Making Dependent Evidence Explicit in Justification Logic

Giuseppe Primiero

FWO - Flemish Research Foundation
Centre for Logic and Philosophy of Science, Ghent University
IEG - Oxford University

Giuseppe.Primiero@Ugent.be
http://www.philosophy.ugent.be/giuseppeprimiero/

Graduate Center, City University of New York
31st May, 2011

Outline

1. Motivation
2. Intuitionistic JL with Dependency
3. Natural Deduction with Global and Local Assumptions
4. Normalization
5. Summary

1. Motivation
2. Intuitionistic JL with Dependency
3. Natural Deduction with Global and Local Assumptions
4. Normalization
5. Summary

What are Dependent Justifications

- The notion of dependent justification is inspired by its formal counterpart in theories of dependent types:

What are Dependent Justifications

- The notion of dependent justification is inspired by its formal counterpart in theories of dependent types:
  - A dependent type is a type expression of the form $B[x]$ with x a free variable ranging over A *type* saying that B is a type whenever $x \in A$;

What are Dependent Justifications

- The notion of dependent justification is inspired by its formal counterpart in theories of dependent types:
  - A dependent type is a type expression of the form $B[x]$ with x a free variable ranging over A *type* saying that B is a type whenever $x \in A$;
  - Σ type: type of all pairs $\langle a, b \rangle$ where $a \in A$ and $b \in B[a]$;

What are Dependent Justifications

- The notion of dependent justification is inspired by its formal counterpart in theories of dependent types:
  - A dependent type is a type expression of the form $B[x]$ with x a free variable ranging over A *type* saying that B is a type whenever $x \in A$;
  - Σ type: type of all pairs $\langle a, b \rangle$ where $a \in A$ and $b \in B[a]$;
  - Π type: type of all functions $\lambda x.b[x]$ where $b[a] \in B[a]$ for any $a \in A$;

What are Dependent Justifications

- The notion of dependent justification is inspired by its formal counterpart in theories of dependent types:
  - A dependent type is a type expression of the form $B[x]$ with x a free variable ranging over A *type* saying that B is a type whenever $x \in A$;
  - Σ type: type of all pairs $\langle a, b \rangle$ where $a \in A$ and $b \in B[a]$;
  - Π type: type of all functions $\lambda x.b[x]$ where $b[a] \in B[a]$ for any $a \in A$;
  - Propositional functions under the props-as-types analogy

What are Dependent Justifications (II)

- A dependent type expresses the definitional relation of a term from a term (parametrization of types);

What are Dependent Justifications (II)

- A dependent type expresses the definitional relation of a term from a term (parametrization of types);
- Similarly, we can express a justification for B dependently on a justification for A:

  "t is a justification for B, whenever A is justified by s"

What are Dependent Justifications (II)

- In a natural deduction translation, this can be reconstructed as validity under assumptions:

What are Dependent Justifications (II)

- In a natural deduction translation, this can be reconstructed as validity under assumptions:

- If $t : B$ depends from $s : A$, a formula of the form $\Delta; \vdash B \mid t$ where $A \in \Delta$ can be used in ND (see [Artëmov and Bonelli, 2007], [Alt and Artemov, 2001]);

$$\frac{\Delta; \cdot \vdash B \mid t}{\Delta; \Gamma \vdash [\![t]\!]B \mid !t} \Box I$$

What are Dependent Justifications (II)

- In a natural deduction translation, this can be reconstructed as validity under assumptions:

- If $t : B$ depends from $s : A$, a formula of the form $\Delta; \vdash B \mid t$ where $A \in \Delta$ can be used in ND (see [Artëmov and Bonelli, 2007], [Alt and Artemov, 2001]);

$$\frac{\Delta; \cdot \vdash B \mid t}{\Delta; \Gamma \vdash [\![t]\!]B \mid !t} \Box I$$

- We want to express the local dependency of a term from a term (evidences);

