Normalisation for general constructive propositional logic

Herman Geuvers
(joint work with Tonny Hurkens and Iris van der Giessen)

Institute for Computing and Information Science
Radboud University
Nijmegen, NL

Types 2018
University of Minho
June 20, 2018

Constructive Logic

Constructively the meaning of a connective is fixed by explaining what a proof is that involves the connective.
Basically, this explains the introduction rule(s) for each connective, from which the elimination rules follow (Prawitz)
By analysing constructive proofs we then also get

- consistency (from proof normalization),
- decidability (from the subformula property),
- Curry-Howard proofs-as-terms (and propositions-as-types) allowing to study normalization as term-reduction

Our work / Overview of the talk

- Derive constructive natural deduction rules for a connective from its truth table definition.
  - Gives natural deduction rules for a connective “in isolation”
  - Also gives constructive rules for connectives that haven’t been studied constructively so far, like nand and if-then-else.
- Curry-Howard: give proof-terms for natural deductions
- Define proof-normalization as term-reduction
  $\Rightarrow$ A general notion of detour conversion (cut-elimination) and permutation conversion for these constructive connectives.
- Weak normalization
- Strong normalization

Standard form for natural deduction rules

$$\frac{\Gamma \vdash A_1 \quad \ldots \quad \Gamma \vdash A_n \qquad \Gamma, B_1 \vdash D \quad \ldots \quad \Gamma, B_m \vdash D}{\Gamma \vdash D}$$

If the conclusion of a rule is $\Gamma \vdash D$, then the hypotheses of the rule can be of one of two forms:

1. $\Gamma, B \vdash D$. We are given extra data B to prove D from Γ. We call B a Case.
2. $\Gamma \vdash A$. instead of proving D from Γ, we now need to prove A from Γ. We call A a Lemma.

One obvious advantage: we don't have to give Γ explicitly, as it can be retrieved:

$$\frac{\vdash A_1 \quad \ldots \quad \vdash A_n \qquad B_1 \vdash D \quad \ldots \quad B_m \vdash D}{\vdash D}$$

Constructive natural deduction rules from truth tables

Let c be an n-ary connective c with truth table t_c.
Each row of t_c gives rise to an elimination rule or an introduction rule for c. (We write $\varphi = c(A_1, \ldots, A_n)$.)

elimination

A_1	$\ldots$	A_n	φ
p_1	$\ldots$	p_n	0

$$\mapsto \quad \frac{\vdash \varphi \ldots \vdash A_j \text{ (if } p_j = 1) \ldots A_i \vdash D \text{ (if } p_i = 0) \ldots}{\vdash D}\,\text{el}$$

introduction

A_1	$\ldots$	A_n	φ
q_1	$\ldots$	q_n	1

$$\mapsto \quad \frac{\ldots \vdash A_j \text{ (if } q_j = 1) \ldots A_i \vdash \varphi \text{ (if } q_i = 0) \ldots}{\vdash \varphi}\,\text{in}^i$$

This is the constructive introduction rule; there is also a classical introduction rule, which we don't discuss now.

Definition of the logics

Given a set of connectives $\mathcal{C} := \{c_1, \ldots, c_n\}$, the constructive natural deduction systems for $\mathcal{C}$, $\mathrm{IPC}_{\mathcal{C}}$, has the following rules.

- The axiom rule

$$\frac{}{\Gamma \vdash A}\,\text{axiom} \quad (\text{if } A \in \Gamma)$$

- The constructive introduction rules for the connectives in $\mathcal{C}$, as derived from the truth table.
- The elimination rules for the connectives in $\mathcal{C}$, as derived from the truth table.

Examples

Derivation rules for $\wedge$ (3 elim rules and one intro rule):

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

$$\frac{\vdash A \wedge B \quad A \vdash D \quad B \vdash D}{\vdash D}\ \wedge\text{-el}_a \qquad \frac{\vdash A \wedge B \quad A \vdash D \quad \vdash B}{\vdash D}\ \wedge\text{-el}_b$$

$$\frac{\vdash A \wedge B \quad \vdash A \quad B \vdash D}{\vdash D}\ \wedge\text{-el}_c \qquad \frac{\vdash A \quad \vdash B}{\vdash A \wedge B}\ \wedge\text{-in}$$

- These rules can be shown to be equivalent to the well-known derivation rules.
- These rules can be optimized to 3 rules.

Examples

Rules for $\neg$: 1 elimination rule and 1 introduction rule.

