# Logarithmic Space and Permutations

(Joint work with Clément Aubert)

Thomas Seiller

Inria — Inventeurs du monde numérique

Journée GeoCal, Lyon, 15 Février 2013

# Introduction

- Linear Logic and Geometry of Interaction (GoI) have lead to a number of work on computational complexity.
- This work develops a new approach for the study of complexity classes proposed by Girard (2012):
  - It uses operator theory, and it is in particular constructed around the construction of the *crossed product algebra*;
  - It comes from Girard's latest GoI construction;
  - We use it to characterize the classes **co-NL** and **L** by sets of operators.

# The Basic Picture

$$\mathfrak{E} = (\bigotimes_{n\in\mathbf{N}} \mathfrak{R}) \rtimes \mathfrak{S}$$

where:

- $\mathfrak{R}$ is the hyperfinite type $\mathrm{II}_1$ factor;
- $\mathfrak{S}$ is the group of finite permutations of $\mathbf{N}$ acting on $\bigotimes_{n\in\mathbf{N}} \mathfrak{R}$:

$$\sigma.(x_0 \otimes x_1 \otimes \cdots \otimes x_n \otimes \dots) = x_{\sigma^{-1}(0)} \otimes x_{\sigma^{-1}(1)} \otimes \cdots \otimes x_{\sigma^{-1}(n)} \otimes \dots$$

# The Basic Picture

$$\mathfrak{E} = (\underbrace{\mathfrak{R}}_{\text{integers}} \otimes \mathfrak{R} \otimes \mathfrak{R} \otimes \ldots) \rtimes \underbrace{\mathfrak{S}}_{\text{machines}}$$

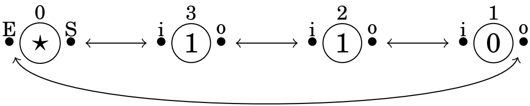
where:

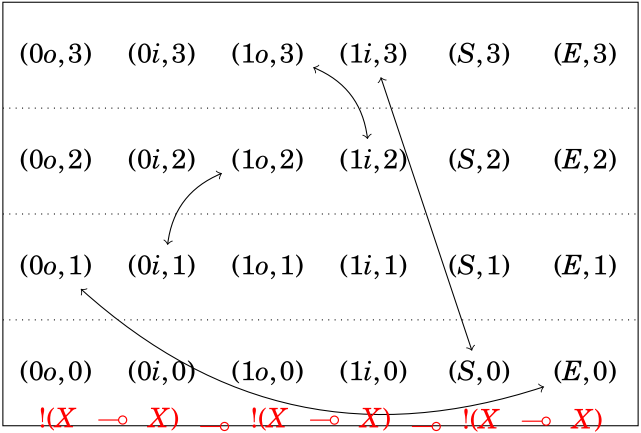
- $\mathfrak{R}$ contains representations of integers;
- $\mathfrak{S}$ generates an algebra $\mathfrak{M}$ in $\mathfrak{E}$ containing the "machines".

# Representation of Integers

- Principle: an integer $n$ is represented as a binary list, i.e. as a proof of
$$\underbrace{!(X \multimap X)}_{0} \multimap \underbrace{!(X \multimap X)}_{1} \multimap !(X \multimap X)$$
- The list can be read from the contraction rules.
- The GoI interpretation of these proofs are the sets of axiom links: we obtain a $6 \times 6$ matrix whose coefficients are $k \times k$ matrices ($k = \log_2(n)$).

# Representation of Integers: Example




$$\underbrace{!(X \multimap X)}_{0} \multimap \underbrace{!(X \multimap X)}_{1} \multimap \underbrace{!(X \multimap X)}_{\star}$$

# Uniform Representation: The Hyperfinite Factor

The last GoI construction takes place in the hyperfinite factor $\mathfrak{R}$ of type $\mathrm{II}_1$. The property we are interested in is that every matrix algebra embeds in the hyperfinite factor.

## Definition

A representation of $n$ is the image of the matrix $M_n$ by a trace-preserving injective $*$-morphism $\mathfrak{M}_{\log_2(n)}(\mathbf{C}) \rightarrow \mathfrak{R}$.

- We now have a uniform representation of integers (all representations live in the same algebra).

# Uniform Representation: The Hyperfinite Factor

**Definition**

A representation of $n$ is the image of the matrix $M_n$ by a trace-preserving injective $*$-morphism $\mathfrak{M}_{\log(n)}(\mathbf{C}) \to \mathfrak{R}$.

