Proof-term reductions for general forms of natural deduction

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

Institute for Computing and Information Science
Radboud University
Nijmegen, NL

Term Rewriting Seminar
Radboud University Nijmegen
May 14 2018

Classical and Constructive Logic

Classically, the meaning of a propositional connective is fixed by its truth table. This immediately implies

- consistency,
- a decision procedure,
- completeness (w.r.t. Boolean algebra's).

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),
- a decision procedure (from the subformula property),
- Curry-Howard proofs-as-terms (and propositions-as-types) allowing to study normalization as term-reduction

This talk

- Derive natural deduction rules for a connective from its truth table definition.
  - For constructive logic!
  - 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.
- Give proof-terms for natural deduction
- Study proof-normalization via term-reduction
- We define a general notion of detour conversion (cut-elimination) for these constructive connectives.

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 the Γ explicitly, as it can be retrieved:

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

Some well-known constructive rules

Rules that follow this format:

$$\frac{\vdash A \vee B \qquad A \vdash D \qquad B \vdash D}{\vdash D}\ \vee\text{-el} \qquad \frac{\vdash A \wedge B \qquad A \vdash D}{\vdash D}\ \wedge\text{-el}$$

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

Rule that does not follow this format:

$$\frac{A \vdash B}{\vdash A \to B}\ \to\text{-in}$$

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

$$\begin{array}{ccc|c} A_1 & \ldots & A_n & \varphi \\ \hline p_1 & \ldots & p_n & 0 \end{array} \quad \mapsto \quad \frac{\vdash \varphi \ldots \vdash A_j \text{ (if } p_j = 1) \ldots A_i \vdash D \text{ (if } p_i = 0) \ldots}{\vdash D}\,\text{el}$$

introduction

$$\begin{array}{ccc|c} A_1 & \ldots & A_n & \varphi \\ \hline q_1 & \ldots & q_n & 1 \end{array} \quad \mapsto \quad \frac{\ldots \vdash A_j \text{ (if } q_j = 1) \ldots A_i \vdash \varphi \text{ (if } q_i = 0) \ldots}{\vdash \varphi}\,\text{in}^i$$

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 $\mathcal{C}$, 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:

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

Lemma I to simplify the rules

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

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

is equivalent to the system with these two rules replaced by

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

Lemma II to simplify the rules

A system with a deduction rule of the form to the left is equivalent to the system with this rule replaced by the rule on the right.

$$\frac{\vdash A_1 \;\ldots\; \vdash A_n \quad B \vdash D}{\vdash D} \qquad\qquad \frac{\vdash A_1 \;\ldots\; \vdash A_n}{\vdash B}$$

The well-known constructive connectives

We have already seen the $\wedge, \neg$ rules. The optimised rules for $\vee, \rightarrow, \top$ and $\bot$ we obtain are:

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

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

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

Substituting a deduction in another

LEMMA: If $\Delta, \varphi \vdash \psi$ and $\Gamma \vdash \varphi$, then $\Delta, \Gamma \vdash \psi$

If Π is a deduction of $\Delta, \varphi \vdash \psi$ and Σ is a deduction of $\Gamma \vdash \varphi$, then we have the following deduction of $\Delta, \Gamma \vdash \psi$:

$$\begin{array}{c}\begin{array}{ccc}\boxed{\Sigma} & & \boxed{\Sigma} \\ \Gamma \vdash \varphi & \cdots & \Gamma \vdash \varphi\end{array} \\ \boxed{\Pi} \\ \Delta, \Gamma \vdash \psi\end{array}$$

In Π, every application of an (axiom) rule at a leaf, deriving $\Delta' \vdash \varphi$ for some $\Delta' \supseteq \Delta$ is replaced by a copy of a deduction Σ, which is also a deduction of $\Delta', \Gamma \vdash \varphi$.

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

Eliminating a detour (detour conversion) (I)

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

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

by

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

Eliminating a detour (detour conversion) (II)

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

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

by

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

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: cut-elimination is non-deterministic in general.

