From pre-models to models

normalization by Heyting algebras

Olivier HERMANT

18 Mars 2008

Deduction System : natural deduction (NJ)

- first-order logic: function and predicate symbols, logical connectors: $\wedge, \vee, \Rightarrow, \neg$, and quantifiers $\forall, \exists$.

$$\frac{}{\Gamma, A \vdash A}\text{axiom}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \wedge\text{-i} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \wedge\text{-e1} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \wedge\text{-e2}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \Rightarrow\text{-i} \qquad \frac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B} \Rightarrow\text{-e}$$

$$\frac{\Gamma \vdash \forall x A[x]}{\Gamma \vdash A[t]}\forall\text{-e, any } t \qquad \frac{\Gamma \vdash A[x]}{\Gamma \vdash \forall x A[x]}\forall\text{-i, } x \text{ free}$$

Deduction modulo: allowed rewriting

- General form (free variables are possible):

$$l \to r$$

Deduction modulo: allowed rewriting

- General form (free variables are possible):

$$l \to r$$

- use: We replace $t = \sigma l$ by σr (unification). Rewriting could be deep in the term.

Deduction modulo: allowed rewriting

- General form (free variables are possible):

$$l \to r$$

- use: We replace $t = \sigma l$ by σr (unification). Rewriting could be deep in the term.
- rewriting on terms:

$$x + S(y) \to S(x + y)$$

Deduction modulo: allowed rewriting

- General form (free variables are possible):

$$l \to r$$

- use: We replace $t = \sigma l$ by σr (unification). Rewriting could be deep in the term.
- rewriting on terms:

$$x + S(y) \to S(x + y)$$

- and on **propositions** (predicate symbols):

$$x * y = 0 \to x = 0 \vee y = 0$$

- advantage: expressiveness

Deduction modulo: allowed rewriting

- General form (free variables are possible):
$$l \to r$$
- use: We replace $t = \sigma l$ by σr (unification). Rewriting could be deep in the term.
- rewriting on terms:
$$x + S(y) \to S(x + y)$$
- and on **propositions** (predicate symbols):
$$x * y = 0 \to x = 0 \vee y = 0$$
- advantage: expressiveness
- we obtain a congruence modulo $\mathcal{R}$ (chosen set of rules): $\equiv$

Natural deduction modulo - first presentation

$$\frac{}{\Gamma, A \vdash A}\text{axiom}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \wedge\text{-i} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \wedge\text{-e1} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \wedge\text{-e2}$$

$$\Rightarrow\text{-i}\ \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \qquad \frac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B} \Rightarrow\text{-e}$$

$$\frac{\Gamma \vdash \forall x A[x]}{\Gamma \vdash A[t]}\forall\text{-e, any } t \qquad \frac{\Gamma \vdash A[x]}{\Gamma \vdash \forall x A[x]}\forall\text{-i, } x \text{ free}$$

Natural deduction modulo - first presentation

$$\frac{}{\Gamma, A \vdash A}\text{axiom}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \wedge\text{-i} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \wedge\text{-e1} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \wedge\text{-e2}$$

$$\Rightarrow\text{-i}\ \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \qquad \frac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B} \Rightarrow\text{-e}$$

$$\frac{\Gamma \vdash \forall x A[x]}{\Gamma \vdash A[t]} \forall\text{-e, any } t \qquad \frac{\Gamma \vdash A[x]}{\Gamma \vdash \forall x A[x]} \forall\text{-i, } x \text{ free}$$

- Add the following conversion rule

$$\frac{\Gamma \vdash A}{\Gamma \vdash B}\ A \equiv B$$

Natural deduction modulo, second version

$$\frac{}{\Gamma, A \vdash B}\text{axiom}, A \equiv B$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash C} \wedge\text{-i}, C \equiv A \wedge B \qquad \frac{\Gamma \vdash C}{\Gamma \vdash A} \wedge\text{-e1}, C \equiv A \wedge B \qquad \frac{\Gamma \vdash C}{\Gamma \vdash B} \wedge\text{-e2}, C \equiv A \wedge B$$

$$\Rightarrow\text{-i}, C \equiv A \wedge B \ \frac{\Gamma, A \vdash B}{\Gamma \vdash C} \qquad \frac{\Gamma \vdash C \qquad \Gamma \vdash A}{\Gamma \vdash B} \Rightarrow\text{-e}, C \equiv A \wedge B$$

$$\frac{\Gamma \vdash A[x]}{\Gamma \vdash B}\forall\text{-i}, x \text{ free}, B \equiv \forall x A[x] \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A[t]}\forall\text{-e, any } t, B \equiv \forall x A[x]$$

Example: 3

- consider the rewriting system $\mathcal{R}$:

$$\begin{array}{rcl} P(0) & \rightarrow & A \\ P(1) & \rightarrow & B \end{array}$$

$$\forall x P(x) \vdash A \wedge B$$

Example: 3

- consider the rewriting system $\mathcal{R}$:

$$\begin{aligned} P(0) &\to A \\ P(1) &\to B \end{aligned}$$

$$\frac{\forall x P(x) \vdash A \qquad \forall x P(x) \vdash B}{\forall x P(x) \vdash A \wedge B} \wedge\text{-i}$$

Example: 3

- consider the rewriting system $\mathcal{R}$:

$$\begin{array}{rcl} P(0) & \rightarrow & A \\ P(1) & \rightarrow & B \end{array}$$

$$\dfrac{\forall\text{-e}\ \dfrac{\forall x P(x) \vdash \forall x P(x)}{\forall x P(x) \vdash A} \qquad \dfrac{\forall x P(x) \vdash \forall x P(x)}{\forall x P(x) \vdash B}\ \forall\text{-e}}{\forall x P(x) \vdash A \wedge B}\ \wedge\text{-r}$$