A	$\neg A$
0	1
1	0

Derivation rules:

$$\frac{\vdash \neg A \quad \vdash A}{\vdash D}\ \neg\text{-el} \qquad \frac{A \vdash \neg A}{\vdash \neg A}\ \neg\text{-in}$$

The well-known constructive connectives

The optimised rules for $\wedge, \neg, \vee, \rightarrow, \top$ and $\bot$ we obtain are:

$$\frac{\vdash A \qquad \vdash B}{\vdash A \wedge B}\ \wedge\text{-in} \qquad \frac{\vdash A \wedge B}{\vdash A}\ \wedge\text{-el}_1 \qquad \frac{\vdash A \wedge B}{\vdash B}\ \wedge\text{-el}_2$$

$$\frac{\vdash A \vee B \quad A \vdash D \quad B \vdash D}{\vdash D}\ \vee\text{-el} \qquad \frac{\vdash A}{\vdash A \vee B}\ \vee\text{-in}_1 \qquad \frac{\vdash B}{\vdash A \vee B}\ \vee\text{-in}_2$$

$$\frac{\vdash A \rightarrow B \quad \vdash A}{\vdash B} \rightarrow\text{-el} \qquad \frac{\vdash B}{\vdash A \rightarrow B} \rightarrow\text{-in}_1 \qquad \frac{A \vdash A \rightarrow B}{\vdash A \rightarrow B} \rightarrow\text{-in}_2$$

$$\frac{\vdash \neg A \quad \vdash A}{\vdash D}\ \neg\text{-el} \qquad \frac{A \vdash \neg A}{\vdash \neg A}\ \neg\text{-in} \qquad \frac{}{\vdash \top}\ \top\text{-in} \qquad \frac{\vdash \bot}{\vdash D}\ \bot\text{-el}$$

Sheffer stroke or NAND connective [I]

The truth table for nand(A, B), which we write as $A \uparrow B$ is as follows.

A	B	$A \uparrow B$
0	0	1
0	1	1
1	0	1
1	1	0

From this we derive the following optimized rules.

$$\frac{A \vdash A \uparrow B}{\vdash A \uparrow B}\uparrow\text{-inl} \qquad \frac{B \vdash A \uparrow B}{\vdash A \uparrow B}\uparrow\text{-inr} \qquad \frac{\vdash A \uparrow B \quad \vdash A \quad \vdash B}{\vdash D}\uparrow\text{-el}$$

Sheffer stroke or NAND connective [II]

The usual connectives can be defined in terms of nand.

$$\begin{array}{rcl} \dot{\neg} A & := & A \uparrow A \\ A \dot{\vee} B & := & (A \uparrow A) \uparrow (B \uparrow B) \\ A \dot{\wedge} B & := & (A \uparrow B) \uparrow (A \uparrow B) \\ A \dot{\to} B & := & A \uparrow (B \uparrow B) \end{array}$$

This gives rise to an embedding $(-)^{\uparrow}$ of intuitionistic proposition logic $\vdash_i$ into the nand-logic $\vdash_{\uparrow}$.

PROPOSITION For A a formula in proposition logic,

$$\vdash_i \neg\neg A \quad \Longleftrightarrow \quad \vdash_{\uparrow} (A)^{\uparrow}.$$

Detours (cuts) in constructive logic

Remember that the rules for c arise from rows in the truth table t_c:

A_1	$\dots$	A_n	$c(A_1,\dots,A_n)$
p_1	$\dots$	p_n	0
q_1	$\dots$	q_n	1

DEFINITION A detour convertibility is a pattern of the following form, where $\varphi = c(A_1,\dots,A_n)$.

$$\dfrac{\dfrac{\dots \quad \begin{array}{c}\boxed{\Sigma_j}\\ \Gamma \vdash A_j\end{array} \quad \dots \quad \begin{array}{c}\boxed{\Sigma_i}\\ \Gamma, A_i \vdash \varphi\end{array} \quad \dots}{\Gamma \vdash \varphi}\,\text{in} \qquad \dots \quad \begin{array}{c}\boxed{\Pi_k}\\ \Gamma \vdash A_k\end{array} \quad \dots \quad \begin{array}{c}\boxed{\Pi_\ell}\\ \Gamma, A_\ell \vdash D\end{array} \quad \dots}{\Gamma \vdash D}\,\text{el}$$

- $q_j = 1$ for A_j and $q_i = 0$ for A_i
- $p_k = 1$ for A_k and $p_\ell = 0$ for A_ℓ

Radboud University

Eliminating a detour (detour conversion) (I)

