A Rewrite System for Strongly Normalizable Terms

IRIF Seminar

Olivier Hermant & Ronan Saillard

CRI, MINES ParisTech, PSL University

June 28, 2018

Intersection Types

$$\frac{(x : F) \in \Gamma}{\Gamma \vdash x : F}\ (\text{Axiom})$$

$$\frac{\Gamma \vdash X : F \to G \qquad \Gamma \vdash Y : F}{\Gamma \vdash X\ Y : G}\ (\to_E) \qquad \frac{\Gamma, x : F \vdash X : G}{\Gamma \vdash \lambda x.X : F \to G}\ (\to_I)$$

$$\frac{\Gamma \vdash X : F \cap G}{\Gamma \vdash X : F}\ (\cap_{E1}) \quad \frac{\Gamma \vdash X : F \cap G}{\Gamma \vdash X : G}\ (\cap_{E2}) \quad \frac{\Gamma \vdash X : F \qquad \Gamma \vdash X : G}{\Gamma \vdash X : F \cap G}\ (\cap$$

- $\lambda x.(x\ x) : (A \cap (A \to A)) \to A$
- if X is typable, it is SN
- if X is SN, it is typable

Statman's System [2012]

- higher-order predicate/new connective $D : \iota \to o \to o \to o$, a discriminator

$$D\ v\ F\ G$$

- properties : $D\ 0\ F\ G$ is F and $D\ 1\ F\ G$ is G
- intuition
  - ★ D is an "*if...then...else...*" operator
  - ★ $\forall v\ DvFG$ encodes $F \cap G$
- meaning given by rewrite rules

$$
\begin{array}{lll}
D0FG & \longrightarrow F & (0) \\
D1FG & \longrightarrow G & (1) \\
Dt(F \to G)(H \to K) & \longrightarrow (DtFH) \to (DtGK) & (\to) \\
Dt(\forall xF)G & \longrightarrow \forall x(DtFG) & (\forall_1^*) \\
DtF(\forall xG) & \longrightarrow \forall x(DtFG) & (\forall_2^*) \\
\forall xF & \longrightarrow F & (\$^\dagger) \\
\forall X \forall yF & \longrightarrow \forall y \forall xF & (\$\$)
\end{array}
$$

($*$) no variable is captured (including those of t)

($\dagger$) x does not appear in F

Typing Rules

$$\frac{(x : F) \in \Gamma}{\Gamma \vdash x : F}\ \text{(Axiom)} \qquad \frac{\Gamma \vdash X : F \qquad F \equiv G}{\Gamma \vdash X : G}\ \text{(Conv)}$$

$$\frac{\Gamma \vdash X : F \to G \qquad \Gamma \vdash Y : F}{\Gamma \vdash X\ Y : G}\ (\to_E) \qquad \frac{\Gamma, x : F \vdash X : G}{\Gamma \vdash \lambda x.X : F \to G}\ (\to_I)$$

$$\frac{\Gamma \vdash X : F \qquad v : \iota \qquad v \notin fv(\Gamma)}{\Gamma \vdash X : \forall v.F}\ (\forall_I) \qquad \frac{\Gamma \vdash X : \forall v.F \qquad t : \iota \qquad t \text{ free for } v \text{ in } F}{\Gamma \vdash X : F[v/t]}\ (\forall_E)$$

Figure – Typing Rules of Minimal Natural Deduction with Conversion

- if X is typable, it is SN
- if X is SN, it is typable

Example

- Give a type to $\lambda x.(x\ x)$

Goals

1. down to first-order intuitionistic logic
   - with rewrite rules
   - suitable framework : Deduction Modulo Theory
2. prove
   - if X is typable, it is SN
   - if X is SN, it is typable
3. (further work) the superconsistency conjecture
   - (G. Dowek) : a type system is superconsistent iff it is SN
   - ($\Rightarrow$) : proof by Dowek
   - ($\Leftarrow$) : ?
   - this system is a candidate to disprove the conjecture

Method

- a type system with conversion is totally acceptable
- conversion is now defined otherwise
- remind Statman's system

$$
\begin{array}{llll}
D0FG & \longrightarrow F & & (0) \\
D1FG & \longrightarrow G & & (1) \\
Dt(F \rightarrow G)(H \rightarrow K) & \longrightarrow (DtFH) \rightarrow (DtGK) & & (\rightarrow) \\
Dt(\forall xF)G & \longrightarrow \forall x(DtFG) & & (\forall^*) \\
DtF(\forall xG) & \longrightarrow \forall x(DtFG) & & (\forall^*) \\
\forall xF & \longrightarrow F & & (\$^\dagger) \\
\forall X \forall yF & \longrightarrow \forall y \forall xF & & (\$\$)
\end{array}
$$

- we need to get rid of *D*
- at least at the *propositional* level
- what can we save from this system in Deduction Modulo Theory ?

