 | 1 |
| ... | | | | ... | |
| $\mathtt{pred(X_i)}$ | $n_j \dot{-} n$ | $n_i$ | $j$ | 0 | $n$ |
| $\mathtt{half(X_i)}$ | $\left\lfloor \frac{n_j \dot{-} n}{2} \right\rfloor$ | $n_i$ | $j$ | 1 | $\left\lfloor \frac{n}{2} \right\rfloor + r$ |

$div_2(MSP(u,v) \dot{-} w) = MSP(u, v+1) \dot{-} (div_2(w) + ch\{bit(u,v) < rem_2(w)\})$.

# Simulation - formal definition

- Let $m_{\mathtt{P}} : \mathbb{N}^b \to \mathbb{N}^b$ be the function such that $m_{\mathtt{P}}(\mathbf{x})$ is the memory state after the computation of a R.M. program P starting from state $\mathbf{x}$.
- Let $M_{\mathtt{Q}} : \mathbb{N}^{a+b} \to \mathbb{N}^b$ be the function such that $(\mathbf{y}, M_{\mathtt{Q}}(\mathbf{y}, \mathbf{z}))$ is the memory state after the computation of a C.M. program Q starting from the state $(\mathbf{y}, \mathbf{z})$.
- For any $\mathbf{x} \in \mathbb{N}^b$, $\mathbf{y} \in \mathbb{N}^a$ and $\mathbf{z} \in \mathbb{N}^{3b}$, if $I^b[i](\mathbf{x}) = e_{3i-2}(\mathbf{y}, \mathbf{z})$ for any $1 \leq i \leq b$, then $I^b[i](m_{\mathtt{P}}(\mathbf{x})) = e_{3i-2}(\mathbf{y}, M_{\mathtt{Q}}(\mathbf{y}, \mathbf{z}))$ for any $1 \leq i \leq b$.

## Step 3: $\mathbf{RM} \subseteq \mathbf{FL} \cap \mathbf{E}$

- A function $f : \mathbb{N}^a \to \mathbb{N}$ is computed by a R.M. program $\mathtt{P}$ with $b$ registers iff

$$f(\mathbf{x}) = I^b[j](m_{\mathtt{P}}(\mathbf{x}, 0, \dots, 0))$$

where $j$ is the index of the output register of $\mathtt{P}$.

- Then, there is a C.M. program $\mathtt{Q}$ with $3b$ counters such that

$$f(\mathbf{x}) = e_{3j-2}(\mathbf{x}, M_{\mathtt{Q}}(\mathbf{x}, 1, 0, 0, \dots, a, 0, 0, \dots, 0)).$$

and the counters of $\mathtt{Q}$ are less than $p(|\mathbf{x}|)$ for some $p$.

# Step 3: **RM** $\subseteq$ **FL** $\cap$ **E** (end)

- Therefore, we encode the counters with a single number

$$c_p(\mathbf{x}, \mathbf{z}) = z_1 p(|\mathbf{x}|)^{(3b-1)} + \ldots + z_{3b-1} p(|\mathbf{x}|) + z_{3b} < p(|\mathbf{x}|)^{3b}$$

and define functions $\tilde{e}_{p,i}, \tilde{M}_{p,\mathtt{Q}} : \mathbb{N}^{a+1} \to \mathbb{N}$ in **FL** $\cap$ **E** s.t.

$$\tilde{e}_{p,i}(\mathbf{x}, c_p(\mathbf{x}, \mathbf{z})) = e_i(\mathbf{x}, \mathbf{z}),\ \tilde{M}_{p,\mathtt{Q}}(\mathbf{x}, c_p(\mathbf{x}, \mathbf{z})) = c_p(\mathbf{x}, M_{\mathtt{Q}}(\mathbf{x}, \mathbf{z}))$$

and

$$f(\mathbf{x}) = \tilde{e}_{p,3j-2}(\mathbf{x}, \tilde{M}_{p,\mathtt{Q}}(\mathbf{x}, c_p(\mathbf{x}, 1, 0, 0, \ldots, a, 0, 0, \ldots, 0))).$$

- Since $c_p \in$ **FL** $\cap$ **E**, we obtain that $f \in$ **FL** $\cap$ **E**.

# Open Questions

- $\mathbf{FL} \cap \mathbf{SB} \subseteq \mathrm{clos}(\{C_n\}_n; SUBST, RN)$ ?
- does **E** equal the set of regressive logspace computable functions?

A function $f$ is *non-size-increasing* iff there is some constant $k$ such that $|f(\mathbf{x})| \leq \max(|\mathbf{x}|, k)$ for any $\mathbf{x}$.
For $i \in \{0, 1\}$, consider the *bounded successor* functions

$$bs_i : x, y \longmapsto 2x + i \text{ if } |x| < |y|;\ x, y \longmapsto x \text{ otherwise}.$$

- It is easy to see that $\mathbf{E}' = \mathrm{clos}(bs_0, bs_1, \{C_n\}_n; SUBST, RN)$ contains the non-size-increasing logspace computable functions.
- $\mathbf{E}' \subseteq \mathbf{FL}$? (Conjecture: $\mathbf{E}'$ recognize $\mathbf{P} \cap \mathbf{LINSPACE}$).

# References


- S. J. Bellantoni, *Predicative Recursion and Computational Complexity*, Ph.D. Thesis, University of Toronto, 1992.
- P. Clote and G. Takeuti, *First order bounded arithmetic and small boolean circuit complexity classes*, in *Feasible Mathematics II*, Birkhäuser Boston, 1995.
- P. C. Fischer, J. C. Warkentin, *Predecessor Machines*, J. Comput. System Sci. 8 (1974) 190-219.
- N. D. Jones, *LOGSPACE and PTIME characterized as programming languages*, Theoret. Comput. Sci. 228 (1999) 151-174.
- L. Kristiansen, *Neat algebraic characterizations of LOGSPACE and LINSPACE*, Comput. Complexity 14 (2005) 72–88.