Example: 3

- consider the rewriting system $\mathcal{R}$:

$$\begin{array}{rcl} P(0) & \to & A \\ P(1) & \to & B \end{array}$$

$$\dfrac{\forall\text{-e}\,\dfrac{\text{conv}\,\dfrac{\forall x P(x) \vdash \forall x P(x)}{\forall x P(x) \vdash P(0)}}{\forall x P(x) \vdash A} \qquad \dfrac{\dfrac{\forall x P(x) \vdash \forall x P(x)}{\forall x P(x) \vdash P(1)}\,\forall\text{-e}}{\forall x P(x) \vdash B}\,\text{conv}}{\forall x P(x) \vdash A \wedge B}\,\wedge\text{-r}$$

Example: 3

- consider the rewriting system $\mathcal{R}$:

$$
\begin{aligned}
P(0) &\rightarrow A \\
P(1) &\rightarrow B
\end{aligned}
$$

$$
\dfrac{\forall\text{-e}\ \dfrac{\forall x P(x) \vdash \forall x P(x)}{\forall x P(x) \vdash A} \qquad \dfrac{\forall x P(x) \vdash \forall x P(x)}{\forall x P(x) \vdash B}\ \forall\text{-e}}{\forall x P(x) \vdash A \wedge B}\ \wedge\text{-r}
$$

Example: 3

- consider the rewriting system $\mathcal{R}$:

$$
\begin{array}{rcl}
P(0) & \rightarrow & A \\
P(1) & \rightarrow & B
\end{array}
$$

$$
\dfrac{\dfrac{\dfrac{}{\forall x P(x) \vdash \forall x P(x)}\ \text{axiom}}{\forall x P(x) \vdash A}\ \forall\text{-e} \qquad \dfrac{\dfrac{}{\forall x P(x) \vdash \forall x P(x)}\ \text{axiom}}{\forall x P(x) \vdash B}\ \forall\text{-e}}{\forall x P(x) \vdash A \wedge B}\ \wedge\text{-r}
$$

A Cut: a detour

$$\dfrac{\Gamma \vdash A \qquad \dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\Rightarrow\text{-i}}{\Gamma \vdash B}\Rightarrow\text{-e}$$

- show $\Gamma \vdash A$ and $\Gamma, A \vdash B$
- then, you have showed $\Gamma \vdash B$
- it is the application of a lemma.

A Cut: a detour

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \qquad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B} \wedge\text{-i}}{\Gamma \vdash A} \wedge\text{-e}$$

General pattern of a cut: an introduction rule, followed by an elimination on the same symbol.
This is unnecessary, consider only π_1.

$$\dfrac{\pi_1}{\Gamma \vdash A}$$

A Cut: a detour

In deduction modulo:

$$\dfrac{\dfrac{\theta}{\Gamma \vdash A'} \qquad \dfrac{\dfrac{\pi}{\Gamma, A \vdash B}}{\Gamma \vdash C} \Rightarrow\text{-i}, C \equiv A \Rightarrow B}{\Gamma \vdash B'} \Rightarrow\text{-e}, C \equiv A' \Rightarrow B'$$

- need for cut elimination: the heart of logic.

A Cut: a detour

In deduction modulo:

$$\dfrac{\dfrac{\theta}{\Gamma \vdash A'} \qquad \dfrac{\dfrac{\pi}{\Gamma, A \vdash B}}{\Gamma \vdash C} \Rightarrow\text{-i},\ C \equiv A \Rightarrow B}{\Gamma \vdash B'} \Rightarrow\text{-e},\ C \equiv A' \Rightarrow B'$$

- need for cut elimination: the heart of logic.
- two main methods:
  - semantic: cut admissibility.
  - syntactic: proof normalization.

A Cut: a detour

In deduction modulo:

$$\dfrac{\dfrac{\theta}{\Gamma \vdash A'} \qquad \dfrac{\dfrac{\pi}{\Gamma, A \vdash B}}{\Gamma \vdash C} \Rightarrow\text{-i}, C \equiv A \Rightarrow B}{\Gamma \vdash B'} \Rightarrow\text{-e}, C \equiv A' \Rightarrow B'$$

- need for cut elimination: the heart of logic.
- two main methods:
  - semantic: cut admissibility.
  - syntactic: proof normalization.
- indecidable, need for conditions on $\mathcal{R}$.

II – The semantic method

The semantical method

$$
\begin{tikzcd}[column sep=huge]
\Gamma \vdash A \arrow[r, shift left, "\text{soundness}"] \arrow[d, "\begin{array}{c}\text{Gentzen}\\ \text{Tait-Girard}\\ \text{Dowek-Werner}\\ \ldots\end{array}"'] & \Gamma \models A \arrow[l, shift left, "\text{completeness}"] \\
\Gamma \vdash_{cf} A &
\end{tikzcd}
$$

The semantical method

$$
\begin{tikzcd}
\Gamma \vdash A \arrow[rr, "\text{soundness}"] \arrow[d, "\begin{array}{c}\text{Gentzen}\\ \text{Tait-Girard}\\ \text{Dowek-Werner}\\ \ldots\end{array}"'] & & \Gamma \models A \arrow[dll, "\text{strong completeness}"'] \\
\Gamma \vdash_{cf} A & &
\end{tikzcd}
$$

Heyting algebras

- a universe Ω
- an order

Heyting algebras

- a universe Ω
- an order
- operations on it: lowest upper bound (join: $\cup$), greatest lower bound (meet: $\cap$), arrow $\rightarrow$ (more that lattice).

$$
\begin{array}{ccc}
a \cap b \le a & a \cap b \le b & c \le a \text{ and } c \le b \text{ implies } c \le a \cap b \\
a \le a \cup b & b \le a \cup b & a \le c \text{ and } b \le c \text{ implies } a \cup b \le c \\
a \le b \rightarrow c & \text{iff} & a \cap b \le c
\end{array}
$$