What are Dependent Justifications (II)

- In a natural deduction translation, this can be reconstructed as validity under assumptions:
- If $t : B$ depends from $s : A$, a formula of the form $\Delta; \vdash B \mid t$ where $A \in \Delta$ can be used in ND (see [Artëmov and Bonelli, 2007], [Alt and Artemov, 2001]);

$$\frac{\Delta; \cdot \vdash B \mid t}{\Delta; \Gamma \vdash [\![t]\!]B \mid !t} \Box I$$

- We want to express the local dependency of a term from a term (evidences);
- We want also to preserve the global dependency of expressions (propositions, as in the λ-calculi).

Tasks

1. Show that a notion of Dependent Evidence mimicking Functions can be formally accomodated in the framework of Justification Logic for expressions of the form

   "t is a justification for B, whenever A is justified by s"

Tasks

1. Show that a notion of Dependent Evidence mimicking Functions can be formally accomodated in the framework of Justification Logic for expressions of the form

   "t is a justification for B, whenever A is justified by s"

2. Provide a general interpretation of functional expressions within a ND system with
   - Dependent Terms
   - Dependent Expressions

Tasks

1. Show that a notion of Dependent Evidence mimicking Functions can be formally accomodated in the framework of Justification Logic for expressions of the form

   "t is a justification for B, whenever A is justified by s"

2. Provide a general interpretation of functional expressions within a ND system with
   - Dependent Terms
   - Dependent Expressions

3. Use the latter to prove some metatheoretical results.

1 Motivation

2 Intuitionistic JL with Dependency

3 Natural Deduction with Global and Local Assumptions

4 Normalization

5 Summary

Axioms and Inference Schemes

Definition (Axioms)

Axioms of the system are:

A0. Axioms schemes of minimal logic in the the language of JL

A1. $[\![s]\!]A \supset A$ "verification"

A2. $[\![s]\!]A \supset [\![!s]\!][\![s]\!]A$ "proof checker"

A3. $A \supset \ll s \gg A$ "assumption maker"

A4. $\ll s \gg A \supset \ll ?s \gg [\![s]\!]A$ "proof dependency maker"

A5. $\ll s \gg A \supset \sim \ll s \gg \sim A$ "consistency of assumption"

A6. $[\![s]\!](A \supset B) \supset ([\![t]\!]A \supset [\![s \cdot t]\!]B)$ "application"

R1. $\Gamma \vdash A \supset B$ and $\Gamma \vdash A$ implies $\Gamma \vdash B$ "modus ponens"

R2. If **A** is an axiom **A0.** – **A6.** and c is a proof constant, then $\vdash [\![c]\!]\mathbf{A}$ "necessitation"

Meaning of Dependent Evidence

- A proper functional or dependent evidence will be given by the construction of $A4$. with distinct polynomials

$$\ll ?s \gg [\![t]\!](B[A])$$

Meaning of Dependent Evidence

- A proper functional or dependent evidence will be given by the construction of $A4$. with distinct polynomials

$$\ll ?s \gg [\![t]\!](B[A])$$

- It reads:

  "t is an evidence of B, whenever s is an evidence of A".

Meaning of Dependent Evidence

- A proper functional or dependent evidence will be given by the construction of *A*4. with distinct polynomials

$$\ll ?s \gg [\![t]\!](B[A])$$

- It reads:

  "t is an evidence of B, whenever s is an evidence of A".

- This gives a way to express in JL expressions of the form:

  "Let A be a proposition with evidence s, and B a proposition dependent on A"

Constructing dependent evidence

$$\frac{A \supset \ll s \gg A \qquad \ll ?s \gg [\![t]\!]B[A]}{(?s)t : B}\ \mathrm{DE}$$

Given an assumed evidence s for A, if there is an evidence $\ll ?s \gg [\![t]\!]B[A]$ of B dependent on A, then an evidence t of B holds assuming an evidence s.