Deduction Modulo Theory

Rewrite Rule

A term (resp. proposition) rewrite rule is a pair of terms (resp. formulæ) $l \longrightarrow r$, where $\mathcal{FV}(l) \subseteq \mathcal{FV}(r)$ and, in the propositiona case, l is atomic.

Examples :

- term rewrite rule :

$$A \cup \emptyset \longrightarrow A$$

- proposition rewrite rule :

$$A \subseteq B \longrightarrow \forall x\; x \in A \Rightarrow x \in B$$

Conversion modulo a Rewrite System

We consider the congruence $\equiv$ generated by a set of proposition rewrite rules $\mathcal{R}$ and a set of term rewrite rules $\mathcal{E}$ (often implicit)

Example :

$$A \cup \emptyset \subseteq A \quad \equiv \quad \forall x\; x \in A \Rightarrow x \in A$$

Typing Rules

$$\frac{(x : F) \in \Gamma}{\Gamma \vdash x : F} \text{ (Axiom)} \qquad \frac{\Gamma \vdash X : F \qquad F \equiv G}{\Gamma \vdash X : G} \text{ (Conv)}$$

$$\frac{\Gamma \vdash X : F \to G \qquad \Gamma \vdash Y : F}{\Gamma \vdash X\ Y : G} \ (\to_E) \qquad \frac{\Gamma, x : F \vdash X : G}{\Gamma \vdash \lambda x.X : F \to G} \ (\to_I)$$

$$\frac{\Gamma \vdash X : F \qquad v : \iota \qquad v \notin fv(\Gamma)}{\Gamma \vdash X : \forall v.F} \ (\forall_I) \qquad \frac{\Gamma \vdash X : \forall v.F \qquad t : \iota \qquad t \text{ free for } v \text{ in } F}{\Gamma \vdash X : F[v/t]} \ (\forall_E)$$

FIGURE – Typing Rules of Minimal Natural Deduction Modulo Theory

Proof of $A \subseteq A$ with and without DM

- without ($\Gamma := z : A \subseteq A \Leftrightarrow \forall x(x \in A \Rightarrow x \in A)$) :

$$\dfrac{\wedge_{E2}\dfrac{(\text{ax})\dfrac{}{\Gamma \vdash z : A \subseteq A \Leftrightarrow \forall x(x \in A \Rightarrow x \in A)}}{\Gamma \vdash z_2 : \forall x(x \in A \Rightarrow x \in A) \Rightarrow A \subseteq A} \qquad \dfrac{\vdots}{\Gamma \vdash \lambda y.y : \forall x(x \in A \Rightarrow x \in A) \Rightarrow A \subseteq A}}{\Gamma \vdash (z_2\ (\lambda y.y)) : A \subseteq A}$$

- with

$$\dfrac{\dfrac{\dfrac{(\text{ax})\dfrac{}{y : x \in A \vdash y : x \in A}}{\vdash \lambda y.y : x \in A \Rightarrow x \in A}}{\vdash \lambda y.y : \forall x(x \in A \Rightarrow x \in A)}\forall_I}{\vdash \lambda y.y : A \subseteq A}(\text{Conv})$$

- "as if" we replaced z_1 and z_2 with $\lambda\alpha.\alpha$ (see also Polarized Deduction Modulo Theory)

Statman's System in First-Order Minimal DMT

- Analysis

$$
\begin{array}{lll}
D0FG & \longrightarrow F & (0) \\
D1FG & \longrightarrow G & (1) \\
Dt(F \to G)(H \to K) & \longrightarrow (DtFH) \to (DtGK) & (\to) \\
Dt(\forall xF)G & \longrightarrow \forall x(DtFG) & (\forall^*) \\
DtF(\forall xG) & \longrightarrow \forall x(DtFG) & (\forall^*) \\
\forall xF & \longrightarrow F & (\$^\dagger) \\
\forall X \forall yF & \longrightarrow \forall y \forall xF & (\$\$)
\end{array}
$$