The *elimination of a detour* is defined by replacing the deduction pattern by another one. If $A_j = A_\ell$ (for some j, ℓ), replace

$$\dfrac{\dfrac{\ldots \quad \begin{array}{c}\boxed{\Sigma_j}\\ \Gamma \vdash A_j\end{array} \quad \ldots \quad \begin{array}{c}\boxed{\Sigma_i}\\ \Gamma, A_i \vdash \varphi\end{array} \quad \ldots}{\Gamma \vdash \varphi}\text{in} \qquad \ldots \quad \begin{array}{c}\boxed{\Pi_k}\\ \Gamma \vdash A_k\end{array} \quad \ldots \quad \begin{array}{c}\boxed{\Pi_\ell}\\ \Gamma, A_\ell \vdash D\end{array} \quad \ldots}{\Gamma \vdash D}\text{el}$$

by

$$\begin{array}{c}\begin{array}{c}\boxed{\Sigma_j}\\ \Gamma \vdash A_j\end{array} \quad \ldots \quad \begin{array}{c}\boxed{\Sigma_j}\\ \Gamma \vdash A_j\end{array}\\ \boxed{\Pi_\ell}\\ \Gamma \vdash D\end{array}$$

Eliminating a detour (detour conversion) (II)

If $A_i = A_k$ (for some i, k), replace

$$\dfrac{\dfrac{\ldots \quad \begin{array}{c}\boxed{\Sigma_j}\\ \Gamma \vdash A_j\end{array} \quad \ldots \quad \begin{array}{c}\boxed{\Sigma_i}\\ \Gamma, A_i \vdash \varphi\end{array} \quad \ldots}{\Gamma \vdash \varphi}\,\text{in} \quad \ldots \quad \begin{array}{c}\boxed{\Pi_k}\\ \Gamma \vdash A_k\end{array} \quad \ldots \quad \begin{array}{c}\boxed{\Pi_\ell}\\ \Gamma, A_\ell \vdash D\end{array} \quad \ldots}{\Gamma \vdash D}\,\text{el}$$

by

$$\dfrac{\begin{array}{c}\begin{array}{c}\boxed{\Pi_k}\\ \Gamma \vdash A_k\end{array} \quad \ldots \quad \begin{array}{c}\boxed{\Pi_k}\\ \Gamma \vdash A_k\end{array}\\ \boxed{\Sigma_i}\\ \Gamma \vdash \varphi\end{array} \quad \ldots \quad \begin{array}{c}\boxed{\Pi_k}\\ \Gamma \vdash A_k\end{array} \quad \ldots \quad \begin{array}{c}\boxed{\Pi_\ell}\\ \Gamma, A_\ell \vdash D\end{array} \quad \ldots}{\Gamma \vdash D}\,\text{el}$$

Observation

$$\dfrac{\dfrac{\ldots \quad \begin{array}{c}\boxed{\Sigma_j}\\ \Gamma \vdash A_j\end{array} \quad \ldots \quad \begin{array}{c}\boxed{\Sigma_i}\\ \Gamma, A_i \vdash \varphi\end{array} \quad \ldots}{\Gamma \vdash \varphi}\,\text{in} \quad \ldots \quad \begin{array}{c}\boxed{\Pi_k}\\ \Gamma \vdash A_k\end{array} \quad \ldots \quad \begin{array}{c}\boxed{\Pi_\ell}\\ \Gamma, A_\ell \vdash D\end{array} \quad \ldots}{\Gamma \vdash D}\,\text{el}$$

- There can be several "matching" (i,k) or (j,ℓ) pairs.
- So: detour conversion ("β-rule") is non-deterministic in general.

Permutation convertibility: Definition

Let c and c' be connectives of arity n and n', with elimination rules r and r' respectively. A permutation convertibility in a derivation is a pattern of the following form, where $\Phi = c(B_1, \ldots, B_n)$, $\Psi = c'(A_1, \ldots, A_{n'})$.

$$\dfrac{\dfrac{\vdash \Psi \ \ldots \ \overset{\boxed{\Sigma_j}}{\vdash A_j} \quad \ldots \quad \overset{\boxed{\Sigma_i}}{A_i \vdash \Phi} \quad \ldots}{\vdash \Phi}\,\mathrm{el}_{r'} \qquad \ldots \ \overset{\boxed{\Pi_k}}{\vdash B_k} \quad \ldots \quad \overset{\boxed{\Pi_\ell}}{B_\ell \vdash D} \quad \ldots}{\vdash D}\,\mathrm{el}_r$$

- A_j ranges over all propositions that have a 1 in the truth table of c'; A_i ranges over all propositions that have a 0,
- B_k ranges over all propositions that have a 1 in the truth table of c; B_ℓ ranges over all propositions that have a 0.