Constructing dependent evidence (Equivalence)

$$\frac{A \equiv A' \supset \ll s \gg A \equiv \ll s' \gg A' \qquad \ll ?s \gg \equiv \ll ?s' \gg [\![t]\!]B[A] \equiv [\![t']\!]B'[A']}{(?s)t : B \equiv (?s')t' : B'} \text{ EQ}$$

We get identical evidences $(?s)t, (?s')t'$ for B, B' when equivalent dependent evidences $\ll ?s \gg [\![t]\!], \ll ?s' \gg [\![t']\!]$ are formulated from equivalent assumed evidences s, s' respectively for A, A'.

Rules for dependent evidence

Discharging of such dependency relation is given by Application:

$$\frac{\ll ?s \gg t : B[A] \qquad [\![s]\!]A}{[\![s]\!] \cdot [\![t]\!] : B} \text{ Function Application}$$

$$\frac{\ll ?s \gg t \equiv \ll ?s' \gg t' : B[A] \equiv B'[A'] \qquad [\![s]\!] \equiv [\![s']\!]A}{[\![s]\!] \cdot [\![t]\!] \equiv [\![s']\!] \cdot [\![t']\!] : B \equiv B'} \text{ EQ}$$

The first rule says that given an evidence $\ll ?s \gg [\![t]\!]B[A]$ of B dependent on A, if there is an evidence s of A, then $[\![s]\!] \cdot [\![t]\!]$ is an evidence of B. Equality holds under equivalent dependent evidences.

Rules for dependent evidence

To construct a functional evidence one can proceed by abstraction on a dependent evidence (plus identity rules):

$$\frac{\ll ?s \gg [\![t]\!] : B[A]}{(?s)t : (\ll s \gg A)B} \text{ Abstraction}$$

$$\frac{[\![s]\!]A \qquad \ll ?s \gg [\![t]\!]B[A]}{(?s)t(!s) = [\![s]\!] \cdot [\![t]\!] : B} \ \beta\text{-rule}$$

$$\frac{\ll ?s \gg [\![t]\!] \equiv \ll ?s \gg [\![t']\!] : B[A]}{(?s)t \equiv (?s)t' : (\ll s \gg A)B} \ \xi\text{-rule}$$

$$\frac{\ll ?s \gg [\![t]\!] : B[A]}{(?s)t = (s')(\ll ?s = ?s' \gg)[\![t]\!] : (\ll s \gg A)B} \ \alpha\text{-rule}$$

where variables in s' are not free in s.

$$\frac{\ll ?s \gg t : B[A]}{(?s)t(!s) \equiv \ll ?s \gg [\![t]\!] : B[A]} \ \eta\text{-rule}$$

where variables in s are not free variables in t.

Dependency from more than one term

The basic case with two assumptions is of the from $\ll s_2 \gg\ll s_1 \gg [\![t]\!]B[A_2[A_1]]$ which reads informally as follows: "t is an evidence of B, provided s_2 is an evidence of A_2, which holds provided s_1 is an evidence of A_1".

$$\frac{\ll ?s_1 \ldots ?s_n \gg [\![t]\!]B[A_1, \ldots, A_n]}{(?s_1 \ldots (?s_{n-1}(?s_n)))t : (\ll ?s_1 \gg A_1, \ll ?s_2 \gg A_2, \ldots, \ll ?s_n \gg A_n)B}$$

"if t is a proof for B dependent on evidences for $A_1, \ldots, A_n$, then t is an evidence for B given that s_n is an evidence for A_n, which holds provided s_{n-1} is an evidence for A_{n-1}, which holds provided ... up to s_1 is an evidence of A_1".

