Track
Structural Proof-Theory and Computational Complexity

Luca Roversi

Dipartimento di Informatica

PROTOCOLLO – Final meeting
15 – 17 September 2004

Outline

Goal

To characterize complexity classes by structural proof-theory:

- **derivations-as-programs**
- **cut elimination-as-evaluation**
- **formulae-as-types**

Methodology

To limit the cut-elimination:

- Derivations structured into levels
- Duplication of specific derivations
- Duplication of derivations at level i implies:
  - New structure at levels $j > i$
  - Interaction only at levels $j > i$

Call it **Strong stratification**

The principles [Gir95]

- sub-recursive classes inside **LL**

ILAL	Elementary	Polynomial
$!(A_1 \otimes \cdots \otimes A_n) \multimap !A$	OK	$!A \multimap !B$
$!A \otimes !B \multimap !(A \otimes B)$	OK	Forbidden
$!A \multimap !A \otimes !A$	OK	OK
$!A \multimap !!A$	Forbidden	Forbidden
$!A \multimap A$	$!A \multimap \S A$	$!A \multimap \S A$
$!A \multimap \mathbb{I}$	OK	OK

- *soundness* and *completeness*: **ELL** w.r.t. elementary time; **LLL, LAL** w.r.t. polynomial time
- Let us call them *Light systems*

Outline

The meaning of characterizations

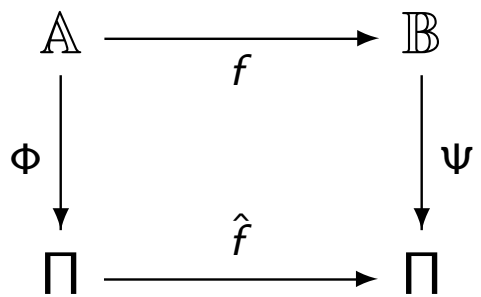
- A Light System **S** characterizes $\mathcal{C}$ if, **fixed** any $f \in \mathcal{C}$:
  - f can be represented by $\hat{f} \in \mathbf{S}$
  - normalizing $\hat{f}(\hat{n})$ costs $O(|\hat{f}(\hat{n})|^{e(d(\hat{f}(\hat{n})))})$:
    - $d(\hat{f}(\hat{n}))$ is the **depth** of $\hat{f}(\hat{n})$
    - e is some function;
  - $|\hat{n}|$ is linear in the value of n
  - $|\hat{f}|$ and $d(\hat{f}(\hat{n}))$ are **constants** w.r.t. $|\hat{n}|$
- **ELL**, **EAL**: e is a tower of exponentials
- **LLL**, **LAL**: e is a constant

Expressive power of fragments

- **ILAL** can also be:
  - **FDtime**$[2^{2^n}]$ sound and complete with **non constant depth** data-types as arguments [MaiMol02]
- **MLL**:
  - has a linear cut-elimination
  - can be **PDtime**$[n^k]$ sound and complete [Mair03,MairTer03]

Uniform representations [DL03]

- Separating, formally, the characterizations inside **ILAL**.
- $\mathbb{A} \xrightarrow{f} \mathbb{B} \in \mathcal{C}$ is **uniformly representable** if:



  - **commutes**
  - Π – set of arguments – has only cut-free derivations
  - $\Phi, \Phi^{-}, \Psi, \Psi^{-}$ are **Space**[ln] transducers

Canonical representations [DL03]

- $\mathbb{A} \xrightarrow{f} \mathbb{B} \in \mathcal{C}$ is **canonically representable** if
  - $\mathbb{A} \xrightarrow{f} \mathbb{B}$ is **uniformly representable**
  - Π – set of cut-free arguments – without $\forall$-left rules:
    - $\forall$-left rules may hide depth/dimension interplay
- **ILAL** is **FDtime**$[n^k]$ canonically sound and complete
- **ILAL** is **FDtime**$[2^{2^k}]$ uniformly, but not canonically, sound and complete.

Uniform representations [BDL04]

$\mathbf{FDtime}[n^k]$	$\supset$	$\mathbf{ILAL}_{\multimap\otimes}$	$\supset \{\text{polynomials on } \mathbb{N}\}$
$\mathbf{FDtime}[n^k]$	$\subset$	$\mathbf{ILAL}_{\multimap\otimes\forall}$	$\forall$ may link depth and dimension
$\mathbf{FDtime}[n^k]$	$=$	$\mathbf{ILAL}_{\multimap\otimes\overline{\mu}}$	linear recursive types don't ...
$\mathbf{FDtime}[n^k]$	$=$	$\mathbf{ILAL}_{\multimap\otimes\oplus\overline{\mu}}$	... as well as additives ...
$\mathbf{FDtime}[n^k]$	$=$	$\mathbf{ILAL}_{\multimap\otimes\overline{\forall}\overline{\mu}}$	... and $\forall$ on linear formulae (1)