Permutation convertibility: Example

$$\dfrac{\dfrac{\vdash A\vee B \qquad \dfrac{A, C\vdash C\to D}{A\vdash C\to D}\to\text{-in}_a \qquad B\vdash C\to D}{\vdash C\to D}\vee\text{-el} \qquad \vdash C}{\vdash D}\to\text{-el}$$

A detour convertibility arising from $\to$ -in$_a$ followed by $\to$ -el is blocked by the $\vee$-el. This is a permutation convertibility, which can be contracted by permuting the $\to$ -el rule over the $\vee$-el rule.

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 $\Phi = c(B_1, \dots, B_n)$, $\Psi = c'(A_1, \dots, A_{n'})$.

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

- 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

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

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

Permutation conversion: Example

A permutation conversion replaces the following derivation

$$\dfrac{\dfrac{\vdash A\vee B \qquad \dfrac{A, C\vdash C\to D}{A\vdash C\to D}\,\to\text{-in}_a \qquad B\vdash C\to D}{\vdash C\to D}\,\vee\text{-el} \qquad \vdash C}{\vdash D}\,\to\text{-el}$$

by

$$\dfrac{\vdash A\vee B \qquad \dfrac{\dfrac{A, C\vdash C\to D}{A\vdash C\to D}\,\to\text{-in}_a \qquad A\vdash C}{A\vdash D}\,\to\text{-el} \qquad \dfrac{B\vdash C\to D \qquad B\vdash C}{B\vdash D}\,\to\text{-el}}{\vdash D}\,\vee\text{-el}$$

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:

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

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

Curry-Howard typing rules

The terms are *typed* using the following derivation rules.

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

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

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

Here, $\overline{p}$ is the sequence of terms $p_1, \ldots, p_{m'}$ for all the 1-entries in 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 the truth table.

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 $j = \ell$ 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 $i = k$ 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.

Example: reductions for $\wedge$-terms

The rules for conjunction are as follows.

$$\frac{\vdash t : A \wedge B \quad x : A \vdash p : D \quad y : B \vdash q : D}{\vdash t \cdot_1^{\wedge} [\,;\, \lambda x.p, \lambda y.q] : D} \qquad \frac{\vdash t : A \wedge B \quad \vdash a : A \quad y : B \vdash q : D}{\vdash t \cdot_2^{\wedge} [a \,;\, \lambda y.q] : D}$$

$$\frac{\vdash t : A \wedge B \quad x : A \vdash p : D \quad \vdash b : B}{\vdash t \cdot_3^{\wedge} [b \,;\, \lambda x.p] : D} \qquad \frac{\vdash a : A \quad \vdash b : B}{\vdash \{a, b \,;\, \}^{\wedge} : A \wedge B}$$

The reduction rules are

$$\begin{aligned}
\{a, b \,;\, \}^{\wedge} \cdot_1^{\wedge} [\,;\, \lambda x{:}A.p, \lambda y{:}B.q] &\longrightarrow_a p[x := a] \\
\{a, b \,;\, \}^{\wedge} \cdot_1^{\wedge} [\,;\, \lambda x{:}A.p, \lambda y{:}B.q] &\longrightarrow_a q[y := b] \\
\{a, b \,;\, \}^{\wedge} \cdot_2^{\wedge} [a' \,;\, \lambda y{:}B.q] &\longrightarrow_a q[y := b] \\
\{a, b \,;\, \}^{\wedge} \cdot_3^{\wedge} [b' \,;\, \lambda x{:}A.p] &\longrightarrow_a p[x := a]
\end{aligned}$$

From the first two cases, we see that the Church-Rosser property (confluence) is lost.

