# Computational interpretations of logics

Silvia Ghilezan
University of Novi Sad, Serbia

Belgrade, January 30, 2009

# Outline of the talk - first part

1. Computational interpretations of intuitionistic logic
    - Axiomatic System - Combinatory Logic
    - Natural Deduction - $\lambda$-calculus
    - Sequent calculus - ?
2. Sequent term calculi for intuitionistic logic
    - $\lambda LJ$-calculus
    - $\bar{\lambda}$-calculus
    - $\lambda^{Gtz}$-calculus
3. $\lambda^{Gtz}$-calculus with intersection types
    - Calculi with gen. application and explicit substitution
    - Ongoing work

# Computational interpretations of intuitionistic logic

Curry-Howard-de Brujin-Lambek correspondence

logic vs term calculus

types as formulae – terms as proofs – terms as programs

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

- axiomatic (Hilbert) system (axioms/Modus Ponens)
  *Combinatory Logic* (combinators/application)
  1930s Schönfinkel, Curry
- natural deduction (introduction/elimination)
  $\lambda$ *calculus* (abstraction/application)
  1940s Church
- sequent calculus (right/left introduction/cut)
  *various attempts* $\lambda$ *calculus*
  (abstraction/application/substitution)
  1970s

# Axiomatic (Hilbert style) system - Combinatory Logic

$(Ax1)\qquad \vdash \quad A \to A$

$(Ax2)\qquad \vdash \quad A \to (B \to A)$

$(Ax3)$

$\vdash \quad (A \to (B \to C)) \to ((A \to B) \to (A \to C))$

$$(MP)\qquad \frac{\vdash \quad A \to B \quad \vdash \quad A}{\vdash B}$$

# Axiomatic (Hilbert style) system - Combinatory Logic

$(Ax1)$ $$\vdash I : A \to A$$

$(Ax2)$ $$\vdash K : A \to (B \to A)$$

$(Ax3)$ $$\vdash S : (A \to (B \to C)) \to ((A \to B) \to (A \to C))$$

$(MP)$ $$\frac{\vdash t : A \to B \quad \vdash s : A}{ts : \vdash B}$$

# Natural Deduction - $\lambda$-calculus

$(axiom)$ $$\overline{\Gamma, \quad A \vdash \quad A}$$

$(\rightarrow_{elim})$ $$\frac{\Gamma \vdash \quad A \rightarrow B \quad \Gamma \vdash \quad A}{\Gamma \vdash \quad B}$$

$(\rightarrow_{intr})$ $$\frac{\Gamma, \quad A \vdash \quad B}{\Gamma \vdash \quad A \rightarrow B}$$

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

# Natural Deduction - $\lambda$-calculus

$(axiom)$ $$\frac{}{\Gamma, x : A \vdash x : A}$$

$(\rightarrow_{elim})$ $(app)$ $$\frac{\Gamma \vdash t : A \rightarrow B \quad \Gamma \vdash s : A}{\Gamma \vdash ts : B}$$

$(\rightarrow_{intr})$ $(abs)$ $$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \rightarrow B}$$

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

# Sequent calculus - ?

$(axiom)$ $$\frac{}{\Gamma, A \vdash A}$$

$(\rightarrow_{left})$ $$\frac{\Gamma \vdash A \quad \Gamma, B \vdash C}{\Gamma, A \rightarrow B \vdash C}$$

$(\rightarrow_{right})$ $$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}$$

$(cut)$ $$\frac{\Gamma \vdash A \quad \Gamma, A \vdash B}{\Gamma \vdash B}$$

# Sequent calculus intuitionistic logic

- Pottinger, Zucker 1970s comparing cut-elimination to proof normalization
- Gallier [1991]
- Mints [1996]
- Barendregt, Ghilezan [2000]: $\lambda LJ$-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.
- Computation over terms reflects cut-elimination
- Espírito Santo [2006]: $\lambda^{Gtz}$-calculus

