Computing with sequent proof terms: progress report

José Espírito Santo[1]

Centro de Matemática
Universidade do Minho
Portugal
jes@math.uminho.pt

TYPES 2016 2016
23 May 2016
Novi Sad, Serbia

[1] Joint work with Maria João Frade, Luís Pinto

BRIEF RECAPITULATION

System $\lambda\mathbf{Jm}$

- Arguably a well chosen system of sequent proof-terms
- Potentially useful, rich, not fully-understood
- Studied in the 2000's: see bib refs in the last slide
- Report on progress obtained early 2013 (unpublished)

The system (1)

- Expressions:

$$
\begin{array}{rrcl}
\text{(terms)} & t, u, v & ::= & x \mid \lambda x.t \mid \underbrace{t(u, l, (x)v)}_{\textit{gm-application}} \\
\text{(lists)} & l & ::= & [] \mid u :: l
\end{array}
$$

- Multiarity: l not necessarily $[]$
- Generality: v not necessarily x
- Subsystems

$$
\begin{array}{lcllr}
t(u, (x)v) & := & t(u, [], (x)v) & \text{g-application} & \text{subsystem } \lambda\mathbf{J} \\
t(u, l) & := & t(u, l, (x)x) & \text{m-application} & \text{subsystem } \lambda\mathbf{m} \\
t(u) & := & t(u, [], (x)x) & \text{application} & \text{subsystem } \lambda
\end{array}
$$

The system (2)

- Reduction rules:

$$
\begin{array}{rrcl}
(\beta_1) & (\lambda x.t)(u, [], (y)v) & \to_{\beta_1} & \mathbf{s}(\mathbf{s}(u, x, t), y, v) \\
(\beta_2) & (\lambda x.t)(u, v :: l, (y)v) & \to_{\beta_2} & \mathbf{s}(u, x, t)(v, l, (y)v) \\
(\pi) & t(u, l, (x)v)(u', l', (y)v') & \to_{\pi} & t(u, l, (x)v(u', l', (y)v')) \\
(\mu) & t(u, l, (x)x(u', l', (y)v')) & \to_{\mu} & t(u, \mathbf{a}(l, u' :: l'), (y)v') \\
 & & & \text{if } x \notin u', l', v'
\end{array}
$$

  where **s** denotes substitution, **a** denotes append
- 1st reduction process (cut-elimination) $=$ $\beta\pi$-reduction
- 2nd reduction process $=$ μ-reduction
- 3rd reduction process (permutative conversions) $= \cdots$

Aspects of the study of λ**Jm**

- Meta theory
- Normal-forms for sequent proof-terms
- How to define the 3rd reduction process (perm. conversion)
- Subsystems of the cut-elim process, mediated by the other reduction processes
- Computational interpretation of the (sub)systems and reduction processes

Third reduction process (permutative conversion)

- $t(u, l, (x)v)$: instruction to substitute $t(u, l)$ for x in v
  - When?
  - How?
- Versions

Version	Year	When	How
p	2003	$v \neq x$	ordinary subst, stepwise
s	2006	$v \neq x$	ordinary subst, in one go
γ	2006	v not x-normal	ordinary subst, in one go
p	2011	$v \neq x$	ordinary subst, mixed

BRIEF PROGRESS REPORT

The natural subsystem (1)

- A term is *natural* if every gm-application $t(u, l, (x)v)$ in it satisfies: x is main and linear in v.
- x is main and linear in v if:
  - $v = x$, or
  - $v = x(u', l', (y)v')$ and $x \notin u', l', v'$
- A *normal* term is a natural and cut-free term
- Natural terms are closed for:
  - $\beta\pi$-reduction
  - μ-reduction
- Cut-elimination in the natural subsystem should be called *normalization*

The natural subsystem (2)

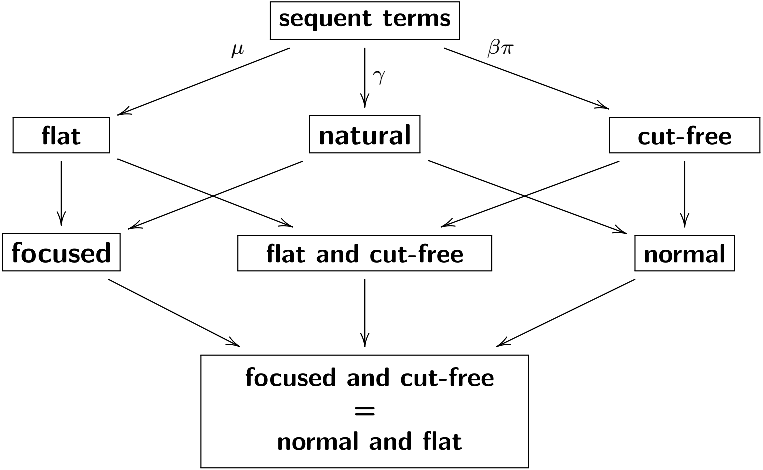
- λ-calculus with
  - application $t(u, l, L)$ where
  - l: list of args
  - hence u, l: non-empty list of args
  - L: list of non-empty lists of args
  - hence (u, l, L): non-empty list of non-empty lists of args (=: multi-list)
- Clear computational interpretation: multi-multiary λ-calculus
  - β: function call with first arg. of the first list of args.
  - π: append of multi-lists
  - μ: flattening of multi-lists
- Generality reduced to a second vectorization mechanism