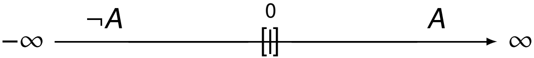
- like Boolean algebras, with weaker complement

an example

- $\mathbb{R}$ and open sets (infinite g.l.b. is not infinite intersection)

an example

- $\mathbb{R}$ and open sets (infinite g.l.b. is not infinite intersection)
- complement is weaker:



A model

- a domain $\mathcal{D}$ to interpret the first-order terms.
- a Heyting algebra Ω
- a interpretation function for each symbol:

$$\begin{array}{rcl} \hat{f} : \mathcal{D}^n & \rightarrow & \mathcal{D} \\ \hat{P} : \mathcal{D}^m & \rightarrow & \Omega \end{array}$$

- that we extend readily to all terms and all formulae and terms:

$$\begin{array}{rcl} (x)^*_\phi & := & \phi(x) \\ (f(t_1, \cdots, t_n))^*_\phi & := & \hat{f}(((t_1)^*_\phi, \cdots, (t_n)^*_\phi)) \\ (P(t_1, \cdots, t_n))^*_\phi & := & \hat{P}(((t_1)^*_\phi, \cdots, (t_n)^*_\phi)) \\ (A \wedge B)^*_\phi & := & (A)^*_\phi \cap (B)^*_\phi \end{array}$$

A model

- a domain $\mathcal{D}$ to interpret the first-order terms.
- a Heyting algebra Ω
- a interpretation function for each symbol:

$$\begin{array}{rcl} \hat{f} : \mathcal{D}^n & \to & \mathcal{D} \\ \hat{P} : \mathcal{D}^m & \to & \Omega \end{array}$$

- that we extend readily to all terms and all formulae and terms:

$$\begin{array}{rcl} (x)^*_\phi & := & \phi(x) \\ (f(t_1, \cdots, t_n))^*_\phi & := & \hat{f}(((t_1)^*_\phi, \cdots, (t_n)^*_\phi)) \\ (P(t_1, \cdots, t_n))^*_\phi & := & \hat{P}(((t_1)^*_\phi, \cdots, (t_n)^*_\phi)) \\ (A \wedge B)^*_\phi & := & (A)^*_\phi \cap (B)^*_\phi \end{array}$$

- degree of freedom: how to choose $\hat{f}$ and $\hat{P}$.
- in deduction modulo, additional condition:

$$A \equiv_{\mathcal{R}} B \text{ implies } A^* = B^*$$

Cannonical model: Lindenbaum algebra

- defined for provability
- elements of Ω: the equivalence class of formulae $[A]$.

$$[A] := \{B \mid \vdash A \Leftrightarrow B\}$$

- order: $[A] \leq [B]$ iff $\vdash A \Rightarrow B$ is provable
- meet: $[A] \cap [B]$ iff $[A \wedge B]$

Cannonical model: Lindenbaum algebra

- defined for provability
- elements of Ω: the equivalence class of formulae $[A]$.

$$[A] := \{B \mid \vdash A \Leftrightarrow B\}$$

- order: $[A] \leq [B]$ iff $\vdash A \Rightarrow B$ is provable
- meet: $[A] \cap [B]$ iff $[A \wedge B]$
- and so on ... (domain $\mathcal{D}$: open terms).
- with this model, one proves completeness

Cannonical model: Lindenbaum algebra

- defined for provability with cuts
- elements of Ω: the equivalence class of formulae $[A]$.

$$[A] := \{B \mid \vdash A \Leftrightarrow B\}$$

- "intersection": $[A] \cap [B]$ iff $[A \wedge B]$
- "order": $[A] \leq [B]$ iff $\vdash A \Rightarrow B$
- and so on ... (domain $\mathcal{D}$: open terms)
- with this model, one proves completeness: cuts are needed for transitivity of the order.

Cut-free cannonical model

- defined for provability without cuts
- elements of Ω: the contexts proving A cut-free.

$$[A] := \{\Gamma \mid \Gamma \vdash^* A\}$$

- the $[A]$ generate Ω with their (arbitrary) intersection and pseudo-union (l.u.b.):

$$a \cup b = \bigcap \{[A] \mid a \subseteq [A] \text{ and } b \subseteq [A]\}$$

- order: $a \leq b$ iff $a \subseteq b$
- and so on ...
- with this model, one proves cut-free completeness.

Deduction modulo

- what about the domain ?
- what about the validity of the rewrite rules ?

$$A \equiv_{\mathcal{R}} B \text{ implies } A^* = B^*$$

Deduction modulo

- what about the domain: it depends on $\mathcal{R}$ (not always open term).
- what about the validity of the rewrite rules: choose carefully the interpretation of predicates and function symbols, depends on $\mathcal{R}$.

An example: Simple Theory of Types

- aka higher-order (intuitionistic) logic.
- basic types o, ι, and arrow: $o \to o$, $o \to \iota$, ...
- constants of each type
- application $(t\ u)$ and λ-abstraction or combinators: S, K
- logical connectors: constants $\wedge : o \to o \to o$, ...
- e.g. we can form the formula: $\forall P.P$
- same deduction rules as NJ plus lambda-conversion.

Cut admissibility in STT

- problem number one, circularity:

$$\dfrac{\dfrac{\vdots}{\vdash \forall.P(P \Rightarrow P)}}{\vdash (\mathfrak{P} \Rightarrow \mathfrak{P})}$$

Cut admissibility in STT

- problem number one, circularity:

$$\dfrac{\dfrac{\vdots}{\vdash \forall.P(P \Rightarrow P)}}{\vdash (\forall P.(P \Rightarrow P) \Rightarrow \forall P.(P \Rightarrow P))}$$

- no more induction on the size of the formulae.