Statman's System in First-Order Minimal DMT

- Analysis

$$
\begin{array}{lll}
D0FG & \longrightarrow F & (0) \\
D1FG & \longrightarrow G & (1) \\
Dt(F \to G)(H \to K) & \longrightarrow (DtFH) \to (DtGK) & (\to) \\
Dt(\forall xF)G & \longrightarrow \forall x(DtFG) & (\forall^*) \\
DtF(\forall xG) & \longrightarrow \forall x(DtFG) & (\forall^*) \\
\forall xF & \longrightarrow F & (\$^\dagger) \\
\forall X \forall yF & \longrightarrow \forall y \forall xF & (\$\$)
\end{array}
$$

- get rid of D
  - reflect it as a term (techniques to embed HOL)
  - introduce the following terms : $\dot{\forall}, \dot{\Rightarrow}, D, 0, 1$
  - and a unique unary predicate ε

Statman's System in First-Order Minimal DMT

- Analysis

$$\begin{array}{lll} D0FG & \longrightarrow F & (0) \\ D1FG & \longrightarrow G & (1) \\ Dt(F \to G)(H \to K) & \longrightarrow (DtFH) \to (DtGK) & (\to) \\ Dt(\forall xF)G & \longrightarrow \forall x(DtFG) & (\forall^*) \\ DtF(\forall xG) & \longrightarrow \forall x(DtFG) & (\forall^*) \\ \forall xF & \longrightarrow F & (\$^\dagger) \\ \forall X \forall yF & \longrightarrow \forall y \forall xF & (\$\$) \end{array}$$

- get rid of D
  - reflect it as a term (techniques to embed HOL)
  - introduce the following terms : $\dot\forall, \dot\Rightarrow, D, 0, 1$
  - and a unique unary predicate ε
- combine terms properly (e.g. forbid $\dot\forall\dot\Rightarrow$)
  - simple types, with ι (0 and 1) and o (propositional terms)

$$\begin{array}{llll} 0,1 & : \iota & D & : \iota \to o \to o \to o \\ \dot\forall & : (\iota \to o) \to o & \dot\Rightarrow & : o \to o \to o \end{array}$$

  - add one propositional symbol $\dot p : o$
- propositional rewriting :

$$\varepsilon(A \dot\Rightarrow B) \longrightarrow \varepsilon(A) \Rightarrow \varepsilon(B)$$

$$\varepsilon(\dot\forall A) \longrightarrow \forall x.\varepsilon(A\ x)$$

Statman's System in First-Order Minimalist Deduction Modulo Theory

- Statman's System :

$$
\begin{array}{lll}
D0FG & \longrightarrow F & (0)\\
D1FG & \longrightarrow G & (1)\\
Dt(F \to G)(H \to K) & \longrightarrow (DtFH) \to (DtGK) & (\to)\\
Dt(\forall xF)G & \longrightarrow \forall x(DtFG) & (\forall^*)\\
DtF(\forall xG) & \longrightarrow \forall x(DtFG) & (\forall^*)\\
\forall xF & \longrightarrow F & (\$^\dagger)\\
\forall X \forall yF & \longrightarrow \forall y \forall xF & (\$\$)
\end{array}
$$

- we can readily define three rewrite rules :

$$
\begin{array}{lll}
D0FG & \longrightarrow F & (0)\\
D1FG & \longrightarrow G & (1)\\
Dt(F \dot{\Rightarrow} G)(H \dot{\Rightarrow} K) & \longrightarrow (DtFH) \dot{\Rightarrow} (DtGK) & (\dot{\Rightarrow})
\end{array}
$$

- still to be defined

$$
\begin{array}{lll}
Dt(\dot{\forall} F)G \longrightarrow \dot{\forall}? & (\forall_1)\\
DtF(\dot{\forall} G) \longrightarrow \dot{\forall}? & (\forall_2)
\end{array}
$$

How to Abstract

Need an equivalent of $Dt(\forall x F)G \longrightarrow \forall x(DtFG)$

- at term level $Dv(\dot{\forall}F)G$
- no "free variable x"
  - ★ hence no "freshness constraint" (good)
  - ★ nevertheless need to define something like

$$Dv(\dot{\forall}F)G \longrightarrow \dot{\forall}(\lambda x.(Dv(Fx)G))$$

    for some fresh x
- two solutions exist in Deduction modulo theory
  1. allow λ-abstraction in the simply-typed term language
  2. replace this by a combinatorial calculus
- choice :
  - ★ Solution 1 cumbersome : explicit substitutions interfere with D
  - ★ Solution 2 cumbersome too
  - ★ could we have dropped explicit substitutions ?

