Deriving derivation rules from truth tables: classically, constructively and proof reduction

Herman Geuvers

Radboud University Nijmegen
and Technical University Eindhoven
NL

World Logic Day 2021
Tallinn Estonia
Joint work with Tonny Hurkens and Iris van der Giessen

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 (following the Brouwer-Heyting-Kolmogorov interpretation), 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),
- completeness (w.r.t. Heyting algebra's and Kripke models).

This talk

- Derive natural deduction rules for a connective from its truth table definition.
  - Also works for constructive logic.
  - Gives natural deduction rules for a connective "in isolation"
  - Also gives (constructive) rules for connectives that haven't been studied so far, like if-then-else and nand.
- General definition, both the constructive and the classical case.
- Relation to "standard" natural deduction rules and known connectives.
- General Kripke model for the constructive connectives. (Sound and Complete)
- Curry-Howard proofs-as-terms interpretation for derivations and normalization of proof-reduction
- Interpreting classical proofs as terms.

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

We don't give the Γ explicitly (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:

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

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 $\Phi = c(A_1, \ldots, A_n)$.)

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

constructive intro

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

classical intro

$$\begin{array}{ccc|c} A_1 & \ldots & A_n & \Phi \\ \hline r_1 & \ldots & r_n & 1 \end{array} \quad \mapsto \quad \frac{\Phi \vdash D \ldots \vdash A_j \text{ (if } r_j = 1) \ldots A_i \vdash D \text{ (if } r_i = 0) \ldots}{\vdash D} \text{in}^c$$

Examples

Constructive 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

$$\frac{\vdash A \wedge B \quad A \vdash D \quad B \vdash D}{\vdash D}\ \wedge\text{-el}_{00} \qquad \frac{\vdash A \wedge B \quad A \vdash D \quad \vdash B}{\vdash D}\ \wedge\text{-el}_{01}$$

$$\frac{\vdash A \wedge B \quad \vdash A \quad B \vdash D}{\vdash D}\ \wedge\text{-el}_{10} \qquad \frac{\vdash A \quad \vdash B}{\vdash A \wedge B}\ \wedge\text{-in}_{11}$$

- Can be shown to be equivalent to the well-known constructive 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

Constructive:

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

Classical:

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

Lemma I to simplify the rules

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

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

is equivalent to the system with these two rules replaced by

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

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.

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

The constructive connectives

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

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

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

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

The rules for the classical $\to$ connective

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

Derivation of Peirce's law:

$$\dfrac{\dfrac{A \vdash A}{A \vdash ((A \to B) \to A) \to A} \qquad \dfrac{\dfrac{\dfrac{(A \to B) \to A \vdash (A \to B) \to A \quad A \to B \vdash A \to B}{A \to B, (A \to B) \to A \vdash A}}{A \to B, (A \to B) \to A \vdash ((A \to B) \to A) \to A}}{A \to B \vdash ((A \to B) \to A) \to A}}{\vdash ((A \to B) \to A) \to A} \to\text{-in}_2^c$$

The "If Then Else" connective

Notation: $A \to B/C$ for if A then B else C.

p	q	r	$p \to q/r$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

The optimized constructive rules are:

$$\frac{\vdash A \to B/C \quad \vdash A}{\vdash B}\ \text{then-el} \qquad \frac{\vdash A \to B/C \quad A \vdash D \quad C \vdash D}{\vdash D}\ \text{else-el}$$

$$\frac{\vdash A \quad \vdash B}{\vdash A \to B/C}\ \text{then-in} \qquad \frac{A \vdash A \to B/C \quad \vdash C}{\vdash A \to B/C}\ \text{else-in}$$

Some facts about constructive “If Then Else”

$A \to B/C$ is logically equivalent to $(A \to B) \wedge (A \vee C)$

We have the well-known classical equivalence

$$\text{if } A \text{ then } B \text{ else } B \equiv B$$

We don't have the other well-known classical equivalences

$$\text{if (if } A \text{ then } B \text{ else } C) \text{ then } D \text{ else } E \quad \nvdash \quad \text{if } A \text{ then (if } B \text{ then } D \text{ else } E) \text{ else (if } C \text{ then } D \text{ else } E)$$

$$\text{if } A \text{ then (if } B \text{ then } D \text{ else } E) \text{ else (if } C \text{ then } D \text{ else } E) \quad \nvdash \quad \text{if (if } A \text{ then } B \text{ else } C) \text{ then } D \text{ else } E$$