Cut admissibility in STT

- problem number one, circularity:

$$\dfrac{\dfrac{\vdots}{\vdash \forall.P(P \Rightarrow P)}}{\vdash (\forall P.(P \Rightarrow P) \Rightarrow \forall P.(P \Rightarrow P))}$$

- no more induction on the size of the formulae.
- solution, same as Girard:

Define R_A: quantify over all R_B: Circular

Avoid circularity: define C a priori, quantify over C instead,
Prove a posteriori that $R_B \in C$.

Cut admissibility in STT

- problem number one, circularity:

$$\dfrac{\dfrac{\vdots}{\vdash \forall.P(P \Rightarrow P)}}{\vdash (\forall P.(P \Rightarrow P) \Rightarrow \forall P.(P \Rightarrow P))}$$

- no more induction on the size of the formulae.
- solution, same as Girard:

> Define R_A: quantify over all R_B: Circular
>
> Avoid circularity: define C a priori, quantify over C instead,
> Prove a posteriori that $R_B \in C$.

- define "semantic candidates" [Okada] for $(A)^*$ without induction:

$$\{\alpha \in \Omega \mid A \in \alpha \subseteq [A]\}$$

Cut admissibility in STT

- problem number one, circularity:

$$\dfrac{\dfrac{\vdots}{\vdash \forall.P(P \Rightarrow P)}}{\vdash (\forall P.(P \Rightarrow P) \Rightarrow \forall P.(P \Rightarrow P))}$$

- no more induction on the size of the formulae.
- solution, same as Girard:

> Define R_A: quantify over all R_B: Circular
>
> Avoid circularity: define C a priori, quantify over C instead,
> Prove a posteriori that $R_B \in C$.

- define "semantic candidates" [Okada] for $(A)^*$ without induction:

$$\{\alpha \in \Omega \mid A \in \alpha \subseteq [A]\}$$

- then quantify over all truth-values candidates. Identifies which of the α is $(A)^*$.

Cut admissibility in STT

- Problem 2: logical intensionality. In STT, as in λProlog:

$$P(A \wedge A) \not\leftrightarrow P(A)$$

No logical extensionality rule:

$$\frac{P(A) \qquad A \Leftrightarrow B}{P(B)}$$

Cut admissibility in STT

- Problem 2: logical intensionality. In STT, as in λProlog:

$$P(A \wedge A) \not\leftrightarrow P(A)$$

  No logical extensionality rule:

$$\frac{P(A) \qquad A \Leftrightarrow B}{P(B)}$$

- implicates: although semantic truth value of A is in Ω, its domain of interpretation should not be Ω.

Cut admissibility in STT

- Problem 2: logical intensionality. In STT, as in λProlog:

$$P(A \wedge A) \not\leftrightarrow P(A)$$

  No logical extensionality rule:

$$\frac{P(A) \qquad A \Leftrightarrow B}{P(B)}$$

- implicates: although semantic truth value of A is in Ω, its domain of interpretation should not be Ω.
- usual trick:

$$\{\alpha \in \Omega \mid A \in \alpha \subseteq [A]\}$$

Cut admissibility in STT

- Problem 2: logical intensionality. In STT, as in λProlog:

$$P(A \wedge A) \not\leftrightarrow P(A)$$

No logical extensionality rule:

$$\frac{P(A) \qquad A \Leftrightarrow B}{P(B)}$$

- implicates: although semantic truth value of A is in Ω, its domain of interpretation should not be Ω.
- usual trick: pairing (V-complexes).

$$\mathrm{D}_o = \{\langle A, \alpha \rangle \mid A \in \alpha \subseteq [A]\}$$

Cut admissibility in STT

- Problem 2: logical intensionality. In STT, as in λProlog:

$$P(A \wedge A) \not\Leftrightarrow P(A)$$

  No logical extensionality rule:

$$\frac{P(A) \qquad A \Leftrightarrow B}{P(B)}$$

- implicates: although semantic truth value of A is in Ω, its domain of interpretation should not be Ω.
- usual trick: pairing (V-complexes).

$$\mathrm{D}_o = \{\langle A, \alpha\rangle \mid A \in \alpha \subseteq [A]\}$$

- interpret everything within those domains, e.g.:

$$\hat{\wedge} := \langle \wedge, \lambda\langle B, b\rangle.\langle \wedge \cdot B, \lambda\langle C, c\rangle.\langle \wedge \cdot B \cdot C, b \cap c\rangle\rangle\rangle$$

Cut admissibility in STT

- Problem 2: logical intensionality. In STT, as in λProlog:

$$P(A \wedge A) \not\leftrightarrow P(A)$$

  No logical extensionality rule:

$$\frac{P(A) \qquad A \Leftrightarrow B}{P(B)}$$

- implicates: although semantic truth value of A is in Ω, its domain of interpretation should not be Ω.
- usual trick: pairing (V-complexes).

$$\mathrm{D}_o = \{\langle A, \alpha\rangle \mid A \in \alpha \subseteq [A]\}$$

- interpret everything within those domains, e.g.:

$$\hat{\wedge} := \langle \wedge, \lambda\langle B, b\rangle.\langle \wedge \cdot B, \lambda\langle C, c\rangle.\langle \wedge \cdot B \cdot C, b \cap c\rangle\rangle\rangle$$

- then, "extract" the truth value:

$$\omega(A^*) = \pi_2(A^*)$$

STT in deduction modulo

- same types, same symbols $\dot{\wedge}, \dot{\forall}, \cdots$
- application:

$$\begin{array}{rcl} K \cdot x \cdot y & \to & x \\ S \cdot x \cdot y \cdot z & \to & (xz)(yz) \end{array}$$

- how to express $\forall P.P$ in a first-order setting ?