Example: reductions for $\to$-terms

The rules for implication are as follows.

$$\frac{x : A \vdash p : A \to B \quad y : B \vdash q : A \to B}{\vdash \{\ ;\lambda x.p, \lambda y.q\}^{\to}_{1} : A \to B}
\qquad
\frac{x : A \vdash p : A \to B \quad \vdash b : B}{\vdash \{b\ ;\lambda x.p\}^{\to}_{2} : A \to B}$$

$$\frac{\vdash t : A \to B \quad \vdash a : A \quad z : B \vdash r : D}{\vdash t \cdot^{\to} [a\ ;\lambda z.r] : D}
\qquad
\frac{\vdash a : A \quad \vdash b : B}{\vdash \{a, b\ ;\ \}^{\to}_{3} : A \to B}$$

The reduction rules are

$$\begin{array}{rcl}
\{\ ;\lambda x{:}A.p, \lambda y{:}B.q\}^{\to}_{1} \cdot^{\to} [a\ ;\lambda z{:}B.r] & \longrightarrow_a & p[x := a] \cdot^{\to} [a\ ;\lambda z{:}B.r] \\
\{b\ ;\lambda x{:}A.p\}^{\to}_{2} \cdot^{\to} [a\ ;\lambda z{:}B.r] & \longrightarrow_a & r[z := b] \\
\{b\ ;\lambda x{:}A.p\}^{\to}_{2} \cdot^{\to} [a\ ;\lambda z{:}B.r] & \longrightarrow_a & p[x := a] \cdot^{\to} [a\ ;\lambda z{:}B.r] \\
\{a', b\ ;\ \}^{\to}_{3} \cdot^{\to} [a\ ;\lambda z{:}B.r] & \longrightarrow_a & r[z := b]
\end{array}$$

In the first and the third case, we see that the elimination remains.

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 $q_j \cdot_{r'} [\overline{s}\ ;\ \overline{\lambda y.r}]$.

Example of permutation reduction on terms

$$\dfrac{\dfrac{\vdash t : A \vee B \quad x : A \vdash p : C \to D \quad y : B \vdash q : C \to D}{\vdash t \cdot^{\vee} [\,;\, \lambda x.p, \lambda y.q] : C \to D} \qquad \vdash c : C \quad z : D \vdash r : E}{\vdash t \cdot^{\vee} [\,;\, \lambda x.p, \lambda y.q] \cdot^{\to} [c \,;\, \lambda z.r] : E}$$

The two consecutive elimination rules can be permuted. The term reduces as follows

$$t \cdot^{\vee} [\,;\, \lambda x.p, \lambda y.q] \cdot^{\to} [c \,;\, \lambda z.r] \longrightarrow_b$$
$$t \cdot^{\vee} [\,;\, \lambda x.p \cdot^{\to} [c \,;\, \lambda z.r], \lambda y.q \cdot^{\to} [c \,;\, \lambda z.r]]$$

Optimized reductions and optimized terms

The usual "pairing" is given by the introduction rule: $\{a, b\ ;\ \}^{\wedge}$.
For elimination, we want to have the "projection" rules:

$$\frac{\vdash t : A \wedge B}{\vdash \pi_1\, t : A} \qquad \frac{\vdash t : A \wedge B}{\vdash \pi_2\, t : B}$$

We can define

$$\pi_1\, t := t \cdot_1^{\wedge} [\ ;\ \lambda x^A.x, \lambda z^B.t \cdot_3^{\wedge} [z\ ;\ \lambda x^A.x]]$$

which has the following reductions.

$$\begin{array}{rcl} \pi_1\, \{a, b\ ;\ \}^{\wedge} & = & \{a, b\ ;\ \}^{\wedge} \cdot_1^{\wedge} [\ ;\ \lambda x^A.x, \lambda z^B.\{a, b\ ;\ \}^{\wedge} \cdot_3^{\wedge} [z\ ;\ \lambda x^A.x]] \\ & \longrightarrow_a & a \\ \pi_1\, \{a, b\ ;\ \}^{\wedge} & = & \{a, b\ ;\ \}^{\wedge} \cdot_1^{\wedge} [\ ;\ \lambda x^A.x, \lambda z^B.\{a, b\ ;\ \}^{\wedge} \cdot_3^{\wedge} [z\ ;\ \lambda x^A.x]] \\ & \longrightarrow_a & \{a, b\ ;\ \}^{\wedge} \cdot_3^{\wedge} [b\ ;\ \lambda x^A.x] \\ & \longrightarrow_a & a \end{array}$$