# $\lambda LJ$-calculus

Barendregt and Ghilezan

- term calculus $\lambda$-calculus
- type system LJ

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

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

# $\lambda LJ$-calculus

Barendregt and Ghilezan

- term calculus $\lambda$-calculus - natural deduction term structure
- type system LJ - sequent types 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 $\lambda LJ$ to $\bar{\lambda}$-calculus

$\lambda LJ$-calculus:

- Using a subsystem $\lambda LJ^{cf}$ 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 $\lambda LJ$ to $\bar{\lambda}$-calculus

## $\lambda LJ$-calculus:

- Using a subsystem $\lambda LJ^{cf}$ 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
  instead of $((yu_1)...u_n)$ the applicative part is $y[u_1; ...; u_n]$.

# $\bar{\lambda}$-calculus

Syntax:

$$\begin{array}{lrcl} \text{(Terms)} & t, u, v & ::= & xl \mid \lambda x.t \mid tl \mid t\langle x = v\rangle \\ \text{(Lists)} & l, l' & ::= & [\,] \mid t :: l \mid l@l' \mid l\langle x = t\rangle \end{array}$$

Reduction rules:

$$\begin{array}{rrcl} (\beta_{cons}) & \lambda x.u(v :: l) & \to & u\langle x = v\rangle l \\ (\beta_{nil}) & \lambda x.u[\,] & \to & \lambda x.u \\ (C_{var}) & (tl)l' & \to & t(l@l') \\ (C_{cons}) & (t :: l)@l' & \to & t :: (l@l') \\ (C_{nil}) & [\,]@l & \to & l \\ (S_{yes}) & (xl)\langle x = v\rangle & \to & vl\langle x = v\rangle \\ (S_{no}) & (yl)\langle x = v\rangle & \to & yl\langle x = v\rangle \\ (S_{\lambda}) & (\lambda y.u)\langle x = v\rangle & \to & \lambda y.(u\langle x = v\rangle) \\ (S_{nil}) & [\,]\langle x = v\rangle & \to & [\,] \\ (S_{cons}) & (u :: l)\langle x = v\rangle & \to & u\langle x = v\rangle :: l\langle x = v\rangle. \end{array}$$

# $\bar{\lambda}$ - simple types

$$\frac{}{\Gamma\,;\,.:A \vdash (\,.[\,])  : A}\,(Ax) \qquad \frac{\Gamma, x:A\,;\,.:A \vdash (\,.l) : B}{\Gamma, x:A; \vdash xl : B}\,(Cont)$$

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

$$\frac{\Gamma; \vdash t : A \quad \Gamma\,;\,.:A \vdash (\,.l) : B}{\Gamma; \vdash tl : B}\,(C_{H1}) \qquad \frac{\Gamma\,;\,.:A \vdash (\,.l) : C \quad \Gamma\,;\,.:C \vdash (\,.l') : B}{\Gamma\,;\,.:A \vdash (\,.l@l') : B}\,(C_{H2})$$

$$\frac{\Gamma; \vdash t : A \quad \Gamma, x:A; \vdash u : B}{\Gamma; \vdash u\langle x = t\rangle : B}\,(C_{M1}) \qquad \frac{\Gamma; \vdash t : C \quad \Gamma, x:C\,;\,.:A \vdash (\,.l) : B}{\Gamma\,;\,.:A \vdash (\,.l\langle x = t\rangle) : B}\,(C_{M2})$$

Curry-Howard correspondence: normal forms of $\bar{\lambda}$ correspond to cut-free proofs in *LJT*.

