Resource control in sequent lambda calculus

Silvia Ghilezan
University of Novi Sad, Serbia

joint work with
Jelena Ivetić, University of Novi Sad, Serbia
Pierre Lescanne, Ecole Normale Superieure, Lyon, France
Dragiša Žunić, University of Novi Sad, Serbia

TYPES 2010
Warsaw, October 2010

Outline of the talk

1. Logic vs Computation
2. From λ towards sequent λ calculi
3. Sequent λ calculus with resource control

Outline

1. Logic vs Computation
2. From λ towards sequent λ calculi
3. Sequent λ calculus with resource control

Computational interpretations of intuitionistic logic

- 1950s Curry
- 1968 (1980) Howard formulae-as-types
- 1970s Lambek - CCC Cartesian Closed Categories
- 1970s de Bruijn AUTOMATH

logic vs term calculus

$$\vdash A \quad \Leftrightarrow \vdash t : A$$

Curry - Howard - Lambek - de Bruijn correspondence

formulae – as – types
proofs – as – terms
proofs – as – programs

Computational interpretations of intuitionistic logic

Logic vs *term calculi*

- Axiomatic (Hilbert) system (axioms/Modus Ponens)
  Combinatory Logic (combinators/application)
  1930s Schönfinkel, Curry
- Natural Deduction (introduction/elimination)
  λ *calculus(abstraction/ application)*
  1940s Church
- Sequent Calculus (right/left introduction/cut)
  various attempts λ *calculus*
  (abstraction/application/substitution)
  1970s

Outline

Sequent calculus intuitionistic logic

- Pottinger, Zucker 1970s comparing cut-elimination to proof normalization
- Gallier [1991]
- Mints [1996]
- Barendregt, Ghilezan [2000]: λLJ-calculus

But in these, terms do not encode derivations.

- Herbelin [1995]: $\bar{\lambda}$-calculus - developed the idea of making terms explicitly represent sequent calculus derivations.
- Dyckoff and Pinto [1998]
- Computation over terms reflects cut-elimination
- Espírito Santo [2006]: λ^{Gtz}-calculus

λLJ-calculus

- type system LJ - sequent types structure
- term calculus λ-calculus

$$\frac{}{\Gamma \quad A \vdash \quad A}\,(axiom) \qquad \frac{\Gamma \vdash \quad A \quad \Gamma, \quad B \vdash \quad C}{\Gamma, \quad A \to B \vdash \qquad C}\,(\to_{left})$$

$$\frac{\Gamma, \quad A \vdash \quad B}{\Gamma \vdash \qquad : A \to B}\,(\to_{right}) \qquad \frac{\Gamma \vdash \quad A \quad \Gamma, \quad A \vdash \quad B}{\Gamma \vdash \qquad : B}\,(cut)$$

λLJ-calculus

- type system LJ - sequent types structure
- term calculus λ-calculus - natural deduction term structure

$$\frac{}{\Gamma x : A \vdash x : A}\ (axiom) \qquad \frac{\Gamma \vdash t : A \quad \Gamma, x : B \vdash s : C}{\Gamma, y : A \to B \vdash s[x := yt] : C}\ (\to_{left})$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash (\lambda x.t) : A \to B}\ (\to_{right}) \qquad \frac{\Gamma \vdash t : A \quad \Gamma, x : A \vdash s : B}{\Gamma \vdash s[x := t] : B}\ (cut)$$

From λLJ to $\bar{\lambda}$-calculus

λLJ-calculus, Barendregt et al. [2010]

- Using a subsystem λLJ^{cf} Gentzen's Hauptsatz (cut-elimination) theorem is easily proved!
- But, the Curry-Howard correspondence fails...

$$\begin{array}{ccccc} u & \mapsto & yz & \mapsto & \lambda x.yz \\ u & \mapsto & \lambda x.u & \mapsto & \lambda x.yz. \end{array}$$

From λLJ to $\bar{\lambda}$-calculus

λLJ-calculus, Barendregt et al. [2010]

- Using a subsystem λLJ^{cf} Gentzen's Hauptsatz (cut-elimination) theorem is easily proved!
- But, the Curry-Howard correspondence fails...

$$\begin{array}{ccccc} u & \mapsto & yz & \mapsto & \lambda x.yz \\ u & \mapsto & \lambda x.u & \mapsto & \lambda x.yz. \end{array}$$

$\bar{\lambda}$-calculus of Herbelin

- introduction of explicit substitution

$$\lambda x.(u\langle u = yz\rangle) \qquad (\lambda x.u)\langle u = yz\rangle$$

- restriction of the sequent logic LJ - LJT:
  $(\Gamma;\ \vdash A)$ i $(\Gamma; B \vdash A)$
- introduction of a new constructor - list of arguments

λ^{Gtz}-calculus

Syntax

$$\begin{array}{llcl} \text{(terms)} & t, u, v & ::= & x \mid \lambda x.t \mid tk \\ \text{(contexts)} & k & ::= & \widehat{x}.t \mid u :: k \end{array}$$