Another ∧-elimination rule

The parallel ∧-elimination rule ("general elimination rule") studied by Schroeder-Heister and Von Plato is:

$$\frac{\Gamma \vdash t : A \wedge B \qquad \Gamma, x : A, y : B \vdash q : D}{\Gamma \vdash t \cdot^{\mathrm{par}} [\lambda x, y.q] : D} \wedge\text{-el}$$

The reduction for this rule is as follows.

$$\{a, b\, ;\, \} \cdot^{\mathrm{par}} [\lambda x, y.q] \longrightarrow_{\mathrm{par}} q[x := a, y := b].$$

We can define the parallel ∧-elimination rule

$$\begin{array}{rcl} t \cdot^{\mathrm{par}} [\lambda x, y.q] & := & t \cdot_1^{\wedge} [\, ;\, \lambda x.p, \lambda z^B.t \cdot_3^{\wedge} [z\, ;\, \lambda x.p]], \\ \text{where } p & = & t \cdot_1^{\wedge} [\, ;\, \lambda u^A.t \cdot_2^{\wedge} [u\, ;\, \lambda y.q], \lambda y.q] \end{array}$$

Then

$$\{a, b\, ;\, \} \cdot^{\mathrm{par}} [\lambda x, y.q] \longrightarrow_a^+ q[x := a, y := b].$$

Relating optimized reductions to the original reductions

- Usually, when studying constructive natural deduction, one uses variants of the optimized rules for natural deduction.
- For the optimized rules, there is also a straightforward definition of proof-terms and of the reduction relations $\longrightarrow_a$ and $\longrightarrow_b$.
- Proposition: The proof-terms for the optimized rules can be defined in terms of the terms for the full calculus.
- Proposition: The reduction rules for the optimized proof terms are an instance of reductions in the full calculus (often multi-step).
- So: Strong Normalization for the optimized calculus follows from Strong Normalization for the full calculus.

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 $j = \ell$ 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 $i = k$ 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

Recently (master thesis of Iris van der Giessen) we have obtained a proof of Strong Normalization for general $\mathrm{IPC}_{\mathcal{C}}$.

Method (generalizing a proof by Philippe De Groote)

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

Consequences of Normalization

The set of terms in normal form of $\mathrm{IPC}_{\mathcal{C}}$, NF is characterized by the following inductive definition.

- $x \in \mathrm{NF}$ for every variable x,
- $\{\overline{p}\ ;\ \overline{\lambda y.q}\} \in \mathrm{NF}$ if all p_i and q_j are in NF,
- $x \cdot [\overline{p}\ ;\ \overline{\lambda y.q}] \in \mathrm{NF}$ if all p_i and q_j are in NF and x is a variable.

As corollaries of Normalization we have, for an arbitrary set of connectives:

- subformula property
- consistency of the logic
- decidability of the logic

The 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.

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

The 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,

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

The Sheffer stroke or NAND connective [III]

The proof-terms for nand-logic are

$$\frac{x : A \vdash p : A \uparrow B}{\vdash \{\ ;\ \lambda x^A.p\}^{\uparrow} : A \uparrow B} \qquad \frac{y : B \vdash q : A \uparrow B}{\vdash \{\ ;\ \lambda y^B.q\}^{\uparrow} : A \uparrow B}$$

$$\frac{\vdash t : A \uparrow B \quad \vdash a : A \quad \vdash b : B}{\vdash t \cdot^{\uparrow} [a, b\ ;\ ] : D}$$

with reduction rules

$$\begin{array}{lll} \{\ ;\ \lambda x^A.p\}^{\uparrow} \cdot^{\uparrow} [a, b\ ;\ ] & \longrightarrow_a & p[x := a] \cdot^{\uparrow} [a, b\ ;\ ] \\ \{\ ;\ \lambda y^B.q\}^{\uparrow} \cdot^{\uparrow} [a, b\ ;\ ] & \longrightarrow_a & q[y := b] \cdot^{\uparrow} [a, b\ ;\ ] \end{array}$$

Conclusions

- Simple general way to derive constructive deduction rules for (new) connectives.
- Study connectives “in isolation”. (Without defining them and without using other connectives.)
- General definition of detour conversion and permutation conversion.
- General proof-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 term calculi for classical logic: subtraction logic, $\lambda\mu$ (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)