# $\lambda^{Gtz}$-calculus

Syntax

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

Reductions

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

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

$$(u :: k)@k' = u :: (k@k') \qquad (\hat{x}.t)@k' = \hat{x}.tk'.$$

# $\lambda^{Gtz}$ - simple types

Types:

$$\mathrm{A}, \mathrm{B} \quad ::= \quad \mathrm{X} \mid \mathrm{A} \to \mathrm{B}$$

Type assignements:

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

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

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

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

# Properties of $\lambda^{Gtz}$

- - Strong normalisation property (Typeability implies SN)
  - Characterisation of strong normalisation (SN implies typeability)???? -fails (normal forms not typeable)

  intersection types
- History: Intersection types are devised in $\lambda$ calculus to capture all strongly normalizing terms
  Coppo, Dezani, Pottinger, Salé 1980s
- Joint ongoing work with
  - *Jose Espírito Santo*, University of Minho, Portugal
  - *Jelena Ivetic*, University of Novi Sad, Serbia
  - *Silvia Likavec*, University of Turin, Italy

# Type system $\lambda^{Gtz}\cap$

$$\frac{}{\Gamma, x : \cap A_i \vdash x : A_i \quad i \geq 1}\,(Ax) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\,(\to_R)$$

$$\frac{\Gamma \vdash t : A_1 \ldots \Gamma \vdash t : A_n \quad \Gamma; B \vdash k : C}{\Gamma; \cap A_i \to B \vdash t :: k : C}\,(\to_L) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma; A \vdash \hat{x}.t : B}\,(Sel)$$

$$\frac{\Gamma \vdash t : A_1 \quad \ldots \Gamma \vdash t : A_n \quad \Gamma; \cap A_i \vdash k : B}{\Gamma \vdash tk : B}\,(Cut)$$

# Calculi with generalised application and explicit substitution

- $\lambda J$ ($\Lambda J$) - $\lambda$-calculus with generalised application [Matthes]
  - *app*2 extra rule in order to characterise SN with intersection types
- $\lambda x$ - $\lambda$-calculus with explicit substitutions [Rose and Bloo]
  - $K - Cut$ and *drop* extra rule in order to characterise SN with intersection types [Lengrand et al.]

Subclasses of $\lambda^{Gtz}$-terms:

$$\begin{array}{llll}(\lambda J\text{-terms}) & t,u,v & ::= & x \mid \lambda x.t \mid t(u,x.v) \\ (\lambda x\text{-terms}) & t,u,v & ::= & x \mid \lambda x.t \mid t(u) \mid v\langle x = t\rangle \\ (\lambda\text{-terms}) & t,u,v & ::= & x \mid \lambda x.t \mid t(u)\end{array}$$

where

- $t(u,x.v)$ denotes $t(u :: \hat{x}.v)$
- $v\langle x = t\rangle$ denotes $t(\hat{x}.v)$
- $t(u)$ denotes $t(u :: \hat{x}.x)$.

# Ongoing work

- Proper and strict types
- Characterisation of weak normalisation

Publications:

- J. Espírito Santo, S. Ghilezan, J. Ivetić: ”Characterizing strongly normalising intuitionistic sequent terms”. TYPES 2007, Lecture Notes in Computer Science 4941: 85-99 (2007)
- J. Espírito Santo, J. Ivetić, S. Likavec: "Intersection types for intuitionistic sequent terms". ITRS’08.

# Outline of the talk - second part

1. Computational interpretations of classical logic
2. Sequent term calculus for classical logic
   - $\overline{\lambda}\mu\widetilde{\mu}$ calculus
   - Duality of computation - CBV vs CBN
3. $\overline{\lambda}\mu\widetilde{\mu}$with intersection types
   - Typeability implies SN
   - Ongoing work

# Computational interpretations of classical logic

- Griffin 1990
  *formulae-as-types notion of control (axiomatic)*
- Parigot 1992
  *algorithmic interpretation of classical logic (natural deduction)*
- Barbanera, Berardi 1996
  *symmetric lambda calculus - classical program extraction*
- Curien, Herbelin 2000
  *symmetric lambda calculus - duality of computation*
- Wadler 2003
  *dual calculus - duality of computation*
- Urban 2000
  *symmetric lambda calculus - cut elimination in classical logic*
- Lescanne, van Bakel 2005