Dependency from more than one term

By repeated application we obtain the inverse operation:

$$\frac{(?s_1(\ldots(\ldots,(?s_n))))t:(A_1,\ldots,A_n)B \qquad [\![s_1]\!]A_1,[\![s_1\cdot,\ldots,\cdot[\![s_{n-1}\cdot s_n]\!]A_n}{[\![[\![s_1]\!]\cdot[\![s_2]\!]\cdot,\ldots,\cdot[\![s_n]\!]]\!]\cdot[\![t]\!]:B}$$

which reads: "if t is an evidence for B given evidences for $A_1,\ldots,A_n$ and provided s_n is a proof of A_n provided s_{n-1} is a proof of A_{n-1} up to s_1 is a proof of A_1, then t is an evidence for B dependent on evidences s_1 applied to s_2, then applied to s_3 up to s_n".

Example 1

Show that a function evidence $C(y)[y:(B(x)[x:A]\!]$ is given by a function $C(x,y)$ with arguments respectively in A and $B[A]$.

$$\dfrac{\dfrac{(?c)c(?c,(a,b))C[!b:B(!a)] \qquad ?c(([\![!a]\!]:A)[\![!b]\!]:B)C}{(?c)c(c,((?a)?b))C}\text{ Abstraction}}{(a\cdot b)\cdot c:C}\text{ Application}$$

Example 2

Show that for any elements s, s', if $\llbracket s \rrbracket \equiv \llbracket s' \rrbracket A$, then $\ll s \gg \llbracket t \rrbracket B[A] \supset \ll s' \gg \llbracket t \rrbracket B[A]$,

$$
\dfrac{\dfrac{\dfrac{\dfrac{(?s)b\!:\!(\ll s \gg A)B \qquad \dfrac{\dfrac{\llbracket s \rrbracket A \qquad \llbracket s' \rrbracket A \qquad A \equiv A}{\llbracket s \rrbracket \equiv \llbracket s' \rrbracket A} \qquad \ll s \gg \llbracket t \rrbracket B[A]}{\ll s \gg \llbracket t \rrbracket B[A] \equiv \ll s' \gg \llbracket t \rrbracket B[A]}}{(?s')b\!:\!(\ll s \gg A)B}}{(?b)b\!:\!(\ll s \gg A)B \supset (\ll s' \gg A)B}}{(?b')(?b)b\!:\!(\llbracket s \rrbracket \equiv \llbracket s' \rrbracket A) \supset ((\ll s \gg A)B \supset (\ll s' \gg A)B)}}{(?s)(?s')(?b')(?b)b\!:\!(s\!:\!A)(s'\!:\!A)(s \equiv s'\!:\!A) \supset ((\ll s \gg A)B \equiv (\ll s' \gg A)B)}
$$

1. Motivation

2. Intuitionistic JL with Dependency

3. **Natural Deduction with Global and Local Assumptions**

4. Normalization

5. Summary

Functions in ND

- Usual translation of functional language in ND: The proof of an implication $A \supset B$ represented by a function which maps proofs of A to proofs of B, e.g. [Pfenning, 2004];

Functions in ND

- Usual translation of functional language in ND: The proof of an implication $A \supset B$ represented by a function which maps proofs of A to proofs of B, e.g. [Pfenning, 2004];

- keep them apart:
  - an implication talks about a term/program for A transformed into one for B; evaluation of terms establishes global validity (a is evaluated and a procedure to get b holds);
  - a function talks about a the dependency of terms in B from terms in A; it expresses the validity of the former locally in view of the latter;
  - the application of a function generates an assumptions which if satisfied instantiates an implication.

