Logarithmic Space and Permutations
(Joint work with Clément Aubert)

Thomas Seiller

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

\[ \mathcal{E} = \bigotimes_{n \in \mathbb{N}} \mathcal{K} \rtimes S \]

where:

- \( \mathcal{K} \) is the hyperfinite type II\(_1\) factor;
- \( S \) is the group of finite permutations of \( \mathbb{N} \) acting on \( \bigotimes_{n \in \mathbb{N}} \mathcal{K} \):

\[ \sigma.(x_0 \otimes x_1 \otimes \cdots \otimes x_n \otimes \ldots) = x_{\sigma^{-1}(0)} \otimes x_{\sigma^{-1}(1)} \otimes \cdots \otimes x_{\sigma^{-1}(n)} \otimes \ldots \]
The Basic Picture

\[ \mathcal{E} = ( R \otimes R \otimes R \otimes \ldots ) \rtimes S \]

where:

- \( R \) contains representations of integers;
- \( S \) generates an algebra \( \mathcal{M} \) in \( \mathcal{E} \) containing the "machines".
Representation of Integers

- Principle: an integer $n$ is represented as a binary list, i.e. as a proof of

  $$\begin{array}{c}
  \underbrace{(X \leftarrow X)}_{0} \rightarrow \underbrace{(X \leftarrow X)}_{1} \rightarrow \underbrace{!(X \leftarrow X)}_{0}
  \end{array}$$

- 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

\[
\begin{array}{cccccc}
(0o,3) & (0i,3) & (1o,3) & (1i,3) & (S,3) & (E,3) \\
(0o,2) & (0i,2) & (1o,2) & (1i,2) & (S,2) & (E,2) \\
(0o,1) & (0i,1) & (1o,1) & (1i,1) & (S,1) & (E,1) \\
(0o,0) & (0i,0) & (1o,0) & (1i,0) & (S,0) & (E,0) \\
\end{array}
\]
The last GoI construction takes place in the hyperfinite factor $\mathcal{R}$ of type $\text{II}_1$. 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 $\star$-morphism $\mathcal{M}_{\log_2(n)}(\mathbb{C}) \to \mathcal{R}$.

- We now have a uniform representation of integers (all representations live in the same algebra).
Definition
A representation of $n$ is the image of the matrix $M_n$ by a trace-preserving injective $*$-morphism $\mathcal{M}_{\log(n)}(\mathbb{C}) \rightarrow \mathcal{K}$.

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 $N_n = u^* N'_n u$. 
The Algebra of Machines

**Proposition**

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

$$\phi N_n \text{ is nilpotent iff } \phi N'_n \text{ is nilpotent}$$

**Definition**

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

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

- We will now define two sets $P_+$ and $P_{+,1}$ of elements* of $\mathcal{M}$ and show that $[P_+] = \text{co-NL}$ and $[P_{+,1}] = L$. 
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 \( p \in \mathbb{N}^* \) pointers is a triple \( M = \{Q, \rightarrow\} \) where:

- \( Q \) is the set of *states*;
- \( \rightarrow \) is the *transition relation*. 
Pointer machines are equivalent to *Multi-Head Two-Way Finite Automata*. Since the latter characterize $\textbf{NL}$ (non-deterministic automata [Holzer, Kutrib, Malcher ’08]) and $\textbf{L}$ (deterministic automata), we obtain:

**Theorem**

\[ \text{DPM} = \text{L} \quad \text{and} \quad \text{NDPM} = \text{co-NL} \]
We can encode the pointer machines as operators:

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

We then obtain:

**Theorem**

*Let $M$ be a non-deterministic pointer machine, $n \in \mathbb{N}$ and $N_n$ a representation of $n$. Then $M$ accepts $n \in \mathbb{N}$ if and only if $M^*N_n$ is nilpotent.*
The encoding of pointer machines are *boolean operators*:

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

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

$$\textsf{co-NL} \subset [P_+]$$
Proposition

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

Proposition

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

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

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

- 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 $\phi \in \mathcal{M}$ such that $\|\phi\|_1 \leq 1$. 
Results

Theorem

\[ co-NL = NDPM = [P_+] \]

Theorem

\[ L = DPM = [P_{+,1}] \]
Perspectives

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