**Remark**

*This is in no way unique!*

**Proposition**

*Let $N_n$ and $N'_n$ be two representations of the same integer. Then there exists a unitary $u$ such that $N_n = u^* N'_n u$.*

## Proposition

*Let $N_n$ and $N'_n$ be two representations of the same integer and $\phi \in \mathfrak{M}$. Then:*

$$\phi N_n \textit{ is nilpotent iff } \phi N_{n'} \textit{ is nilpotent}$$

## Definition

For $\phi \in \mathfrak{M}$, one can define:

$$[\phi] = \{n \in \mathbf{N} \mid \phi N_n \text{ is nilpotent}\}$$

- We will now define two sets $P_+$ and $P_{+,1}$ of elements* of $\mathfrak{M}$ and show that $[P_+] = \textbf{co-NL}$ and $[P_{+,1}] = \mathbf{L}$.

# Pointer Machines

- Pointers: move back and forth on the input tape, but never write.
- The input tape is cyclic.

## Definition

A non-deterministic pointer machine with $p \in \mathbf{N}^*$ pointers is a triple $M = \{Q, \rightarrow\}$ where:

- $Q$ is the set of *states*;
- $\rightarrow$ is the *transition relation*.

# Pointer Machines and Logarithmic Space

Pointer machines are equivalent to *Multi-Head Two-Way Finite Automata*. Since the latter characterize **NL** (non-deterministic automata [Holzer, Kutrib, Malcher '08]) and **L** (deterministic automata), we obtain:

Theorem

$$\boldsymbol{DPM} = \boldsymbol{L} \qquad \textit{and} \qquad \boldsymbol{NDPM} = \boldsymbol{co\text{-}NL}$$

# Pointer Machines and Operators

We can encode the pointer machines as operators:

- We encode the basic instructions (move forward/backward and change the state) as partial isometries in $\mathfrak{M}$;
- We define $\rightarrow^*$ as the sum of these atomic transitions.
- We obtain an encoding $M^*$ of $M$ as an operator in $\mathfrak{M}$.

We then obtain:

**Theorem**

*Let $M$ be a non-deterministic pointer machine, $n \in \mathbf{N}$ and $N_n$ a representation of $n$. Then $M$ accepts $n \in \mathbf{N}$ if and only if $M^* N_n$ is nilpotent.*

# Operators and Logarithmic Space: Non-deterministic case

The encoding of pointer machines are *boolean operators*:

**Definition**

A boolean operator is an element of $\mathfrak{M}_{6\times q}(\mathfrak{E})$ such that each coefficient is a finite sum of unitaries induced by $\mathfrak{S}$.

**Proposition**

*If $P_+$ denotes the set of boolean operators.*

$$\textbf{\textit{co-NL}} \subset [P_+]$$

# Operators and Logarithmic Space: deterministic case

**Proposition**

*If $M$ is deterministic pointer machine, then $M^*$ satisfies* $\|M^*\|_1 \leqslant 1$.

**Proposition**

*If $P_{+,1}$ denotes the set of boolean operators $\phi$ such that $\|\phi\|_1 \leqslant 1$.*

$$\boldsymbol{L} \subset [P_{+,1}]$$

# Converse Inclusions

To get the converse inclusion, we prove a technical lemma:

**Lemma**

*Let $\phi$ be a boolean operator and $N_n$ the representation of an integer. Then there exists matrices $\bar{\phi}$ and $M_n$ such that:*

$$\phi N_n \textit{ is nilpotent iff } \bar{\phi} M_n \textit{ is nilpotent}$$

- To check nilpotency of $\phi N_n$, one can check the nilpotency of $\bar{\phi} M_n$ which can be done by a Turing machine using only logarithmic space.
- This Turing machine can be chosen deterministic if we restrict to $\phi \in \mathfrak{M}$ such that $\|\phi\|_1 \leqslant 1$.

# Results

**Theorem**

$$\textbf{\textit{co-NL}} = \textbf{\textit{NDPM}} = [P_{+}]$$

**Theorem**

$$\textbf{\textit{L}} = \textbf{\textit{DPM}} = [P_{+,1}]$$

# Perspectives

- Extend to other classes (other groups, supersets of $P_+$);
- Obtain a real connection with GoI (construct **co-NL** and **L** types);
- Solve the separation problem ?