- binary numerals in (1): $[01\epsilon] = \lambda 01\epsilon.0(\lambda 01\epsilon.1(\lambda 01\epsilon.\epsilon))$
- support conditional, but not iteration:

$$\begin{aligned} \mathtt{cond}\ \overline{2n}\ E\ O\ N &= E\ \overline{n} \\ \mathtt{cond}\ \overline{2n+1}\ E\ O\ N &= O\ \overline{n} \\ \mathtt{cond}\ \epsilon\ E\ O\ N &= N \end{aligned}$$

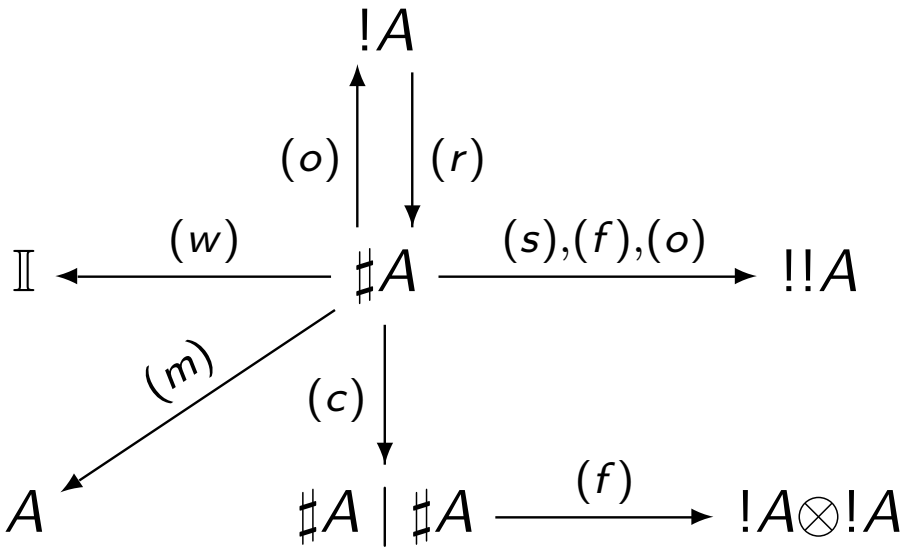
- $\mathtt{cond} = \lambda a.a(\lambda neo\epsilon.e\ n)(\lambda neo\epsilon.o\ n)(\lambda neo\epsilon.\epsilon)$

Outline

Initial table obsolete? [Maz04]

- $\vdash_{\mathbf{xUC}} \Sigma \parallel \Gamma$ with
  - $\Sigma = \alpha_1, \ldots, \alpha_n$
    - structural operations (contraction, ...)
    - α_i built with linear and structural symbols $\sharp, \flat$
  - $\Gamma = A_1, \ldots, A_n$
    - logical operations
    - A_i usual **LL** formula, possibly with §
- $! = \downarrow \sharp$, with $\downarrow$ positive shift
- $? = \uparrow \flat$, with $\uparrow$ negative shift
- $\uparrow$ / $\downarrow$ move formulae between Σ and Γ

Initial table obsolete? [Maz04]



Light system	Principles
ELL	forbid (o) and (r), *i.e.* $!A \neq \sharp A \ldots$
LLL	$\ldots +$ constraint on (f)
SLL	forbid (o) $+$ constraint on (f)

Initial table obsolete?

$$\begin{array}{rcl} \overline{\mathbf{LLL}} & = & \mathbf{ELL}\text{-DJ} \\ & + & \dfrac{\vdash ?\Gamma, \Delta}{\vdash ?\Gamma, \S\Delta}\ (\S) \\ & + & \text{affine !-boxes} \\ & + & \text{conditions on maximal exponential paths [Maz02]} \end{array}$$

$$\begin{array}{rcl} \mathbf{LLL}^* & = & \mathbf{LL} \\ & + & \dfrac{\vdash ?\Gamma, \Delta}{\vdash ?\Gamma, \S\Delta}\ (\S) \\ & + & \text{weights on formulae/rules recovering the meaning of} \\ & & \text{additive blocks in } \mathbf{LLL} \text{ [Maz03]} \end{array}$$

- both sound and complete w.r.t. $\mathbf{FDtime}[n^k]$
- $\mathbf{LLL}^*$ and $\mathbf{LLL}$ are equivalent w.r.t. provability

Outline

Light systems vs. recursive ones

$$\begin{array}{rcl} \textbf{HOLRR} & = & \text{linear affine } \lambda\text{-calculus} \\ & + & \text{higher-order ramified recursion [DLMR04]} \end{array}$$

- Sound and complete w.r.t. $\textbf{FDtime}[n^k]$
- $A ::= \textbf{Words} \mid A \otimes A \mid A \multimap A$
- *ramification*:
  - level on types that models *sorts*
  - $\neq$ stratification of *Light systems*
- *Recursion*: **binder of higher-order variables**
- *Polynomial normalization*:
  - the exponent of the polynomial depends on the nesting levels of the recursive operator
  - rewrites terms under a specific strategy
- *Completeness*: embedding **Ramified recurrence**