Combinatorial Calculus SKI

- we introduce

$$\begin{aligned} I &: \iota \to \iota \\ K &: \tau \to \iota \to \tau \\ S &: (\iota \to \tau \to \alpha) \to (\iota \to \tau) \to \iota \to \alpha \end{aligned}$$

- and application symbols :
  - ★ denote in those slides as white space
- usual reduction rules

$$\begin{array}{lll} I\ x & \longrightarrow x & (I) \\ K\ x\ y & \longrightarrow x & (K) \\ S\ x\ y\ z & \longrightarrow x\ z\ (y\ z) & (S) \end{array}$$

Combinatorial Calculus SKI

- defining abstraction
  - it *is* possible (see textbooks)
  - we only need
    $$Dv(\dot\forall F)G \longrightarrow \dot\forall(\lambda x.(Dv(Fx)G))$$
    with x fresh.
  - so, define $\lambda x.(Dv(Fx)G)$ as
    $$S\ \ (K\ DvF)\ \ (S\ (KG)\ I)$$
  - similarly for
    $$DvF(\dot\forall G) \longrightarrow \dot\forall(\lambda x.(DvF(Gx)))$$
  - note :
    - no new (rewriting) redex is created
    - some redex might be destroyed (take 0 or 1 for v)

The Final Rewriting System

- encoding the logic :

$$\begin{array}{llll} \varepsilon(A \dot{\Rightarrow} B) \longrightarrow \varepsilon(A) \Rightarrow \varepsilon(B) & (\varepsilon_{\Rightarrow}) & I\ x \longrightarrow x & (I) \\ \varepsilon(\dot{\forall} A) \longrightarrow \forall x.\varepsilon(A\ x) & (\varepsilon_{\forall}) & K\ x\ y \longrightarrow x & (K) \\ & & S\ x\ y\ z \longrightarrow x\ z\ (yz) & (S) \end{array}$$

- encoding Statman's rules :

$$\begin{array}{lll} D0FG & \longrightarrow F & (0) \\ D1FG & \longrightarrow G & (1) \\ Dt(F \dot{\Rightarrow} G)(H \dot{\Rightarrow} K) & \longrightarrow (DtFH) \dot{\Rightarrow} (DtGK) & (\dot{\Rightarrow}) \\ Dv(\dot{\forall} F)G & \longrightarrow \dot{\forall}(\lambda x.(Dv(Fx)G)) & (\forall_1) \\ DvF(\dot{\forall} G) & \longrightarrow \dot{\forall}(\lambda x.(DvF(Gx))) & (\forall_2) \end{array}$$

- no way to encode
  - ★ ($) - pruning unnecessary quantifiers,
  - ★ ($$) - permuting quantifiers
  - ★ nonterminating rules
- but we need confluence ! 7 critical pairs :
  - ★ $Dv(\dot{\forall} F)(\dot{\forall} G)$ (needs ($$))
  - ★ $D0(\dot{\forall} F)G$ (reducing by $\forall_1$ "freezes" the (0))
  - ★ impossible for terms, weak confluence at the ε-level (sweat)

Termination First

- encoding the logic :

$$
\begin{array}{lll}
\varepsilon(A \dot{\Rightarrow} B) \longrightarrow \varepsilon(A) \Rightarrow \varepsilon(B) & (\varepsilon_{\Rightarrow}) \\
\varepsilon(\dot{\forall} A) \longrightarrow \forall x.\varepsilon(A\ x) & (\varepsilon_{\forall})
\end{array}
\qquad
\begin{array}{lll}
I\ x & \longrightarrow x & (I) \\
K\ x\ y & \longrightarrow x & (K) \\
S\ x\ y\ z & \longrightarrow x\ z\ (yz) & (S)
\end{array}
$$