Functions in ND

1. Derivability of a term under valid assumptions (global validity) defines Unconditional Evidence;

$$\Delta; \cdot \vdash A \mid s \qquad \text{UnEvid}$$

Functions in ND

1. Derivability of a term under valid assumptions (global validity) defines Unconditional Evidence;

$$\Delta; \cdot \vdash A \mid s \qquad \text{UnEvid}$$

2. Derivability of a term under true assumptions (local validity) defines Dependent Evidence;

$$\Delta; \Gamma \vdash A \parallel s \qquad \text{DepEvid}$$

$IJL_{nd\Diamond}$

Definition (Language)

The syntax is defined by the following alphabet:

Proof Terms
$s := x \mid s \cdot s \mid !s \mid XTRT\ s\ AS\ v{:}A\ IN\ s \mid ?s \mid ASSM\ s\ AS\ a{:}A\ INs$

Propositions
$A := P \mid A \supset B \mid B[A] \mid [\![s]\!]A \mid \ll s \gg A \mid \ll s \gg [\![t]\!]B[A]$

Truth Contexts $\Gamma := \cdot \mid \Gamma, a{:}A$

Validity Contexts $\Delta := \cdot \mid \Delta, v{:}A$

$IJL_{nd\Diamond}$

Definition (The Logic $JL_{nd\Diamond}$)

$JL_{nd\Diamond}$ is defined by the following schemes:

$$\frac{}{\Delta; v:A, \Delta' \vdash A \mid v}\ ValVar$$

$$\frac{\Delta, v:A; \cdot \vdash B \mid s}{\Delta; \Gamma \vdash A \supset B \mid \lambda v:A.s} \supset I \qquad \frac{\Delta; \cdot \vdash A \supset B \mid s \qquad \Delta; \cdot \vdash A \mid t}{\Delta; \Gamma \vdash B \mid s \cdot t} \supset E$$

$$\frac{}{\Delta; a:A; \cdot \vdash A \mid\mid a}\ TruVar$$

$$\frac{\Delta; a:A \vdash B \mid\mid t}{\Delta; \cdot \vdash \ll s \gg [\![t]\!]B[A]}\ Function\ Formation$$

$$\frac{\Delta; \Gamma \vdash \ll s \gg [\![t]\!]B[A] \qquad \Delta; \Gamma \vdash [\![s]\!]A \mid !s}{\Delta; \Gamma \vdash B \mid !([\![s]\!] \cdot [\![t]\!])}\ Function\ Application$$

$IJL_{nd\Diamond}$

Now modalities can be used to internalise dependencies:

Definition

$$\frac{\Delta; \cdot \vdash A \mid s}{\Delta; \Gamma \vdash [\![s]\!]A \mid !s} \Box I \qquad \frac{\Delta; \cdot \vdash [\![r]\!]A \mid !s \qquad \Delta, v{:}A \vdash C \mid t}{\Delta; \Gamma \vdash C^v_r \mid XTRT\ s\ AS\ v{:}A\ IN\ t} \Box E$$

$$\frac{\Delta; \Gamma \vdash A \mid\mid s}{\Delta; \Gamma; \cdot \vdash \ll s \gg A \mid ?s} \Diamond I$$

$$\frac{\Delta; \Gamma \vdash \ll r \gg A \mid ?s \qquad \Delta, a{:}A; \cdot \vdash C \mid\mid t}{\Delta; \Gamma; \cdot \vdash C^a_r \mid\mid ASSM\ s\ AS\ a{:}A\ IN\ t} \Diamond E$$

$IJL_{nd\Diamond}$

Lemma (Properties)

The system satisfies

1. (Exchange) If $\Delta; v : A, v : B; \cdot \vdash C \mid s$ then If $\Delta; v : B, v : A; \cdot \vdash C \mid s$
2. (Exchange) If $\Delta; a : A; b : B; \Gamma \vdash C \mid\mid s$ then If $\Delta; b : B; a : A; \Gamma \vdash C \mid\mid s$
3. (Weakening) If $\Delta; \cdot \vdash A \mid s$ then $\Delta, a : B \vdash A \mid s$
4. (Weakening) If $\Delta; \Gamma \vdash A \mid\mid s$ then $\Delta, v : B, \Gamma \vdash A \mid\mid s$
5. (Contraction) If $\Delta; u : A, v : A; \Delta', \cdot \vdash A \mid s$ then $\Delta; w : A; \Delta', \cdot \vdash A^{u,v}_{w} \mid s^{u,v}_{w}$ for w fresh
6. (Contraction) If $\Delta; \Gamma, a : A, b : A \vdash A \mid\mid s$ then $\Delta; \Gamma, c : A \vdash A^{a,b}_{c} \mid\mid s^{a,b}_{c}$ for c fresh.