Light systems vs. recursive ones [Rov04]

- *Goal*: compositionally simulating recursive schemes
- The simulation can proceed bottom-up inside $\mathtt{T}^{\mathbf{w}}$:
  - $\mathtt{T}^{\mathbf{w}} \supset$ **ILAL**
  - !-boxes of $\mathtt{T}^{\mathbf{w}}$ may depend on arbitrary number of premises:
    - at most one can be !-discharged
    - the others must be §-discharged
- *Main question*: Is $\mathtt{T}^{\mathbf{w}}$ significantly more expressive than **ILAL**?

Outline

A hierarchy of complexity classes on rings [BlumShubSmale-end80]

- *Computational model*:
  - Cells of TM extended to contain "arbitrary" structure
  - Control operations on cells have unitary cost ...
    - multiplication on reals, included!
  - *Computation step*:
    - depend on the current state
    - *does not* depend on the cell being read

FDtime$[n^k]$ on $\mathbb{R}$ inside **ILAL** [BP04]

- *Goals*
  - To extend **ILAL** with δ-rules to simulate operations and relations on rings
  - To give alternative proof of polynomial time soundness and completeness of the model
- *Problem*
  - The effective expressivity of **ILAL** is to low
  - A "while" operator is required

Outline

Outline

Lamping's algorithm

- It implements *optimal reductions à la Lévy* on λ-calculus
- It requires an oracle: maintain the consistence of graphs representing terms
- "No oracle" would mean *great speed-up*:

  most of the reductions may originate from the oracle
- **The λ-terms typeable in EAL do not require oracle!**

Automatic assessment of program complexity

- [CRDR03]
  - Let M be a pure λ-term
  - *Canonical abstract* **EAL**-*terms*: they highlight the structure that can duplicated
  - *Type inference*: it solves linear constraints to get **EAL** formulae as *principal* types
- [CM04]
  - Let M be typed Intuitionistic Logic (natural deduction)
  - *Maximal decoration of* M: symbolic distribution of as much boxes as possible
  - *Solving linear constraints*: they yield an **EAL** formula as type

Outline

- Informal workshop on 15 — 16, October 2003, Bertinoro.
  Participants: Coppola (Udine), Dal Lago (Bologna), Hoffman (Munich), Mairson (Boston - MA), Martini (Bologna), Mazza (Marseille), Ronchi Della Rocca (Torino), Roversi (Torino), Terui (Tokio), Tortora de Falco (Roma).
- Informal workshop on 28 — 29, June 2004, Torino.
  Participants: Coppo (Torino), Coppola (Udine), Dal Lago (Bologna), de'Liguoro (Torino), Dezani (Torino), Falzetta (Torino), Martini (Bologna), Masini (Verona), Mazza (Marseille), Paolini (Torino), Pedicini (Roma), Ronchi Della Rocca (Torino), Roversi (Torino).


P. Baillot and U. Dal Lago.
Fragments of light affine logic vs. polynomial functions.
2004.


P. Baillot and M. Pedicini.
Light affine logic and polynomial computations on rings.
2004.


P. Coppola and S. Martini.
Optimizing optimal reductions.
ACM Transactions on Computational Complexity, To appear, 2004.


P. Coppola and S. Ronchi Della Rocca.
Principal typing for elementary affine logic.
In M. Hofmann, editor, *Typed Lambda Calculi and Applications: 6th International Conference (TLCA 2003)*, volume 2701 of *LNCS*, pages 90–104, Valencia, Spain, 2003. Springer-Verlag.


U. Dal Lago.
On the expressive power of light affine logic.
In C. Blundo and C. Laneve, editors, *Eight Italian Conference on Theoretical Computer Science, Proceedings*, volume 2841 of *Lecture Notes in Computer Science*, pages 216–227. Springer, 2003.


U. Dal Lago, S. Martini, and L. Roversi.
Higer-order linear ramified recurrence.
In *Proceedings of TYPES'04*, volume 3085 of *Lecture Notes in Computer Science*, pages ??–?? Springer Verlag, 2004.


D. Mazza.
Logica lineare e complessità computazionale.
Master's thesis, Università degli Studi Roma Tre, 2002.
Tesi di laurea.


D. Mazza.
Notes on light linear logic.

Technical report, Institut de Mathématiques de Luminy, 2003.

D. Mazza.
Unifying light logics.
Technical report, Institut de Mathématiques de Luminy, 2004.

L. Roversi.
Light languages and primitive recursive functions.
Invited talk at the workshop "Implicit Computational Complexity and Logic" of the Project "ACI Nouvelles interfaces des mathématiques GEOCAL", Paris-Nord (Villetaneuse), 6 – 7, September 2004.