Third reduction process (permutative conversion)

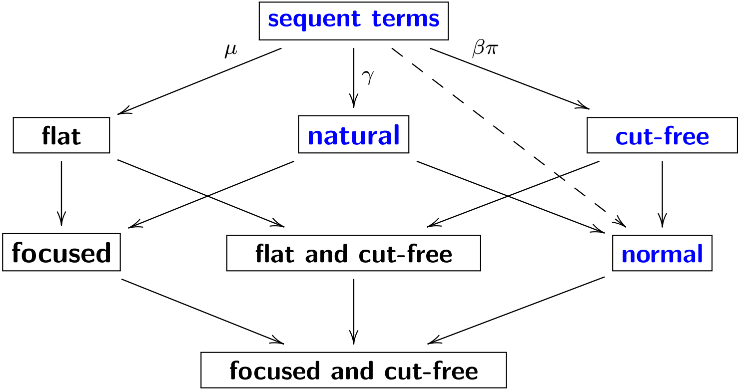
Version	Year	When	How
p	2003	$v \neq x$	ordinary subst, stepwise
s	2006	$v \neq x$	ordinary subst, in one go
γ	2006	v not x-normal	ordinary subst, in one go
p	2011	$v \neq x$	ordinary subst, mixed
γ	2013	(*)	special subst, in one go

(*) x not main-and-linear in v

Taxonomy

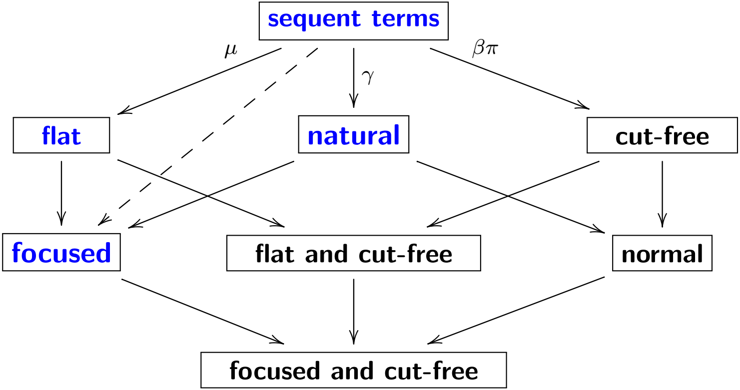

sequent terms
μ
γ
$\beta\pi$
flat
natural
cut-free
focused
flat and cut-free
normal
focused and cut-free
=
normal and flat

Normalization



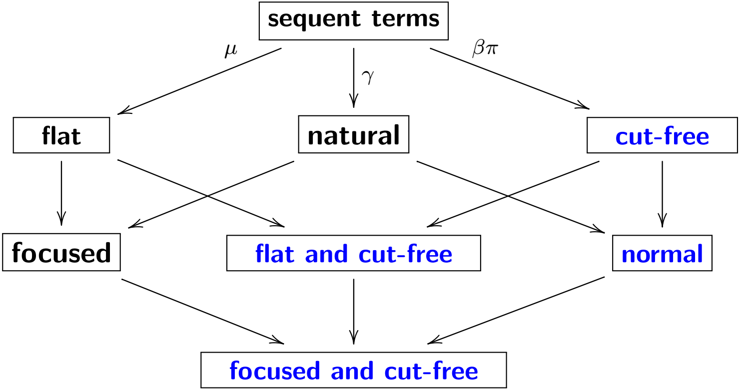
- Commutative square
- Normalization extended to all sequent terms

Ceci n'est pas un cube (1)



- Square does not commute
- Focalization $= \mu \circ \gamma$

Ceci n'est pas un cube (2)



- Each proof determines 8 cut-free forms (rather than 4)

Final remarks:progress report

- Computational interpretation of the (sub)systems and reduction processes
  - Natural system as multi-multiary λ-calculus, where
  - generality is a 2nd vectorization mechanism
- How to define the 3rd reduction process (perm. conversion)
  - New definition of γ
- Meta theory
  - Commutation and preservation between reduction processes
  - Definition of normalization and focalization
- Normal-forms for sequent proof-terms
  - Each proof determines 8 cut-free forms

Bibliographic references

- J. Espírito Santo and L. Pinto, *Permutative conversions in intuitionistic multiary sequent calculus with cuts*, *TLCA'03*, LNCS 2701, 286–300, 2003.
- J. Espírito Santo and L. Pinto, *Confluence and strong normalisation of the generalised multiary λ-calculus*, *TYPES 2003*, LNCS 3085, 194–209, 2004.
- J. Espírito Santo and M.J. Frade and L. Pinto, *Structural proof theory as rewriting* , *RTA'06*, LNCS 4098, 197–211, 2006.
- J. Espírito Santo and L. Pinto, *A calculus of multiary sequent terms*, *ACM Transactions on Computational Logic*, 12:3, art. 22, 2011.