"If Then Else" $+\top+\bot$ is functionally complete

We can define the usual constructive connectives in terms of if-then-else, $\top$ and $\bot$:

$$A \mathbin{\dot\vee} B \;:=\; A{\to}A/B \qquad A \mathbin{\dot\wedge} B \;:=\; A{\to}B/A$$

$$A \mathbin{\dot\to} B \;:=\; A{\to}B/\top \qquad \dot\neg A \;:=\; A{\to}\bot/\top$$

LEMMA The defined connectives satisfy the original constructive deduction rules for these same connectives.

COROLLARY The constructive connective if-then-else, together with $\top$ and $\bot$, is functionally complete.

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,

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

Kripke semantics for the constructive rules

For each n-ary connective c, we assume a truth table $t_c : \{0,1\}^n \to \{0,1\}$ and the defined constructive deduction rules.

DEFINITION A Kripke model is a triple $(W, \leq, \mathrm{at})$ where W is a set of worlds, $\leq$ a reflexive, transitive relation on W and a function $\mathrm{at} : W \to \wp(\mathrm{At})$ satisfying $w \leq w' \Rightarrow \mathrm{at}(w) \subseteq \mathrm{at}(w')$.

We define the notion φ is true in world w (usually written $w \Vdash \varphi$) by defining $[\![\varphi]\!]_w \in \{0,1\}$

DEFINITION of $[\![\varphi]\!]_w \in \{0,1\}$, by induction on φ:

- (atom) if φ is atomic, $[\![\varphi]\!]_w = 1$ iff $\varphi \in \mathrm{at}(w)$.
- (connective) for $\varphi = c(\varphi_1, \ldots, \varphi_n)$, $[\![\varphi]\!]_w = 1$ iff for each $w' \geq w$, $t_c([\![\varphi_1]\!]_{w'}, \ldots, [\![\varphi_n]\!]_{w'}) = 1$ where t_c is the truth table of c.

$\Gamma \models \psi$:= for each Kripke model and each world w, if $[\![\varphi]\!]_w = 1$ for each φ in Γ, then $[\![\psi]\!]_w = 1$.

Kripke semantics for the constructive rules

LEMMA (Soundness) If $\Gamma \vdash \psi$, then $\Gamma \models \psi$
Proof. Induction on the derivation of $\Gamma \vdash \psi$.

For completeness we need to construct a special Kripke model.

- In the literature, the completeness of Kripke semantics is proved using *prime theories*.
- A theory is prime if it satisfies the disjunction property: if $\Gamma \vdash A \vee B$, then $\Gamma \vdash A$ or $\Gamma \vdash B$.
- We may not have $\vee$ in our set of connective, and we may have others that "behave $\vee$-like"',
- (But we can generalize the disjunction property to arbitrary n-ary constructive connectives that are "splitting".)
- We apply a kind of Lindenbaum construction (also used by Milne for the classical case).

Kripke semantics for the constructive rules

DEFINITION For ψ a formula and Γ a set of formulas, we say that Γ is ψ-maximal if

- $\Gamma \nvdash \psi$ and
- for every formula $\varphi \notin \Gamma$ we have: $\Gamma, \varphi \vdash \psi$.

NB. Given ψ and Γ such that $\Gamma \nvdash \psi$, we can extend Γ to a ψ-maximal set Γ' that contains Γ.

Simple important facts about ψ-maximal sets Γ:

1. For every φ, we have $\varphi \in \Gamma$ or $\Gamma, \varphi \vdash \psi$.
2. For every φ, if $\Gamma \vdash \varphi$, then $\varphi \in \Gamma$.