STT in deduction modulo

- same types, same symbols $\dot{\wedge}, \dot{\forall}, \cdots$
- application:

$$\begin{array}{rcl} K \cdot x \cdot y & \rightarrow & x \\ S \cdot x \cdot y \cdot z & \rightarrow & (xz)(yz) \end{array}$$

- how to express $\forall P.P$ in a first-order setting ?
- solution: embed P into $\varepsilon(P)$, and define:

$$\begin{array}{rcl} \varepsilon(\dot{\wedge} \cdot A \cdot B) & \rightarrow & \varepsilon(A) \wedge \varepsilon(B) \\ \varepsilon(\dot{\forall} A) & \rightarrow & \forall x.\varepsilon(Ax) \end{array}$$

STT in deduction modulo

- same types, same symbols $\dot{\wedge}, \dot{\forall}, \cdots$
- application:

$$\begin{aligned} K \cdot x \cdot y &\rightarrow x \\ S \cdot x \cdot y \cdot z &\rightarrow (xz)(yz) \end{aligned}$$

- how to express $\forall P.P$ in a first-order setting ?
- solution: embed P into $\varepsilon(P)$, and define:

$$\begin{aligned} \varepsilon(\dot{\wedge} \cdot A \cdot B) &\rightarrow \varepsilon(A) \wedge \varepsilon(B) \\ \varepsilon(\dot{\forall} A) &\rightarrow \forall x.\varepsilon(Ax) \end{aligned}$$

- duplication of “connectors”: $\dot{\wedge}$ (of the type hierarchy) connecting terms and $\wedge$, connecting propositions.
- two “formulae”: P, a term, and $\varepsilon(P)$, at the logical level.
- ε is the only predicate symbol.

STT in deduction modulo

- same types, same symbols $\dot\wedge, \dot\forall, \cdots$
- application:

$$\begin{array}{rcl} K \cdot x \cdot y & \rightarrow & x \\ S \cdot x \cdot y \cdot z & \rightarrow & (xz)(yz) \end{array}$$

- how to express $\forall P.P$ in a first-order setting ?
- solution: embed P into $\varepsilon(P)$, and define:

$$\begin{array}{rcl} \varepsilon(\dot\wedge \cdot A \cdot B) & \rightarrow & \varepsilon(A) \wedge \varepsilon(B) \\ \varepsilon(\dot\forall A) & \rightarrow & \forall x.\varepsilon(Ax) \end{array}$$

- duplication of “connectors”: $\dot\wedge$ (of the type hierarchy) connecting terms and $\wedge$, connecting propositions.
- two “formulae”: P, a term, and $\varepsilon(P)$, at the logical level.
- ε is the only predicate symbol.
- ε embeds in the syntax the ω is in the semantics: separates truth value and propositional content.

III - Normalization

Curry-Howard correspondence

- Notation for proofs. Give a name to each of the hypothesis:

$$\Gamma = x_1 : A_1, \ldots, x_n : A_n$$

$$\frac{}{\Gamma, x : A \vdash x : A}\ \text{Axiom} \qquad \frac{\Gamma \vdash \pi : A \wedge B}{\Gamma \vdash \mathit{fst}(\pi) : A}\ \wedge\text{-e1}$$

$$\frac{\Gamma \vdash \pi_1 : A \qquad \Gamma \vdash \pi_2 : B}{\Gamma \vdash \langle \pi_1, \pi_2 \rangle : A \wedge B}\ \wedge\text{-i} \qquad \frac{\Gamma \vdash \pi : A \wedge B}{\Gamma \vdash \mathit{snd}(\pi) : A}\ \wedge\text{-e2}$$

$$\frac{\Gamma, x : A \vdash \pi : B}{\Gamma \vdash \lambda x.\pi : A \Rightarrow B}\ \Rightarrow\text{-i} \qquad \frac{\Gamma \vdash \pi' : A \qquad \Gamma \vdash \pi : A \Rightarrow B}{\Gamma \vdash (\pi\pi') : B}$$

Curry-Howard correspondence

- Notation for proofs. Give a name to each of the hypothesis:

$$\Gamma = x_1 : A_1, \ldots, x_n : A_n$$

$$\frac{}{\Gamma, x : A \vdash x : A}\ \text{Axiom} \qquad \frac{\Gamma \vdash \pi : A \wedge B}{\Gamma \vdash \mathit{fst}(\pi) : A}\ \wedge\text{-e1}$$

$$\frac{\Gamma \vdash \pi_1 : A \qquad \Gamma \vdash \pi_2 : B}{\Gamma \vdash \langle \pi_1, \pi_2 \rangle : A \wedge B}\ \wedge\text{-i} \qquad \frac{\Gamma \vdash \pi : A \wedge B}{\Gamma \vdash \mathit{snd}(\pi) : A}\ \wedge\text{-e2}$$

$$\frac{\Gamma, x : A \vdash \pi : B}{\Gamma \vdash \lambda x.\pi : A \Rightarrow B}\ \Rightarrow\text{-i} \qquad \frac{\Gamma \vdash \pi' : A \qquad \Gamma \vdash \pi : A \Rightarrow B}{\Gamma \vdash (\pi\pi') : B}$$

- **very** similar to a type system

Curry-Howard correspondence

- Notation for proofs. Give a name to each of the hypothesis:

$$\Gamma = x_1 : A_1, \ldots, x_n : A_n$$

$$\frac{}{\Gamma, x : A \vdash x : A}\ \text{Axiom} \qquad \frac{\Gamma \vdash \pi : A \wedge B}{\Gamma \vdash fst(\pi) : A}\ \wedge\text{-e1}$$

$$\frac{\Gamma \vdash \pi_1 : A \qquad \Gamma \vdash \pi_2 : B}{\Gamma \vdash \langle \pi_1, \pi_2 \rangle : A \wedge B}\ \wedge\text{-i} \qquad \frac{\Gamma \vdash \pi : A \wedge B}{\Gamma \vdash snd(\pi) : A}\ \wedge\text{-e2}$$

$$\frac{\Gamma, x : A \vdash \pi : B}{\Gamma \vdash \lambda x.\pi : A \Rightarrow B}\ \Rightarrow\text{-i} \qquad \frac{\Gamma \vdash \pi' : A \qquad \Gamma \vdash \pi : A \Rightarrow B}{\Gamma \vdash (\pi\pi') : B}$$