Equality

Definition (Axiom and Inference Schemes for Identity with Unconditional and Dependent Evidence)

It is proven that the following hold:

- substitution on terms
- context equivalence
- reflexivity on unconditional and dependent evidence
- symmetry on unconditional and dependent evidence
- transitivity on unconditional and dependent evidence
- equivalence on λ-terms and application for implication
- equivalence on β/η redexes for $\Box$, $\Diamond$
- equivalence on Introduction/Elimination Rules for $\Box$, $\Diamond$
- equivalence on Functional Terms and Application

Contractions/Expansions

Transformation of derivations by contractions and expansions on connectives, adding appropriate operations for dependent evidence. (Connectives behave well with intro/elimination).
Contraction for $\supset$

$$\dfrac{\dfrac{\overline{\Delta; v:A; \cdot \vdash B \mid s}}{\Delta; \Gamma \vdash A \supset B \mid \lambda v:A.s} \supset I \qquad \Delta; \cdot \vdash A \mid t}{\Delta; \Gamma \vdash B \mid (\lambda v:A.s)\cdot t} \supset E$$

contracts to

$$\dfrac{\dfrac{\pi}{\Delta; \Gamma \vdash B \mid s^a_t} \qquad \dfrac{\Delta; v:A \vdash B \mid s \qquad \Delta, \cdot \vdash A \mid t}{\Delta, \Gamma \vdash s^v_t \equiv (\lambda v:A.s)\cdot t : B} Eq\beta}{\Delta; \Gamma \vdash B \mid (\lambda v:A.s)\cdot t} \text{EqUnEv}$$

where π is a derivation according to the Term Substituiton Theorem with Unconditional Evidence.

Contractions/Expansions

Contraction for Function

$$\dfrac{\dfrac{\Delta; a{:}A \vdash B \mid\mid t}{\Delta; \cdot \vdash \ll s \gg [\![t]\!]B[A]}\ \textit{Function Formation} \qquad \dfrac{\Delta; \cdot \vdash A \mid s}{\Delta; \Gamma \vdash [\![s]\!]A \mid !s}\ \Box I}{\Delta; \Gamma \vdash B \mid !([\![s]\!] \cdot [\![t]\!])}\ \textit{Appl}$$

contracts to

$$\dfrac{\dfrac{\dfrac{\Delta; \Gamma; \cdot \vdash \ll s \gg A \mid ?s \qquad \Delta; a{:}A \vdash B \mid\mid t}{\Delta; \Gamma; \cdot \vdash B^{a}_{s} \mid\mid \textit{ASSM } s \textit{ AS } a{:}A \textit{ IN } t}\ \Diamond E}{\Delta; \Gamma \vdash (?a{:}A)(a \cdot t) \equiv \ll s \gg [\![t]\!] : B[A]}\ \text{eq}\eta \qquad \dfrac{\Delta; \cdot \vdash A \mid s}{\Delta, \Gamma \vdash [\![s]\!]A \mid !s}\ \Box I}{\Delta; \Gamma \vdash B \mid !([\![s]\!] \cdot [\![t]\!])}\ \text{Appl}$$