# $\overline{\lambda}\mu\widetilde{\mu}$ calculus

Curien, Hereblin [2000]

Syntax

$$\begin{array}{llll} \text{Term:} & r & ::= & x \mid \lambda x.r \mid \mu\alpha\,.\,c \\ \text{Coterm:} & e & ::= & \alpha \mid r \bullet e \mid \widetilde{\mu}x\,.\,c \\ \text{Command:} & c & ::= & \langle r \parallel e \rangle \end{array}$$

Reduction rules

$$\begin{array}{lrcl} (\lambda) & \langle \lambda x.r \parallel s \bullet e \rangle & \longrightarrow & \langle s \parallel \widetilde{\mu}x.\langle r \parallel e \rangle \rangle \\ (\mu - red) & \langle \mu\alpha.c \parallel e \rangle & \longrightarrow & c\{e/\alpha\} \\ (\widetilde{\mu} - red) & \langle r \parallel \widetilde{\mu}x.c \rangle & \longrightarrow & c\{r/x\} \end{array}$$

# Classical Sequent Calculus

$$\frac{}{\Gamma \quad A \vdash \Delta, \quad A}\,(axR) \qquad \frac{}{A, \Gamma \vdash \quad A, \Delta}\,(axL)$$

$$\frac{\Gamma \vdash \Delta, \quad A \qquad B, \Gamma \vdash \Delta}{A \to B, \Gamma \vdash \Delta}\,(\to L) \qquad \frac{\Gamma, \quad A \vdash \Delta, \quad B}{\Gamma \vdash \Delta, \quad A \to B}\,(\to R)$$

$$\frac{\Gamma \vdash \Delta, \quad A \qquad A, \Gamma \vdash \Delta}{: \ (\Gamma \vdash \Delta)}\,(cut)$$

# Proof terms for Classical Sequent Calculus – $\overline{\lambda}\mu\widetilde{\mu}$ calculus

$$\frac{}{\Gamma, x : A \vdash \Delta, x : A}(axR) \qquad \frac{}{\alpha : A, \Gamma \vdash \alpha : A, \Delta}(axL)$$

$$\frac{\Gamma \vdash \Delta, r : A \qquad e : B, \Gamma \vdash \Delta}{r \bullet e : A \to B, \Gamma \vdash \Delta}(\to L) \qquad \frac{\Gamma, x : A \vdash \Delta, r : B}{\Gamma \vdash \Delta, \lambda x.r : A \to B}(\to R)$$

$$\frac{c : (\Gamma, x : A \vdash \Delta)}{\widetilde{\mu} x.c : A, \Gamma \vdash \Delta}(\widetilde{\mu}) \qquad \frac{c : (\Gamma \vdash \alpha : A, \Delta)}{\Gamma \vdash \Delta, \mu\alpha.c : A}(\mu)$$

$$\frac{\Gamma \vdash \Delta, r : A \qquad e : A, \Gamma \vdash \Delta}{\langle r \parallel e \rangle : (\Gamma \vdash \Delta)}(cut)$$

# Critical pair, failure of confluence, nondeterminism

Failure of confluence

$$\langle\ \mu\alpha.\langle y \parallel \beta\rangle \parallel \widetilde{\mu}x.\langle z \parallel \gamma\rangle\ \rangle \rightarrow \langle y \parallel \beta\rangle$$

$$\langle\ \mu\alpha.\langle y \parallel \beta\rangle \parallel \widetilde{\mu}x.\langle z \parallel \gamma\rangle\ \rangle \rightarrow \langle z \parallel \gamma\rangle$$