Permutation conversion

The permutation conversion is defined by replacing the derivation pattern on the previous slide by

$$\dfrac{\vdash \Psi \ldots \quad \begin{array}{c}\vdots\ \boxed{\Sigma_j}\\ \vdash A_j\end{array} \qquad \ldots \qquad \dfrac{\begin{array}{c}\vdots\ \boxed{\Sigma_i}\\ A_i \vdash \Phi\end{array} \quad \ldots \quad \begin{array}{c}\vdots\ \boxed{\Pi_k}\\ A_i \vdash B_k\end{array} \quad \ldots \quad \begin{array}{c}\vdots\ \boxed{\Pi_\ell}\\ A_i, B_\ell \vdash D\end{array} \quad \ldots}{A_i \vdash D}\,\mathsf{el}_r}{\vdash D}\,\mathsf{el}_{r'}$$

This gives rise to copying of sub-derivations: for every A_i we copy the sub-derivations $\Pi_1, \ldots, \Pi_n$.

Curry-Howard proofs-as-terms

We define rules for the judgment $\Gamma \vdash t : A$, where

- A is a formula,
- Γ is a set of declarations $\{x_1 : A_1, \ldots, x_m : A_m\}$, where the A_i are formulas and the x_i are term-variables,
- t is a proof-term:

$$t ::= x \mid \{\bar{t}\,;\,\overline{\lambda x : A.t}\}_r \mid t \cdot_r [\bar{t}\,;\,\overline{\lambda x : A.t}]$$

where x ranges over variables and r ranges over the rules.

For a connective $c \in \mathcal{C}$, r an introduction rule for c and r' an elimination rule for c, we have

- an introduction term $\{\bar{t}\,;\,\overline{\lambda x : A.t}\}_r$
- an elimination term $t \cdot_{r'} [\bar{t}\,;\,\overline{\lambda x : A.t}]$

Curry-Howard typing rules

Let $\Phi = c(A_1, \ldots, A_n)$ and r a rule for c.

$$\frac{}{\Gamma \vdash x_i : A_i} \text{ if } x_i : A_i \in \Gamma$$

$$\frac{\ldots \Gamma \vdash p_j : A_j \ldots \quad \ldots \Gamma, y_i : A_i \vdash q_i : \Phi \ldots}{\Gamma \vdash \{\overline{p}\,;\, \overline{\lambda y : A.q}\}_r : \Phi} \text{ in}$$

$$\frac{\Gamma \vdash t : \Phi \quad \ldots \Gamma \vdash p_k : A_k \ldots \quad \ldots \Gamma, y_\ell : A_\ell \vdash q_\ell : D}{\Gamma \vdash t \cdot_r [\overline{p}\,;\, \overline{\lambda y : A.q}] : D} \text{ el}$$

Here, $\overline{p}$ is the sequence of terms $p_1, \ldots, p_{m'}$ for all the 1-entries in rule r of the truth table, and $\overline{\lambda y : A.q}$ is the sequence of terms $\lambda y_1 : A_1.q_1, \ldots, \lambda y_m : A_m.q_m$ for all the 0-entries in r.

Reductions on terms for detours

Term reduction rules that correspond to detour conversions.

- For simplicity we write the “matching cases” as last term of the sequence.
- For the $A_j = A_\ell$ case, that is, $p_j : A_j$ and $y_\ell : A_\ell$ with $A_j = A_\ell$:

$$\{\overline{p, p_j} \; ; \; \overline{\lambda x.q}\} \cdot [\overline{s} \; ; \; \overline{\lambda y.r, \lambda y_\ell.r_\ell}] \longrightarrow_a r_\ell[y_\ell := p_j]$$

- For the $A_i = A_k$ case, that is, $x_i : A_i$ and $s_k : A_k$ with $A_i = A_k$:

$$\{\overline{p} \; ; \; \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot [\overline{s, s_k} \; ; \; \overline{\lambda y.r}] \longrightarrow_a q_i[x_i := s_k] \cdot [\overline{s, s_k} \; ; \; \overline{\lambda y.r}]$$

$\overline{p, p_j}$ should be understood as a sequence $p_1, \ldots, p_j, \ldots p_{m'}$, where the p_j that matches the r_ℓ in $\overline{\lambda y.r, \lambda y_\ell.r_\ell}$ has been singled out.