- **very** similar to a type system
- in deduction modulo, rewrite rules are silent:

$$\frac{\Gamma \vdash \pi : A}{\Gamma \vdash \pi : B}\ A \equiv B$$

Cut elimination with proof terms

- Cut elimination is a **process**, similar to function execution.

$$\dfrac{\dfrac{\Gamma \vdash \pi_1 : A \qquad \Gamma \vdash \pi_2 : B}{\Gamma \vdash \langle \pi_1, \pi_2 \rangle : A \wedge B}\ \wedge\text{-i}}{\Gamma \vdash \mathit{fst}(\langle \pi_1, \pi_2 \rangle) : A}\ \wedge\text{-e} \quad \rhd \quad \Gamma \vdash \pi_1 : A$$

$$\dfrac{\Gamma \vdash \theta : A \qquad \dfrac{\Gamma, x : A \vdash \pi : B}{\Gamma \vdash \lambda x.\pi : A \Rightarrow B}\ \Rightarrow\text{-i}}{\Gamma \vdash (\lambda x.\pi)\theta : B}\ \Rightarrow\text{-e} \quad \rhd \quad \Gamma \vdash \{\theta / x\}\pi : B$$

Cut elimination with proof terms

- Cut elimination is a **process**, similar to function execution.

$$\dfrac{\dfrac{\Gamma \vdash \pi_1 : A \qquad \Gamma \vdash \pi_2 : B}{\Gamma \vdash \langle \pi_1, \pi_2 \rangle : A \wedge B}\wedge\text{-i}}{\Gamma \vdash \mathit{fst}(\langle \pi_1, \pi_2 \rangle) : A}\wedge\text{-e} \quad \rhd \quad \Gamma \vdash \pi_1 : A$$

$$\dfrac{\Gamma \vdash \theta : A \qquad \dfrac{\Gamma, x : A \vdash \pi : B}{\Gamma \vdash \lambda x.\pi : A \Rightarrow B}\Rightarrow\text{-i}}{\Gamma \vdash (\lambda x.\pi)\theta : B}\Rightarrow\text{-e} \quad \rhd \quad \Gamma \vdash \{\theta/x\}\pi : B$$

- showing that every proof normalizes: the cut elimination process terminates.

Normalization [Dowek,Werner]

- deduction modulo is high-level: circularity hence reducibility candidates.

Normalization [Dowek,Werner]

- deduction modulo is high-level: circularity hence reducibility candidates.
- A reducibility candidate: a set of normalizing proof terms (and other closure properties).

Normalization [Dowek,Werner]

- deduction modulo is high-level: circularity hence reducibility candidates.
- A reducibility candidate: a set of normalizing proof terms (and other closure properties).
- to each formula A, associates a candidate $[\![A]\!]$: this is a $\mathcal{C}$-valued model (pre-model).

Normalization [Dowek,Werner]

- deduction modulo is high-level: circularity hence reducibility candidates.
- A reducibility candidate: a set of normalizing proof terms (and other closure properties).
- to each formula A, associates a candidate $[\![A]\!]$: this is a $\mathcal{C}$-valued model (pre-model).
- in deduction modulo, if $A \equiv B$, additional constraint:

$$[\![A]\!] = [\![B]\!]$$

Normalization [Dowek,Werner]

- deduction modulo is high-level: circularity hence reducibility candidates.
- A reducibility candidate: a set of normalizing proof terms (and other closure properties).
- to each formula A, associates a candidate $[\![A]\!]$: this is a $\mathcal{C}$-valued model (pre-model).
- in deduction modulo, if $A \equiv B$, additional constraint:

$$[\![A]\!] = [\![B]\!]$$

- then prove the main theorem:

Theorem: if $\Gamma \vdash \pi : A$ then for any ψ substitution, ϕ model assignment, θ environment (mapping $\alpha : B \in \Gamma$ to $[\![A]\!]_\phi$), we have $\theta\psi\pi \in [\![A]\!]_\phi$

IV – From Normalization to usual semantics

- such methods are defined in deduction modulo (Heyting arithmetic, higher-order logic, Zermelo's set theory, ...)

- such methods are defined in deduction modulo (Heyting arithmetic, higher-order logic, Zermelo's set theory, ...)
- the pre-model have a structure: pseudo Heyting algebras, or truth value algebras (TVA) [Dowek].

Heyting algebras

- a universe Ω
- an order
- operations on it: lowest upper bound (join: $\cup$), greatest lower bound (meet: $\cap$ – intersection).

$$a \cap b \leq a \quad a \cap b \leq b \quad c \leq a \text{ and } c \leq b \text{ implies } c \leq a \cap b$$
$$a \leq a \cup b \quad b \leq a \cup b \quad a \leq c \text{ and } b \leq c \text{ implies } a \cup b \leq c$$

- like Boolean algebras, with weaker complement

pseudo-Heyting algebras, aka Truth Values Algebras

- a universe Ω
- a pre-order: $a \leq b$ and $b \leq a$ with $a \neq b$ possible.
- operations on it: lowest upper bound (join: $\cup$ – pseudo union), greatest lower bound (meet: $\cap$ – intersection).

$$a \cap b \leq a \quad a \cap b \leq b \quad c \leq a \text{ and } c \leq b \text{ implies } c \leq a \cap b$$

$$a \leq a \cup b \quad b \leq a \cup b \quad a \leq c \text{ and } b \leq c \text{ implies } a \cup b \leq c$$

Candidates form a pseudo-Heyting algebra

- $\top = \bot = \mathcal{SN}$
- $[\![A]\!] \cap [\![B]\!] = [\![A \wedge B]\!]$
- and so on.
- pre-order: trivial one.
- But $[\![A \wedge A]\!] \leq\geq [\![A]\!]$ only.
- of course:
$$A \equiv B \text{ implies } [\![A]\!] = [\![B]\!]$$

Super consistency

- the pre-model construction (domain, ...) does not depends on the properties of $\mathcal{C}$.
- consistency: there exists a model.