- encoding Statman's rules :

$$
\begin{array}{lll}
D0FG & \longrightarrow F & (0) \\
D1FG & \longrightarrow G & (1) \\
Dt(F \dot{\Rightarrow} G)(H \dot{\Rightarrow} K) & \longrightarrow (DtFH) \dot{\Rightarrow} (DtGK) & (\dot{\Rightarrow}) \\
Dv(\dot{\forall} F)G & \longrightarrow \dot{\forall}(\lambda x.(Dv(Fx)G)) & (\forall_1) \\
DvF(\dot{\forall} G) & \longrightarrow \dot{\forall}(\lambda x.(DvF(Gx))) & (\forall_2)
\end{array}
$$

- termination
  1. ε reduces the number of $\dot{\forall}$, $\dot{\Rightarrow}$
  2. typed-restricted S and K imply no duplication of $\dot{\forall}$ and $\dot{\Rightarrow}$
  3. $\dot{\forall}$ goes up in terms
  4. $\dot{\Rightarrow}$ goes up and decreases
  5. simply-typed SKI terminates
- automatically prove termination ?

Confluence

- impossible at the term level

$$
\begin{array}{ccc}
 & D0(\dot\forall F)G & \\
\swarrow & & \searrow \\
\dot\forall(\lambda x.(D0(F\ x)G) & & \dot\forall F
\end{array}
$$

- fixed at the proposition level

$$
\begin{array}{ccc}
 & \varepsilon(D0(\dot\forall F)G) & \\
\swarrow & & \searrow \\
\varepsilon(\dot\forall(\lambda x.(D0(F\ x)G)) & & \varepsilon(\dot\forall F) \\
\searrow\!\!\searrow & & \swarrow \\
 & \forall x.\varepsilon(F\ x) &
\end{array}
$$

- still problematic for $D0F(\dot\forall G)$:

$$\forall y \varepsilon(F)\ {}^*\!\!\longleftarrow \quad \longrightarrow^* \ \varepsilon(F)$$

- and $Dv(\dot\forall F)(\dot\forall G)$:

$$\forall x \forall y \varepsilon(Dv(Fx)(Gy))\ {}^*\!\!\longleftarrow \quad \longrightarrow^* \ \forall y \forall x \varepsilon(Dv(Fx)(Gy))$$

Confluence up to equivalence

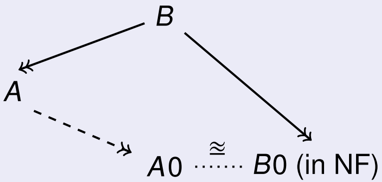
- but we can have confluence up to : variable renaming, pruning and inversion of quantifiers
- does not fit in a term equational theory $\mathcal{E}$

Definition

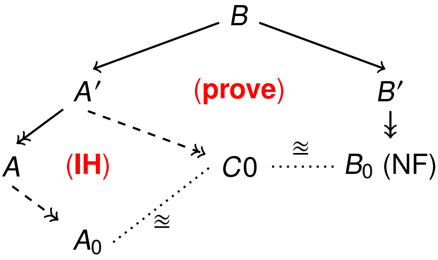
$A \approxeq B$:

- if A and B normal, and
- either $A = \forall\vec{x}.\varepsilon(t_A), B = \forall\vec{y}.\varepsilon(t_B)$ and $\varepsilon(t_A) \equiv \varepsilon(t_B)$
- or $A = \forall\vec{x}.(A_1 \to A_2), B = \forall\vec{y}.(B_1 \to B_2)$ and $A_1 \equiv B_1$ and $A_2 \equiv B_2$.

Lemma



Proof of the Lemma : Strategy



- get rid of no rewrite step from B to A or to $B0$
- prove the existence of C_0
- induction on : the height of the reduction tree of B, noted $|B|$
- easy if $A' \longleftarrow B \longrightarrow B'$ does not involve a critical pair
- let us see the $\dot{\forall} F \longleftarrow D0(\dot{\forall} F)G \longrightarrow \dot{\forall} \lambda x.(D0(F\ x)G)$ case

One Critical Pair

- we have

$$
\begin{array}{ccc}
 & B = \mathcal{K}[D0(\dot{\forall}F)G] & \\
\swarrow & & \searrow \\
A' = \mathcal{K}[\dot{\forall}F] & & B' = \mathcal{K}[\dot{\forall}\lambda x.(D0(F\ x)G)] \\
 & & \downarrow\!\!\!\!\downarrow \\
 & & B_0\ (\text{NF})
\end{array}
$$