NB There is always (at least one) matching case, because intro/elim rules comes from different lines in the truth table.

Reductions on terms for permutations

We add the following reduction rules for permutation conversions.

$$(t \cdot_r [\overline{p}\ ;\ \overline{\lambda x.q}]) \cdot_{r'} [\overline{s}\ ;\ \overline{\lambda y.r}] \quad \longrightarrow_b \quad t \cdot_r [\overline{p}\ ;\ \overline{\lambda x.(q \cdot_{r'} [\overline{s}\ ;\ \overline{\lambda y.r}])}]$$

Here, $\overline{\lambda x.(q \cdot [\overline{s}\ ;\ \overline{\lambda y.r}])}$ should be understood as a sequence $\lambda x_1.q_1, \ldots, \lambda x_m.q_m$ where each q_j is replaced by $q_j \cdot_{r'} [\overline{s}\ ;\ \overline{\lambda y.r}]$.

Optimized reductions on optimized terms

- On optimized terms, one can also, in a canonical way, define detour conversion $\longrightarrow_a$ and permutation conversion $\longrightarrow_b$.
- Detour reduction on optimised terms translates to (multi-step) detour reduction on the full terms.
- So, strong normalization on optimised terms follows from strong normalization on full terms.
- Other well-known rules, like the general elimination rules studied by Schroeder-Heister and Von Plato, can similarly be translated to our full rules.

Normalization

THEOREM The reduction $\longrightarrow_b$ is strongly normalizing

$$(t \cdot_r [\overline{p}\ ;\ \overline{\lambda x.q}]) \cdot_{r'} [\overline{s}\ ;\ \overline{\lambda y.r}] \quad \longrightarrow_b \quad t \cdot_r [\overline{p}\ ;\ \overline{\lambda x.(q \cdot_{r'} [\overline{s}\ ;\ \overline{\lambda y.r}])}]$$

PROOF The measure $|-|$ decreases with every reduction step.

$$\begin{array}{rcl} |x| & := & 1 \\ |\{\overline{p}\ ;\ \overline{\lambda y.q}\}| & := & \Sigma|p_i| + \Sigma|q_j| \\ |t \cdot [\overline{s}\ ;\ \overline{\lambda y.u}]| & := & |t|(2 + \Sigma|s_k| + \Sigma|u_\ell|) \end{array}$$

Normalization

THEOREM The reduction $\longrightarrow_a$ is strongly normalizing.

$$\{\overline{p, p_j} ; \overline{\lambda x.q}\} \cdot [\overline{s} ; \overline{\lambda y.r, \lambda y_\ell.r_\ell}] \longrightarrow_a r_\ell[y_\ell := p_j]$$

(for the $A_j = A_\ell$ case, $p_j : A_j$ and $y_\ell : A_\ell$ with $A_j = A_\ell$)

$$\{\overline{p} ; \overline{\lambda x.q, \lambda x_i.q_i}\} \cdot [\overline{s, s_k} ; \overline{\lambda y.r}] \longrightarrow_a q_i[x_i := s_k] \cdot [\overline{s, s_k} ; \overline{\lambda y.r}]$$

(for the $A_i = A_k$ case, $x_i : A_i$ and $s_k : A_k$ with $A_i = A_k$)

PROOF We adapt the saturated sets method of Tait.

COROLLARY the combination $\longrightarrow_{ab}$ is weakly normalizing.
Basically: take the $\longrightarrow_b$-normal-form and then contract the innermost $\longrightarrow_a$-redex of highest rank. (This generalizes the Gandy-Turing SN proof for simple type theory, $\lambda \rightarrow$.)

Strong Normalization

New: we have obtained a proof of Strong Normalization for general $\mathrm{IPC}_{\mathcal{C}}$.

Rough outline of the proof (generalizing a proof of SN for IPC by Philippe De Groote):

- Define a "double negation" translation from $\mathrm{IPC}_{\mathcal{C}}$ formulas to $\lambda \to$-types.
- Define a reduction preserving "CPS" translation from $\mathrm{IPC}_{\mathcal{C}}$ terms to $\lambda \to$-parallel.
  ($\lambda \to$ extended with $[M_1, \ldots, M_n] : A$ if $M_i : A$ for $1 \leq i \leq n$.)
- Prove SN for $\lambda \to$-parallel.