- condition in DM: $A \equiv B$ implies $[\![A]\!] = [\![B]\!]$

Super consistency

- the pre-model construction (domain, ...) does not depends on the properties of $\mathcal{C}$.
- consistency: there exists a model.
- super-consistency: for every TVA, there exists a model (interpretation): construction has to be uniform.
- condition in DM: $A \equiv B$ implies $[\![A]\!] = [\![B]\!]$

Super consistency

- the pre-model construction (domain, ...) does not depends on the properties of C.
- consistency: there exists a model.
- super-consistency: for every TVA, there exists a model (interpretation): construction has to be uniform.
- condition in DM: $A \equiv B$ implies $[\![A]\!] = [\![B]\!]$
- Super consistency implies cut elimination.

Super consistency

- e.g. higher-order logic is super-consistent:

$$
\begin{aligned}
M_\iota &= \iota \text{ (dummy)} \\
M_o &= \mathcal{C} \\
M_{t \to u} &= M_u^{M_t}
\end{aligned}
$$

Super consistency

- e.g. higher-order logic is super-consistent:

$$\begin{array}{rcl} M_\iota & = & \iota \text{ (dummy)} \\ M_o & = & \mathcal{C} \\ M_{t\to u} & = & M_u^{M_t} \end{array}$$

- hence, it has a model in the pseudo-Heying Algebra of candidates
- $\Gamma \vdash \pi : A$ implies $\pi \in [\![A]\!]$.
- the system enjoys proof normalization.

Towards usual semantics

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\Gamma \vdash A \arrow[r, "\text{soundness}"] \arrow[d, swap, "\begin{array}{c}\text{Gentzen}\\ \text{Tait-Girard}\\ \text{Dowek-Werner}\\ \ldots\end{array}"] & \Gamma \models A \arrow[dl, "\text{strong completeness}"] \\
\Gamma \vdash_{cf} A &
\end{tikzcd}
$$

Super consistency

- e.g. higher-order logic is super-consistent:

$$
\begin{array}{rcl}
M_\iota & = & \iota \text{ (dummy)} \\
M_o & = & \mathcal{C} \\
M_{t \to u} & = & M_u^{M_t}
\end{array}
$$

Super consistency

- e.g. higher-order logic is super-consistent:

$$\begin{array}{rcl} M_\iota & = & \iota \text{ (dummy)} \\ M_o & = & \mathcal{C} \\ M_{t \to u} & = & M_u^{M_t} \end{array}$$

- hence, it has a model in the pseudo-Heying Algebra of reducibility candidates

$$[\![A]\!] = \{\pi \text{ such that ...}\}$$

- but, $[\![\top]\!] \wedge [\![\top]\!] \neq [\![\top]\!]$

Towards usual semantics

- How to transform a TVA into a Heyting algebra.
- assume we have a model $\mathcal{M}, [\![\_]\!]$ in the previous pseudo-Heyting algebra of sequents.
- first idea: pseudo-Heyting to Heyting by quotienting.

Towards usual semantics

- How to transform a TVA into a Heyting algebra.
- assume we have a model $\mathcal{M}$, $[\![\_]\!]$ in the previous pseudo-Heyting algebra of sequents.
- first idea: pseudo-Heyting to Heyting by quotienting.
- trivial pseudo order implies $\top = \bot$.

The link: extract contexts

- Assumption: we have a pre-model ($[\![A]\!]_\phi$, model $\mathcal{M}$ defined).
  Set:

$$[A]^\sigma_\phi \quad = \quad \{\Gamma \mid \Gamma \vdash \pi : \sigma A, \text{ and for any environment } \theta, \text{ assignment } \psi, \\ \theta\psi\pi \in [\![A]\!]_\phi\}$$

The link: extract contexts

- Assumption: we have a pre-model ($[\![A]\!]_\phi$, model $\mathcal{M}$ defined). Set:

$$[A]_\phi^\sigma = \{\Gamma \mid \Gamma \vdash \pi : \sigma A, \text{ and for any environment } \theta, \text{ assignment } \psi, \theta\psi\pi \in [\![A]\!]_\phi\}$$

- $[\![A]\!]_\phi$ contains proof terms associated to $\Delta \vdash \pi : B$. Extract the contexts corresponding to A.
- this forms a Heyting algebra ($[A]$: basis)

The link: extract contexts

- Assumption: we have a pre-model ($[\![A]\!]_\phi$, model $\mathcal{M}$ defined). Set:

$$[A]^\sigma_\phi = \{\Gamma \mid \Gamma \vdash \pi : \sigma A, \text{ and for any environment } \theta, \text{ assignment } \psi, \theta\psi\pi \in [\![A]\!]_\phi\}$$

- $[\![A]\!]_\phi$ contains proof terms associated to $\Delta \vdash \pi : B$. Extract the contexts corresponding to A.
- this forms a Heyting algebra ($[A]$: basis)
- interpretation of formulas in it:

$$A^* = [A]^\sigma_\phi$$

Wait a minute !