# Critical pair, failure of confluence, nondeterminism

Failure of confluence

$$\langle\ \mu\alpha.\langle y \parallel \beta\rangle \parallel \widetilde{\mu}x.\langle z \parallel \gamma\rangle\ \rangle \to \langle y \parallel \beta\rangle$$

$$\langle\ \mu\alpha.\langle y \parallel \beta\rangle \parallel \widetilde{\mu}x.\langle z \parallel \gamma\rangle\ \rangle \to \langle z \parallel \gamma\rangle$$

Reflects a well-known phenomenon in cut-elimination

# Duality of computation: Call-by-value, Call-by-name

$$\begin{array}{llcll} (\mu - red) & \langle \mu\alpha.c \parallel \widetilde{\mu}x.d \rangle & \longrightarrow & c\{\widetilde{\mu}x.c/\alpha\} & CBV \\ (\widetilde{\mu} - red) & \langle \mu\alpha.c \parallel \widetilde{\mu}x.d \rangle & \longrightarrow & d\{\mu\alpha.c/x\} & CBN \end{array}$$

Two confluent subsystems of $\overline{\lambda}\mu\widetilde{\mu}_{CBV}$ and $\overline{\lambda}\mu\widetilde{\mu}_{CBN}$

Strong normalization of CBV and CBN:

- CPS translations of $\overline{\lambda}\mu\widetilde{\mu}_{CBV}$ and $\overline{\lambda}\mu\widetilde{\mu}_{CBN}$ into simply-typed $\lambda$-calculus

# Duality of computation: Call-by-value, Call-by-name

$$\begin{array}{llcll} (\mu - red) & \langle \mu\alpha.c \parallel \widetilde{\mu}x.d \rangle & \longrightarrow & c\{\widetilde{\mu}x.c/\alpha\} & CBV \\ (\widetilde{\mu} - red) & \langle \mu\alpha.c \parallel \widetilde{\mu}x.d \rangle & \longrightarrow & d\{\mu\alpha.c/x\} & CBN \end{array}$$

Two confluent subsystems of $\overline{\lambda}\mu\widetilde{\mu}_{CBV}$ and $\overline{\lambda}\mu\widetilde{\mu}_{CBN}$

Strong normalization of CBV and CBN:

- CPS translations of $\overline{\lambda}\mu\widetilde{\mu}_{CBV}$ and $\overline{\lambda}\mu\widetilde{\mu}_{CBN}$ into simply-typed $\lambda$-calculus

Strong normalization of free reduction?

# Previous work

Typeability implies SN: results for sequent-based term calculi:

- Barbanera and Berardi (symmetric candidates based on fixed points)
- Urban and Bierman
- Lengrand
- Polonovski (symmetric candidates or explicit substitutions)
- David and Nour (arithmetic proofs)

# Previous work

Typeability implies SN: results for sequent-based term calculi:

- Barbanera and Berardi (symmetric candidates based on fixed points)
- Urban and Bierman
- Lengrand
- Polonovski (symmetric candidates or explicit substitutions)
- David and Nour (arithmetic proofs)

SN implies Typeability?
Joint work with
- *Daniel Dougherty*, Worcester Polytechnic Institute, USA
- *Pierre Lescanne*, Ecole Normale Superieure, Lyon, France

# Intersection types in $\overline{\lambda}\mu\widetilde{\mu}$ calculus

Raw types

$$\mathcal{T} ::= TVar \mid \mathcal{T} \rightarrow \mathcal{T} \mid \mathcal{T}^\circ \mid \mathcal{T} \cap \mathcal{T}$$

$A^\circ$ is said to be the *dual* type of type $A$ ($A^{\circ\circ} = A$) Taxonomy of term types and coterm types

| | term-types | coterm-types |
|---|---|---|
| | $\mathcal{T}$ | $\mathcal{T}^\circ$ |
| | $(A_1 \rightarrow A_2)$ | $(A_1 \rightarrow A_2)^\circ$ |
| for $n \geq 2$ : | $(A_1 \cap A_2 \cap \cdots \cap A_n)$ | $(A_1 \cap A_2 \cap \cdots \cap A_n)^\circ$ |
| for $n \geq 2$ : | $(A_1{}^\circ \cap A_2{}^\circ \cap \cdots \cap A_n{}^\circ)^\circ$ | $(A_1{}^\circ \cap A_2{}^\circ \cap \cdots \cap A_n{}^\circ)$ |