Contractions/Expansions

Expansion for $\supset$

$$\Delta;\Gamma \vdash A \supset B \mid s$$

expands to

$$\dfrac{\dfrac{\dfrac{\Delta;\cdot \vdash A \supset B \mid s \qquad \Delta;\cdot \vdash A \mid v}{\Delta; v{:}A;\Gamma \vdash B \mid s \cdot v}\supset E}{\Delta;\Gamma \vdash A \supset B \mid \lambda v{:}A.(s \cdot v)}\supset I \qquad \dfrac{\Delta;\Gamma \vdash A \supset B \mid s \qquad v \notin \mathtt{fv}(t)}{\Delta;\Gamma \vdash \lambda v{:}A.(s \cdot v) \equiv s : A \supset B}\mathrm{Eq}{\supset}\mathrm{I}}{\Delta;\Gamma \vdash A \supset B \mid s}\mathit{EqUnE}$$

Contractions/Expansions

Expansion for Function

$$\Delta; \cdot \vdash \ll s \gg [\![t]\!]B[A]$$

expands to

$$\dfrac{\dfrac{\dfrac{\dfrac{\Delta; \Gamma \vdash \ll s \gg [\![t]\!]B[A] \qquad \Delta; \Gamma[\![s]\!]A \;|!s}{\Delta; \Gamma \vdash B \;|!([\![s]\!] \cdot [\![t]\!])}\,\text{Appl} \qquad \Delta; v : A \vdash B \mid t}{\Delta; \Gamma \vdash B^{s}_{[\![s]\!] \cdot [\![t]\!]} \mid \mathit{XTRT}\ s\ \mathit{AS}\ v : A\ \mathit{IN}\ t}\,\Box\text{E}}{\Delta; \Gamma; \cdot \vdash s^{v}_{t} \equiv s \cdot t : B}\,\text{Eq}\beta \qquad \Delta; \Gamma \vdash A \;||\; s}{\Delta; \cdot \vdash \ll s \gg [\![t]\!]B[A]}\,\mathit{EqDepEv}$$

1 Motivation

2 Intuitionistic JL with Dependency

3 Natural Deduction with Global and Local Assumptions

4 Normalization

5 Summary

Normal Forms

Definition (Predicates *INF* and *FNF*)

The normal form predicates *INF* and *FNF* are defined according to the following schemas:

$$\frac{\Delta; \cdot \vdash A \mid s}{\Delta; \Gamma \vdash FNF(s)} \qquad \frac{\Gamma \vdash A \mid\mid s}{\Gamma; \cdot \vdash INF(s)}$$

$$\frac{\Delta; \cdot \vdash FNF(A) \qquad \Delta; a{:}A \vdash FNF(t)}{\Delta; \Gamma; \cdot \vdash FNF([a/v] \cdot t)}$$

$$\frac{\Gamma; \cdot \vdash INF(A) \qquad \Delta; a{:}A \vdash FNF(t)}{\Delta; \Gamma; \cdot \vdash INF(t[a{:}A])}$$

Rewriting Rules

Definition (Rewriting Rules for *INF* / *FNF* predicates)

The η-expansion rewriting rules for *INF* / *FNF* predicates are defined according to the following rules:

$$\frac{\Delta;\Gamma \vdash A \supset B \mid s \cdot t \qquad \Delta;\Gamma \vdash FNF(s \cdot t)}{\Delta;\Gamma \vdash s \rightarrow_{\eta FNF} v : A.s}$$

$$\frac{\Delta;\cdot \vdash t \rightarrow_{\eta FNF} t'}{\Delta;\Gamma \vdash t \rightarrow_{\eta INF} t'} \qquad \frac{\Delta; v : A \vdash t \rightarrow_{\eta FNF} t'}{\Delta;\Gamma, \vdash t[a : A] \rightarrow_{\eta INF} t'[a : A]}$$

$$\frac{\Delta;\Gamma \vdash A \rightarrow_{\eta INF} A'}{\Delta;\Gamma \vdash B[A] \rightarrow_{\eta INF} B[A']} \qquad \frac{\Delta;\cdot \vdash A \rightarrow_{\eta FNF} A'}{\Delta;\Gamma \vdash A[B] \rightarrow_{\eta INF} A'[B]}$$