- long time to wait before joining at the ε level
- introduce the inductive invariant

$t_1 \sim t_2$ if there exist a context $\mathcal{K}$ and two terms θ_1, θ_2 such that :

- $A' \longrightarrow^* t_1 = \mathcal{K}[\dot{\forall}\theta_1]$,
- $B' \longrightarrow^* t_2 = \mathcal{K}[\dot{\forall}\theta_2] \longrightarrow^* B_0$,
- $P(\theta_1, \theta_2)$ where $P(u_1, u_2)$ is :
  - $u_2 x \longrightarrow^* {}^*\!\!\longleftarrow u_1 x$ and
  - if $u_2 \longrightarrow u_2'$ then for some $u_1 \longrightarrow^* u_1'$, $P(u_1', u_2')$

- $P(\dot{\forall}F, \dot{\forall}\lambda x.(D0(F\ x)G))$

Critical Pair : Interesting Subcases

$t_1 \sim t_2$ if there exist a context $\mathcal{K}$ and two terms θ_1, θ_2 such that :

- $A' \longrightarrow^* t_1 = \mathcal{K}[\dot{\forall}\theta_1]$,
- $B' \longrightarrow^* t_2 = \mathcal{K}[\dot{\forall}\theta_2] \longrightarrow^* B_0$,
- $P(\theta_1, \theta_2)$ where $P(u_1, u_2)$ is :
  - $u_2 x \longrightarrow^* {}^*\!\longleftarrow u_1 x$ and
  - if $u_2 \longrightarrow u_2'$ then for some $u_1 \longrightarrow^* u_1'$, $P(u_1', u_2')$

- proof the existence of C_0 by induction on $t_2 \longrightarrow^* B_0$
- $t_2 = \mathcal{L}[\varepsilon(\dot{\forall}\theta_2)] \longrightarrow \mathcal{L}[\forall x.\varepsilon(\theta_2 x)]$
  - Confluence case. Saved ! (big IH as $|t_2| < |B|$)
- $t_2 = \mathcal{L}[Dv(\dot{\forall}\theta_2)Z] \longrightarrow \mathcal{L}[\dot{\forall}\lambda^x(Dv(\theta_2 x)Z)]$
  - IH, since $P(\lambda x.(Dv(\theta_1 x)Z), \lambda x.(Dv(\theta_2 x)Z))$
- $t_2 = \mathcal{K}[\dot{\forall}\theta_2] \longrightarrow \mathcal{K}'[\dot{\forall}\theta_2]$: fits
- $t_2 = \mathcal{K}[\dot{\forall}\theta_2] \longrightarrow \mathcal{K}[\dot{\forall}\theta_2']$: fits

Digging Terms

- confluence up to $\approxeq$ at the proposition level !
- with intersection types, merging derivations :

$$\frac{\Gamma \vdash X : F \qquad \Gamma \vdash X : G}{\Gamma \vdash X : F \cap G} (\cap_I)$$

- derivation transformations in Statman's system

Derivation Merge (Statman)

If $\Gamma_1 \vdash X : F$ and $\Gamma_2 \vdash X : G$ then $Dv\Gamma_1\Gamma_2 \vdash X : DvFG$

- we must
  - ★ prove the result
  - ★ perform this on terms (D is a term)

Proposition Reification

$$\begin{array}{lcl} \gamma(\varepsilon(t)) & := & t \\ \gamma(F \to G) & := & \gamma(F) \dot{\Rightarrow} \gamma(G) \\ \gamma(\forall x.F) & := & \dot{\forall}(\lambda x.(\gamma(F))) \end{array}$$

Reification

Proposition Reification

$$\begin{array}{lcl} \gamma(\varepsilon(t)) & := & t \\ \gamma(F \Rightarrow G) & := & \gamma(F)\dot{\Rightarrow}\gamma(G) \\ \gamma(\forall x.F) & := & \dot{\forall}(\lambda x.(\gamma(F))) \end{array}$$

$\gamma(F)$ noted $\dot{F}$.