Reductions

$$\begin{array}{lrcl} (\beta) & (\lambda x.t)(u :: k) & \to & u\widehat{x}.(tk) \\ (\pi) & (tk)k' & \to & t(k@k') \\ (\sigma) & t\widehat{x}.v & \to & v\langle x := t\rangle \\ (\mu) & \widehat{x}.xk & \to & k, \ \text{ako } x \notin k \end{array}$$

- $v\langle x := t\rangle$ is a meta-substitution;
- $k@k'$ is an append

λ^{Gtz} - simple types

Types:

$$A, B \;::=\; p \mid A \to B$$

Type assignments:

- $\Gamma \vdash t : A$ - for terms;
- $\Gamma; B \vdash k : A$ - for contexts

$$\frac{}{\Gamma, x : A \vdash x : A}\,(Ax) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma; A \vdash \widehat{x}.t : B}\,(Sel)$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\,(\to_R) \qquad \frac{\Gamma \vdash t : A \quad \Gamma; B \vdash k : C}{\Gamma; A \to B \vdash t :: k : C}\,(\to_L)$$

$$\frac{\Gamma \vdash t : A \quad \Gamma; A \vdash k : B}{\Gamma \vdash tk : B}\,(Cut)$$

Properties

- Subject reduction
- Strong normalisation property (Typeability implies SN)
- Characterisation of strong normalisation with intersection types (SN implies typeability)

  Typeability $\Leftrightarrow$ SN

  [Espirito Santo, S.G., Ivetić, Likavec]

Properties

- Subject reduction
- Strong normalisation property (Typeability implies SN)
- Characterisation of strong normalisation with intersection types (SN implies typeability)

  $$\text{Typeability} \Leftrightarrow \text{SN}$$

  [Espirito Santo, S.G., Ivetić, Likavec]
- ΛJ - generalised application [Matthes]
- λx - explicit substitutions [Rose and Bloo]
- $\lambda x \cap$ - with intersection types [S.Lengrand et al.]

Outline

Sequent calculus - LJ
EXPLICIT structural rules

$$\frac{}{A \vdash A}\,(Ax)$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}\,(\rightarrow_R) \qquad \frac{\Gamma \vdash A \qquad \Delta, B \vdash C}{\Gamma, \Delta, A \rightarrow B \vdash C}\,(\rightarrow_L)$$

$$\frac{\Gamma \vdash A \qquad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}\,(Cut)$$

$$\frac{\Gamma \vdash B}{\Gamma, A \vdash B}\,(Weak) \qquad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\,(Cont)$$

Implicit structural rules

- contexts are sets
- Axiom: $\Gamma, A \vdash A$
- context-sharing rules

Explicit structural rules

- contexts are multisets
- Axiom: $A \vdash A$
- context-splitting rules

Sequent lambda calculus with resource control- ®$\underline{\lambda}$

- Derived from λ^{Gtz} by adding explicit operators for weakening and contraction
- Inspired by λlxr-calculus [Kesner and Lengrand, 2005]
- Terms control resources:
  1. every variable occurs (at most) once
  2. every binder does bind an occurrence of a free variable

Example

$I = \lambda x.x$

$K = \lambda xy.x$

$\Delta = \lambda x.x(x :: \widehat{z}.z) \qquad (\lambda x.xx)$

$W = \lambda xy.x(y(y :: \widehat{z}.z) :: \widehat{z}.z) \qquad (\lambda xy.xyy)$

The syntax of ®$\underline{\lambda}$

Values	T	$::=$	$x \mid \lambda x.t \mid x \odot t \mid x <^{x_1}_{x_2} t$
Terms	t	$::=$	$T \mid tk$
Contexts	k	$::=$	$\widehat{x}.t \mid t :: k \mid x \odot k \mid x <^{x_1}_{x_2} k$

New features:

- new constructors for weakening ($x \odot e$) and contraction($x <^{x_1}_{x_2} e$)
- value - a new syntactic category for regaining confluence

Example

$I = \lambda x.x$	
$K = \lambda xy.x$	$\lambda xy.y \odot x$
$\Delta = \lambda x.x(x :: \widehat{z}.z)$	$\lambda x.x <^{x_1}_{x_2} (x_1(x_2 :: \widehat{z}.z))$
$W = \lambda xy.x(y(y :: \widehat{z}.z) :: \widehat{z}.z)$	$\lambda xy.x(y <^{y_1}_{y_2} (y_1(y_2 :: \widehat{z}.z) :: \widehat{z}.z)$

Reduction rules of ®$\underline{\lambda}$

Reductions of λ^{Gtz}:

$$\begin{array}{lrcl} (\beta) & (\lambda x.t)(u :: k) & \to & u(\widehat{x}.tk) \\ (\pi) & (tk)k' & \to & t(k@k') \\ (\mu) & \widehat{x}.xk & \to & k \end{array}$$

Substitution: (σ) is divided into several (5) reductions

Contraction propagation:

$$(\gamma_1) \quad x <^{x_1}_{x_2} (\lambda y.t) \quad \to \quad \lambda y.x <^{x_1}_{x_2} t$$