$$\frac{\Delta;\cdot \vdash A \mid s \qquad \Delta; v : A \vdash B \mid t \rightarrow_{\eta FNF} B' \mid t'}{\Delta;\Gamma \vdash B^v_t \mid XTRT\ s\ AS\ v : A\ IN\ t \rightarrow_{\eta FNF} B^v_t \mid XTRT\ s\ AS\ v : A\ IN\ t'}$$

$$\frac{\Delta;\Gamma \vdash A \mid\mid s \qquad \Delta; a : A;\Gamma \vdash B \mid t \rightarrow_{\eta FNF} B' \mid t'}{\Delta;\Gamma \vdash B^a_t \mid ASSM\ s\ AS\ a : A\ IN\ t \rightarrow_{\eta INF} B^a_t \mid ASSM\ s\ AS\ a : A\ IN\ t'}$$

Normal Forms

More intermediate steps are required:

- every INF/FNF reduction is either a β or a η reduction;
- every INF/FNF ends in a β normal redux;
- equivalence is a beta reduction.

Lemma (Normalisation)

If $\Delta; \Gamma \vdash FNF(t)$, then t is in normal form.

Normal Forms

Lemma (Confluence)

1. *$\rightarrow_{INF\,/\,FNF}$-normal forms are unique,*
2. *confluence corresponds to saying that every term reduces to a normal formal*
3. *Since every term has a unique β-normal formal*
4. *and by reductions $\rightarrow_{\eta INF\,/\,FNF}$ preserve β-normal forms,*
5. *normalisation of $\rightarrow_{INF\,/\,FNF}$ is reduced to normalisation of $\rightarrow_{\eta INF\,/\,FNF}$.*

Normal Forms

Lemma (Confluence)

1. $\rightarrow_{INF\,/\,FNF}$*-normal forms are unique,*
2. *confluence corresponds to saying that every term reduces to a normal formal*
3. *Since every term has a unique* β*-normal formal*
4. *and by reductions* $\rightarrow_{\eta INF\,/\,FNF}$ *preserve* β*-normal forms,*
5. *normalisation of* $\rightarrow_{INF\,/\,FNF}$ *is reduced to normalisation of* $\rightarrow_{\eta INF\,/\,FNF}$.

Lemma

By induction, $\rightarrow_{\eta INF\,/\,FNF}$ *for expressions with dependent evident requires at most a finite number of function application rule instances to reduce to an unconditional evidence.*

Strong Normalization

Lemma (Strong Normalization)

There are no infinite sequences of reductions
$\Delta; \Gamma \vdash t \rightarrow_{\eta INF/FNF} t' \rightarrow_{\eta INF/FNF} t'' \ldots$

1 Motivation

2 Intuitionistic JL with Dependency

3 Natural Deduction with Global and Local Assumptions

4 Normalization

5 Summary

Summary

- Introduced functional expressions over evidences;
- Used it to define a natural deduction calculus which distinguishes between unconditional and dependent evidence;
- Extended it to extensional equivalence;
- Proven that this extension is conservative w.r.t. the calculus with simple evidence from [Artëmov and Bonelli, 2007] by showing (Strong) Normalization.

References I

Alt, J. and Artemov, S. (2001).
Reflective λ-calculus.
In Kahle, R., Schroeder-Heister, P., and Stärk, R. F., editors, *Proof Theory in Computer Science*, volume 2183 of *Lecture Notes in Computer Science*, pages 22–37. Springer.

Artëmov, S. N. and Bonelli, E. (2007).
The intensional lambda calculus.
In Artëmov, S. N. and Nerode, A., editors, *LFCS*, volume 4514 of *Lecture Notes in Computer Science*, pages 12–25. Springer.

Pfenning, F. (2004).
Automated theorem proving.
Technical report, Carnegie Mellon.
Handouts for Course on Automated Theorem Proving.