- $\varepsilon(\gamma(F)) \longrightarrow^* F$
- we need (for conversions)

$$\text{if } F \equiv F', G \equiv G' \text{ then } Dv\dot{F}\dot{G} \equiv Dv\dot{F}'\dot{G}'$$

- problem : not preserved by γ ! Counter-example :
  - $F = \forall x.\varepsilon(D0AB)$ and $F' = \forall x.\varepsilon(A)$
  - $\dot{F}$ is not convertible with $\dot{F}'$
  - When F contains quantifiers : redexes frozen by λ

Reification

Proposition Reification

$$\begin{array}{lcl}\gamma(\varepsilon(t)) & := & t \\ \gamma(F \Rightarrow G) & := & \gamma(F)\dot{\Rightarrow}\gamma(G) \\ \gamma(\forall x.F) & := & \dot{\forall}(\lambda x.(\gamma(F)))\end{array}$$

$\gamma(F)$ noted $\dot{F}$.

We actually need

Merges are Convertible

if $F \equiv F', G \equiv G'$ then $\varepsilon(Dv\dot{F}\dot{G}) \equiv \varepsilon(Dv\dot{F}'\dot{G}')$

- Works separately for $F_1 \equiv G_1$ and $F_2 \equiv G_2$, but not for deeper combinations :

$$\varepsilon(Dv(\dot{F}_1 \dot{\Rightarrow} \dot{F}_2)\dot{p}) \text{ not convertible with } \varepsilon(Dv(\dot{G}_1 \dot{\Rightarrow} \dot{G}_2)\dot{p})$$

- $\dot{p}$ is not implicational : ε cannot expose the structure

Digging

Proposition Reification

$$\begin{array}{lcl} \gamma(\varepsilon(t)) & := & t \\ \gamma(F \Rightarrow G) & := & \gamma(F) \dot{\Rightarrow} \gamma(G) \\ \gamma(\forall x.F) & := & \dot{\forall}(\lambda x.(\gamma(F))) \end{array}$$

$\gamma(F)$ noted $\dot{F}$.

- $\varepsilon(Dv(\dot{F}_1 \dot{\Rightarrow} \dot{F}_2)\dot{p})$ not convertible with $\varepsilon(Dv(\dot{G}_1 \dot{\Rightarrow} \dot{G}_2)\dot{p})$
- idea : dig out the implicational structure
  - replace all $\dot{p}$ with $\dot{p} \dot{\Rightarrow} \dot{p}$
  - potentially, n times
  - notation $t\{n\}$

Merge Conversion

If $F_1 \equiv F_2$ and $G_1 \equiv G_2$ then, for some n,
$\varepsilon(Dv\dot{F}_1\{n\}\dot{G}_1\{n\}) \equiv \varepsilon(Dv\dot{F}_2\{n\}\dot{G}_2\{n\})$.

All SN terms are Typable

- finally able to follow Statman's line
- derivations are organized in segments

$$\frac{\Gamma \vdash X : F}{\Gamma \vdash X : G} (\forall_I),(\text{Conv}),(\forall_E)$$

- that we can organize as

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash X : F}{\Gamma \vdash X : G}(\forall_E)}{\Gamma \vdash X : G'}(\text{Conv})}{\Gamma \vdash X : H}(\forall_I)$$

- we can merge two segments :

$$\frac{\Gamma_1 \vdash X : F'}{\Gamma_1 \vdash X : F} (\text{seg}) \qquad \frac{\Gamma_2 \vdash X : G'}{\Gamma_2 \vdash X : G} (\text{seg})$$

  - ★ with some digging
  - ★ into $\Gamma \vdash X : \varepsilon(D v \dot{F}\{n\} \dot{G}\{n\})$

- we can also merge two derivations of the same term X

All SN terms are Typable, and conversely

- SN $\Rightarrow$ typable :
  1. follow more or less Statman's way (with more explanations)
  2. All terms in NF are typable
  3. if a reduct of a head β-redex is typable, so is the redex
  4. if X is SN, it is typable
- typable $\Rightarrow$ SN
  - find the "worse reduction strategy", $F_\infty(M)$ (Barendregt)
  - prove that it terminates
  - or use Reducibility Candidates ? (cf. plain intersection types)

Conclusion

- we have intersection types in Deduction Modulo Theory
- no new connectives, etc
- rewrite rules instead
  - interesting confluence property
  - regain “nice” properties and some extensionality : lot of plumbing
- further work
  - do we have a model with values on the reducibility candidates ?
  - one technical detail to fix (variable renaming)