Weakening extraction:

$$(\omega_1) \quad \lambda x.(y \odot t) \quad \to \quad y \odot (\lambda x.t), \quad x \neq y$$

Simply typed ®$\underline{\lambda}$

$$\frac{}{x : A \vdash x : A}\ (Ax)$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\ (\to_R) \qquad \frac{\Gamma \vdash t : A \quad \Delta; B \vdash k : C}{\Gamma, \Delta; A \to B \vdash t :: k : C}\ (\to_L)$$

$$\frac{\Gamma \vdash t : A \quad \Delta; A \vdash k : B}{\Gamma, \Delta \vdash tk : B}\ (Cut) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma; A \vdash \widehat{x}.t : B}\ (Sel)$$

$$\frac{\Gamma, x : A, y : A \vdash t : B}{\Gamma, z : A \vdash z <^{x}_{y} t : B}\ (Cont_t) \qquad \frac{\Gamma \vdash t : B}{\Gamma, x : A \vdash x \odot t : B}\ (Weak_t)$$

$$\frac{\Gamma, x : A, y : A; C \vdash k : B}{\Gamma, z : A; C \vdash z <^{x}_{y} k : B}\ (Cont_k) \qquad \frac{\Gamma; C \vdash k : B}{\Gamma, x : A; C \vdash x \odot k : B}\ (Weak_k)$$

The properties of $\text{®}\underline{\lambda}$

- Curry Howard correspondence with **BCKW** and substructural logics **BCK**, **BCW**

The properties of ®$\underline{\lambda}$

- Curry Howard correspondence with **BCKW** and substructural logics **BCK**, **BCW**
- *Lambek calculus - associativity* ***B****, commutativity* ***C****?*

The properties of ®$\underline{\lambda}$

- Curry Howard correspondence with **BCKW** and substructural logics **BCK**, **BCW**
- *Lambek calculus - associativity **B**, commutativity **C**?*
- Confluence - parallel reductions technique (Takahashi)
- Subject reduction - typing rules syntax directed
- Strong normalization - rewriting techniques

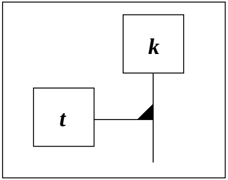
The properties of ®$\underline{\lambda}$

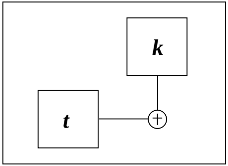
- Curry Howard correspondence with **BCKW** and substructural logics **BCK**, **BCW**
- *Lambek calculus - associativity **B**, commutativity **C**?*

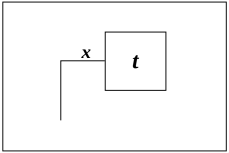
- Confluence - parallel reductions technique (Takahashi)
- Subject reduction - typing rules syntax directed
- Strong normalization - rewriting techniques

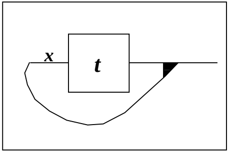
- *Characterization of strong normalization?*
- *Diagrammatic representations?*

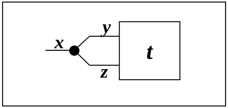
Ongoing - Diagrams for ®$\underline{\lambda}$

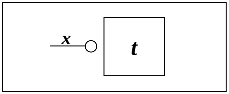
Cons: 

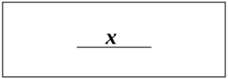
Application: 

Selection: 

Abstraction: 

Contraction: 

Weakening: 

Variable: 

Rewriting, Deduction and Programming RDP 2011

May 29 – June 3, 2011
Novi Sad, Serbia
http://www.rdp2011.uns.ac.rs/

- **RTA 2011** – PC Chair: *Manfred Schmidt-Schauss*
  Proceedings LIPIcs
  Deadline: **January 7, 2011**
- **TLCA 2011** – PC Chair: *Luke Ong*
  Proceedings LNCS/ARCoSS
  Deadline: **January 26, 2011**
- **Workshop** proposals
  Deadline: **November 19, 2010**

About

- RDP'11
- Venue
- Programme
- Organisation
- Sponsors

Conferences

- RTA
- TLCA

Workshops

- Call For Workshop

Practical

- Registration
- Student Grants
- Accomodation
- Travel

RDP 2011 Federated Conference on Rewriting, Deduction, and Programming

Sunday, May 29, 2011 to Friday, June 3, 2011
Novi Sad, Serbia

RDP'11 is the sixth edition of the International Conference on Rewriting, Deduction, and Programming, consisting of two main conferences

- Rewriting Techniques and Applications (RTA'11)
- Typed Lambda Calculi and Applications (TLCA'11)

and related events.

Previous editions of RDP took place in

- Valencia (Spain) 2003
- Aachen (Germany) 2004
- Nara (Japan) 2005
- Paris (France) 2007
- Brasilia (Brasil) 2009