Completeness of Kripke semantics

DEFINITION We define the Kripke model $U = (W, \leq, \text{at})$:

- $W := \{(\Gamma, \psi) \mid \Gamma \text{ is a } \psi\text{-maximal set}\}$.
- $(\Gamma, \psi) \leq (\Gamma', \psi') := \Gamma \subseteq \Gamma'$.
- $\text{at}(\Gamma, \psi) := \Gamma \cap \text{At}$.

LEMMA In the model U we have, for all worlds $(\Gamma, \psi) \in W$:

$$\varphi \in \Gamma \iff [\![\varphi]\!]_{(\Gamma,\psi)} = 1 \quad (\forall \varphi)$$

Proof: Induction on the structure of φ.

THEOREM If $\Gamma \models \psi$, then $\Gamma \vdash \psi$.
Proof. Suppose $\Gamma \models \psi$ and $\Gamma \not\vdash \psi$. Then we can find a ψ-maximal superset Γ' of Γ such that $\Gamma' \not\vdash \psi$. In particular: ψ is not in Γ'. So (Γ', ψ) is a world in the Kripke model U in which each member of Γ is true, but ψ is not. Contradiction to $\Gamma \models \psi$, so $\Gamma \vdash \psi$.

Some general proof-theoretic properties

The n-ary connective c is i, j-splitting in case

$$t_c(p_1, \ldots, p_{i-1}, 0, p_{i+1}, \ldots, p_{j-1}, 0, p_{j+1}, \ldots, p_n) = 0$$

for all $p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_{j-1}, p_{j+1}, \ldots, p_n \in \{0, 1\}$.

LEMMA For c an i, j-splitting connective, if $\vdash c(A_1, \ldots, A_n)$, then $\vdash A_i$ or $\vdash A_j$.
For example: if $\vdash A \to B / C$, then $\vdash A$ or $\vdash C$. (And also: if $\vdash A \to B / C$, then $\vdash B$ or $\vdash C$.)

An n-ary connective c is monotonic if $t_c : \{0, 1\}^n \to \{0, 1\}$ is monotonic under the ordering induced by $0 \leq 1$.

LEMMA For c monotonic, the classical and constructive derivation rules are equivalent.

LEMMA For c_1, c_2 non-monotonic, if we take the classical rules for c_1 and the constructive rules for c_2, we can derive the classical rules for c_2.

Substituting a deduction in another

LEMMA: If $\Gamma \vdash A$ and $\Delta, A \vdash B$, then $\Gamma, \Delta \vdash B$

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

$$\begin{array}{ccc} \vdots\,\Sigma & & \vdots\,\Sigma \\ \Gamma \vdash A & \ldots & \Gamma \vdash A \\ & \vdots\,\Pi & \\ & \Delta \vdash B & \end{array}$$

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

Detours (cuts) in constructive logic

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

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

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

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

- $q_j = 1$ 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, so $A_j = A_\ell$), replace

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

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

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

by

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

Observation

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

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

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

- 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

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

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

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:

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

  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 $t \cdot_{r'} [\overline{t} \,;\, \overline{\lambda x : A.t}]$

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 $j = \ell$ 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.

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 optimized terms translates to (multi-step) detour reduction on the full terms.
- So, strong normalization on optimized 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 WN proof for simple type theory, $\lambda\rightarrow$.)

Strong Normalization

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

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

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

- 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)

We have a translation $\widehat{M}$ and 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

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

From this we derive Strong Normalization.

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

Classical logic

For classical logic, we have:

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

classical intro

$$\begin{array}{ccc|c} A_1 & \dots & A_n & \Phi \\ \hline r_1 & \dots & r_n & 1 \end{array} \quad \mapsto \quad \frac{\color{red}{\Phi \vdash D} \dots \vdash A_j \text{ (if } r_j = 1) \dots A_i \vdash D \text{ (if } r_i = 0) \dots}{\vdash D}\text{in}$$

- If $p_j = 1$ (or $r_j = 1$) in t_c, then A_j occurs as Lemma in the rule
- If $p_j = 0$ (or $r_j = 0$) t_c, then A_i occurs as Casus in the rule

We call $\vdash \Phi$ (resp. $\Phi \vdash D$) the major premise and the other hypotheses of the rule we call the minor premises.