# Intersection types in $\overline{\lambda}\mu\widetilde{\mu}$ calculus

$$\frac{}{\Sigma,\ \ v : (T_1 \cap \cdots \cap T_k) \ \vdash \ v : T_i}\ (\mathrm{ax})$$

$$\frac{\Sigma,\ \ x : A \ \vdash \ r : B}{\Sigma \ \vdash \ \lambda x.r \ : \ A \to B}\ (\to \mathrm{r})$$

$$\frac{\Sigma \ \vdash \ r : A_i \quad i = 1, \ldots, k \qquad \Sigma \ \vdash \ e : B^{\circ}}{\Sigma \ \vdash \ r \bullet e : ((A_1 \cap \cdots \cap A_k) \to B)^{\circ}}\ (\to$$

$$\frac{\Sigma,\ \ \alpha : A^{\circ} \ \vdash \ c : \bot}{\Sigma \ \vdash \ \mu\alpha.c : A}\ (\mu)$$

$$\frac{\Sigma,\ \ x : A \ \vdash \ c : \bot}{\Sigma \ \vdash \ \widetilde{\mu}x.c : A^{\circ}}\ (\widetilde{\mu})$$

$$\frac{\Sigma \ \vdash \ r : A_1 \ \ldots \ \Sigma \ \vdash \ r : A_n \qquad \Sigma \ \vdash \ e : \cap A_i{}^{\circ}}{\Sigma \ \vdash \ \langle r \parallel e \rangle : \bot}\ (\mathrm{cut})$$

# Typeability implies SN

- The difficulty in proving SN in $\overline{\lambda}\mu\widetilde{\mu}$ using a traditional reducibility (or "candidates") argument arises from the critical pairs $\langle\mu\alpha.c \parallel \widetilde{\mu}x.d\rangle$
- Neither of the expressions here can be identified as the preferred redex one cannot define candidates by induction on the structure of types
- This difficulty arises already in the simply-(arrow)-typed case
- The "symmetric candidates" technique of Barbanera and Berardi uses a fixed-point technique to define the candidates and suffices to prove strong normalization for simply-typed $\overline{\lambda}\mu\widetilde{\mu}$
- The interaction between intersection types and symmetric candidates is technically problematic (David and Nour)

# Future work

- Characterize *weak normalization, head normalization, etc?*
- Proof technique for symmetric lambda calculi
- Cube of classical lambda calculi

# Publications

- D. Dougherty, S. Ghilezan and P. Lescanne:
  Characterizing strong normalization in the Curien-Herbelin symmetric lambda calculus: extending the Coppo-Dezani heritage, Theoretical Computer Science 398: 114-128 (2008)
- D. Dougherty, S. Ghilezan, and P. Lescanne:
  A general technique for analyzing termination in symmetric proof calculi, *Workshop on Termination WST'07*, Paris, France (2007)
- D.Dougherty, S.Ghilezan, P.Lescanne and S.Likavec:
  Strong normalization of the classical sequent calculus, *Conference on Logic Programming and Artificial Reasoning, LPAR 2005*, Jamaica, Lecture Notes in Computer Science 3835: 169-183 (2005).

# Another line of research

- H. Herbelin and S. Ghilezan:
  An approach to call-by-name delimited continuations, ACM SIGPLAN - SIGACT Symposium on Principles of Programming Languages POPL 2008 San Francisco, *ACM SIGPLAN Notices* 43 (1): 383-394 (2008)
- joint work with H. Herbelin and A. Saurin.