$\lambda\rightarrow$-parallel

- Types: $\sigma ::= o \mid (\sigma \to \sigma)$
- Terms: $M ::= x \mid (M\,M) \mid (\lambda x.M) \mid [M_1, \ldots, M_n]$ $(n > 1)$.
- Typing rules

$$\frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M\,N : B} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma \vdash M_1 : A \quad \ldots \quad \Gamma \vdash M_n : A}{\Gamma \vdash [M_1, \ldots, M_n] : A}$$

- Reduction rules: $(\lambda x.M)\,N \longrightarrow_\beta M[x := N]$ plus

$$[M_1, \ldots, M_n]\,N \longrightarrow_\beta [M_1\,N, \ldots, M_n\,N]$$

SN can be proved by adapting the well-known Tait proof.

Translating formulas to types (outline)

Abbreviate $\neg A := A \to o$.

- For a proposition letter, $\widehat{A} := \neg\neg A$.
- For $\Phi = c(A_1, \ldots, A_n)$ with elimination rules $r_1, \ldots, r_t$

$$\widehat{\Phi} := \neg(E_1 \to \cdots \to E_t \to o),$$

where

$$E_s := \widehat{A_{k_1}} \to \ldots \to \widehat{A_{k_m}} \to \neg\widehat{A_{l_1}} \to \ldots \to \neg\widehat{A_{l_{n-m}}} \to o$$

with the A_k the 1-entries and the A_l are the 0-entries in the truth table.

For example

$$\widehat{A \wedge B} = \neg(\neg\neg\widehat{A} \to \neg\neg\widehat{B} \to o)$$

$$\widehat{A \vee B} = \neg((\neg\widehat{A} \to \neg\widehat{B} \to o) \to o)$$

Translating proof-terms to $\lambda\rightarrow$-parallel terms (outline)

Let $\Phi = c(A_1, \ldots, A_n)$ have elimination rules $r_1, \ldots, r_t$.

- $\widehat{x} := \lambda h.x\, h$.
- Elimination term:

$$M \cdot_{r_s} \widehat{[\overline{N}\,;\,\overline{\lambda x.Q}]} := \lambda h.\widehat{M}\,(\lambda g_1 \ldots g_t.g_s\,\overline{\widehat{N}}(\overline{\lambda x.\widehat{Q}\,h})).$$

- Introduction term

$$\widehat{\{\overline{N}\,;\,\overline{\lambda y.M}\}_r} := \lambda h.h\, e_1^h\, \ldots\, e_t^h,$$

where e_s^h is the possibly parallel term containing

- $\overline{\lambda f}.f_\ell\,\widehat{N_j}$ for ℓ in rule r_s and j with $A_j = A_\ell$.
- $\overline{\lambda f}.(\lambda y_i.\widehat{M_i}\,h)$ for k in rule r_s and i with $A_i = A_k$.

Translating proof-terms to $\lambda \rightarrow$-parallel terms (outline)

Using the translation $\widehat{M}$ we define a second translation $\widehat{\widehat{M}}$. (This is a generalization of the CPS translation $\overline{\overline{M}}$ of Plotkin, that De Groote also uses.)

We can prove

- If $M \longrightarrow_b N$, then $\widehat{\widehat{M}} = \widehat{\widehat{N}}$
- If $\widehat{\widehat{M}} \subset K$ ($\widehat{\widehat{M}}$ is a subterm of K), then

$$\begin{array}{ccccc} M & \mapsto & \widehat{\widehat{M}} & \subset & K \\ \Big\downarrow a & & & & \Big\downarrow \beta\,+ \\ N & \mapsto & \widehat{\widehat{N}} & \subset & \exists K' \end{array}$$

From this we can derive Strong Normalization.

Conclusions

- Simple general way to derive constructive deduction rules for (new) connectives.
- Study connectives “in isolation”. (Without other connectives.)
- General definition of detour conversion and permutation conversion.
- General Curry-Howard proofs-as-terms interpretation.
- General Strong Normalization proof.

Future work and Related

- Meaning of the new connectives as inductive data types.
- Study conditions for the set of rules to be Church-Rosser.
- General definition of classical detour/permutation conversion
- Relation with other well-known term calculi for classical logic: subtraction logic (Crolard), $\lambda\mu$ (Parigot), $\bar{\lambda}\mu\tilde{\mu}$ (Curien, Herbelin).

Related work:

- Roy Dyckhoff, Peter Milne, Jan von Plato and Sara Negri, Peter Schroeder-Heister, ...
- "Harmony" in logic (following Prawitz)

Questions?