- interpretation ? $[A]^{\sigma}_{\phi}$.

Wait a minute !

- interpretation ? $[A]^{\sigma}_{\phi}$.
- Need for *one single* substitution. hybridization: $\sigma \times \phi$.

$$D = \mathcal{T} \times M$$

Wait a minute !

- interpretation ? $[A]^{\sigma}_{\phi}$.
- Need for *one single* substitution. hybridization: $\sigma \times \phi$.

$$D = \mathcal{T} \times M$$

- interpretation for symbols were, in $\mathcal{C}$:

$$\hat{f}^{\mathcal{M}}(d_1, ..., d_n) \in M \qquad \hat{P}^{\mathcal{M}}(d_1, ..., d_n) \in \mathcal{C}$$

Wait a minute !

- interpretation ? $[A]^{\sigma}_{\phi}$.
- Need for *one single* substitution. hybridization: $\sigma \times \phi$.

$$D = \mathcal{T} \times M$$

- interpretation for symbols were, in $\mathcal{C}$:

$$\hat{f}^{\mathcal{M}}(d_1, ..., d_n) \in M \qquad \hat{P}^{\mathcal{M}}(d_1, ..., d_n) \in \mathcal{C}$$

- Now they are:

$$\begin{array}{rcl} \hat{f}^{\mathcal{D}}(\langle t_1, d_1 \rangle, ..., \langle t_n, d_n \rangle) & = & \langle f(t_1, ..., t_n), \hat{f}^{\mathcal{M}}(d_1, ..., d_n) \rangle \\ \hat{P}^{\mathcal{D}}(\langle t_1, d_1 \rangle, ..., \langle t_n, d_n \rangle) & = & [P]^{(t_1/x_1, ..., t_n/x_n)}_{(d_1/x_1, ..., d_n/x_n)} \\ & = & \{\Gamma \mid (\Gamma \vdash \pi : P(\overrightarrow{t})) \in [\![P]\!]_{(\overrightarrow{d}/\overrightarrow{x})}\} \end{array}$$

Wait a minute !

- interpretation ? $[A]_{\phi}^{\sigma}$.
- Need for *one single* substitution. hybridization: $\sigma \times \phi$.

$$D = \mathcal{T} \times M$$

- interpretation for symbols were, in $\mathcal{C}$:

$$\hat{f}^{\mathcal{M}}(d_1, ..., d_n) \in M \qquad \hat{P}^{\mathcal{M}}(d_1, ..., d_n) \in \mathcal{C}$$

- Now they are:

$$\begin{aligned} \hat{f}^{\mathcal{D}}(\langle t_1, d_1\rangle, ..., \langle t_n, d_n\rangle) &= \langle f(t_1, ..., t_n), \hat{f}^{\mathcal{M}}(d_1, ..., d_n)\rangle \\ \hat{P}^{\mathcal{D}}(\langle t_1, d_1\rangle, ..., \langle t_n, d_n\rangle) &= [P]_{(d_1/x_1,...,d_n/x_n)}^{(t_1/x_1,...,t_n/x_n)} \\ &= \{\Gamma \mid (\Gamma \vdash \pi : P(\overrightarrow{t})) \in [\![P]\!]_{(\vec{d}/\vec{x})}\} \end{aligned}$$

- Holds for any theory in DM. extends the V-complexes.
- pointwise application

$$\langle t, v\rangle \odot \langle t', v'\rangle = \langle (tt'), (vv')\rangle$$

instead of $\langle t, v\rangle \odot \langle t', v'\rangle = \langle (tt'), (v(\langle t', v'\rangle))\rangle$

- Need to prove $[A \wedge B] = [A] \cap [B]$ to have a model interpretation.
  Usually (semantic cut elim), only:

$$A \wedge B \in [A] \cap [B] \subset [A \wedge B]$$

- Need to prove $[A \wedge B] = [A] \cap [B]$ to have a model interpretation.
  Usually (semantic cut elim), only:

$$A \wedge B \in [A] \cap [B] \subset [A \wedge B]$$

- proof resembles the proof for normalization.

Cut admissibility

Assume $\Gamma \vdash A$ has a proof (with cuts)

- $[\Gamma] \leq [A]$ in $\mathcal{D}$ by (usual) soundness
- $\Gamma \in [\Gamma]$
- $\Gamma \in [A]$ implies $\Gamma \vdash_{cf} A$
- Q.E.D.

Cut admissibility

Assume $\Gamma \vdash A$ has a proof (with cuts)

- $[\Gamma] \leq [A]$ in $\mathcal{D}$ by (usual) soundness
- $\Gamma \in [\Gamma]$
- $\Gamma \in [A]$ implies $\Gamma \vdash_{cf} A$
- Q.E.D.
- compared to the former main lemma: $\Gamma \vdash \pi : A$ implies $\pi \in [\![A]\!]$, and hence π is $\mathcal{SN}$.

$$\begin{tikzcd}[column sep=huge, row sep=large]
\Gamma \vdash A \arrow[rr, "\text{soundness}"] \arrow[d, swap, "\begin{array}{c}\text{Gentzen}\\ \text{Tait-Girard}\\ \text{Dowek-Werner}\\ \dots\end{array}"] & & \Gamma \models A \arrow[dll, swap, "\text{strong completeness}"] \\
\Gamma \vdash_{cf} A & &
\end{tikzcd}$$

- This diagram does not commute in deduction modulo.

Further work

- what is the computational content of this algorithm ?
- there is normalization by evaluation work, but in a Kripke style: links ?
- do the proof terms (candidates) always have a “pseudo-” structure ?
- realizing rewrite rule not with $\lambda x.x$ (not silently), could recover (some) normalization and make the previous diagram commute again.

$$\frac{\Gamma \vdash \pi : A}{\Gamma \vdash \pi : B}\ A \equiv B$$