Proof terms for classical logic

$$t ::= x \mid (\lambda y : A.t) \star_r \{\overline{t}\ ;\ \overline{\lambda x : A.t}\} \mid t \cdot_r [\overline{t}\ ;\ \overline{\lambda x : A.t}]$$

where x ranges over variables and r ranges over the rules of all the connectives.

The terms are typed using the following derivation rules.

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

$$\frac{\Gamma, z : \Phi \vdash t : D \ldots \Gamma \vdash p_i : A_i \ldots \quad \ldots \Gamma, y_j : A_j \vdash q_j : D \ldots}{\Gamma \vdash (\lambda z : \Phi.t) \star_r \{\overline{p}\ ;\ \overline{\lambda y : A.q}\} : D} \text{ in}$$

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

Reduction for proof terms in classical logic

- First perform permutation reductions.
- Then we perform detour reductions.

This is similar to the constructive case, except for now

- a term is in permutation normal form if all lemmas are variables,
- a detour is an elimination of Φ followed by an introduction of Φ.

NB: in constructive logic, a "detour" is an introduction directly followed by an elimination. Here it is the other way around, and the introduction need not follow the elimination directly.

This is the abstract syntax N for permutation normal forms:

$$N ::= x \mid (\lambda y : A.N) \star \{\overline{z}\ ;\ \overline{\lambda x : A.N}\} \mid y \cdot [\overline{z}\ ;\ \overline{\lambda x : A.N}],$$

where x, y, z range over variables.

Detours for proof terms in classical logic

A detour is a pattern of the following shape

$$(\lambda x : \Phi. \ldots (x \cdot [\overline{v} \,;\, \overline{\lambda w : A.s}]) \ldots) \star \{\overline{z} \,;\, \overline{\lambda y : A.q}\}$$

that is, an elimination of $\Phi = c(A_1, \ldots, A_n)$ followed by an introduction of Φ, with an arbitrary number of steps in between.

For terms in permutation normal form, detours can be eliminated, obtaining a term in normal form which satisfies the sub-formula property.

Notes to the pattern of a detour:

- the indicated occurrence need not be the only occurrence of x
- variable x may not occur at all; that is the simplest situation.

Eliminating detours

Eliminating detours is done by the following reduction steps:

- $(\lambda x : \Phi. \ldots (x \cdot [\overline{v} \,;\, \overline{\lambda w : A.s}]) \ldots) \star \{\overline{z} \,;\, \overline{\lambda y : A.q}\} \longrightarrow_a$
  $(\lambda x : \Phi. \ldots (s_\ell[w_\ell := z_i]) \ldots) \star \{\overline{z} \,;\, \overline{\lambda y : A.q}\}$

  if $i = \ell\,(A_i = A_\ell)$ is a "matching case" for the subformulas of Φ.

- $(\lambda x : \Phi. \ldots (x \cdot [\overline{v} \,;\, \overline{\lambda w : A.s}]) \ldots) \star \{\overline{z} \,;\, \overline{\lambda y : A.q}\} \longrightarrow_a$
  $(\lambda x : \Phi. \ldots (q_j[y_j := v_k]) \ldots) \star \{\overline{z} \,;\, \overline{\lambda y : A.q}\}$

  if $j = k\,(A_j = A_k)$ is a "matching case" for the subformulas of Φ.

- $(\lambda x : \Phi.t) \star \{\overline{z} \,;\, \overline{\lambda y : A.q}\} \longrightarrow_a t \qquad$ if $x \notin \mathrm{FV}(t)$.

Tonny Hurkens has given a proof that this normalizes

Conclusions

- Simple general way to derive constructive and classical deduction rules for (new) connectives.
- Study connectives “in isolation”. (Without other connectives.)
- Generic Kripke semantics for constructive logic
- General definitions 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.
- Study the computational meaning of classical proof terms.
- 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:

- Dyckhoff; Milne; von Plato and Negri; Schroeder-Heister; Joachimski and Matthes; Baaz, Fermüller and Zach; Abel; ...
- "Harmony" in logic (following Prawitz)

Questions?