A computational look at soundness, completeness and reducibility

Hugo Herbelin

Journée de rentrée PPS

22 September 2014

Context

- The Curry-Howard correspondence suggests to think of proofs as programs
- With proof assistants, we tend to see models or semantics (functions, predicates, ...) as syntax of the meta-language
- With this respect, what does a proof of normalisation or a proof of soundness, or a proof of completeness say?

Outline

- Semantic normalisability via soundness and completeness
- Reflecting semantic normalisability: Normalisation-by-Evaluation
- A comparison with normalisation by reducibility
- The case of Tarskian semantics (Gödel’s completeness)

Simply-typed λ-calculus for minimal negative propositional logic

$$\begin{array}{rcl} A & ::= & X \mid A \dot{\rightarrow} A \\ \Gamma & ::= & \emptyset \mid \Gamma, A \end{array}$$

$$\frac{}{\Gamma, A, A_1, ..., A_n \vdash A}\ \mathsf{Var}_n$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \dot{\rightarrow} B}\ \mathsf{Lam} \qquad \frac{\Gamma \vdash A \dot{\rightarrow} B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ \mathsf{App}$$

Weakening is admissible:

$$\frac{\Gamma \subset \Gamma' \qquad \Gamma \vdash A}{\Gamma' \vdash A}\ \mathsf{Weak}$$

We write $\Gamma \vdash_{nf} A$ for a normal derivation and $\Gamma \vdash_{neut} A$ for a neutral derivation, i.e. a normal derivation made of an iteration of App over Var.

Semantic normalisability

Assume we have a sound and *strongly* complete notion of validity in minimal propositional logic w.r.t some kinds of model (Kripke model, Tarskian model, phase/pretopology semantics, ...)

$sound : \mathcal{T} \vdash A \Rightarrow \mathcal{T} \vDash A$

$compl : \mathcal{T} \vDash A \Rightarrow \mathcal{T} \vdash_{nf} A$

Then, composing soundness and strong completeness gives normalisability (C. Coquand 1993, Okada 2002, ...)

Applicable to intuitionistic logic, classical logic by taking for $\mathcal{T}$ all instances of ex falso quodlibet and all instances of double-negation elimination respectively.

An approach popular from the 90's: Normalisation-by-Evaluation

Let $\pi : \mathcal{T} \vdash A$, then $compl(sound(\pi)) : \mathcal{T} \vdash_{nf} A$

If the meta-theory has cut-elimination, then, for any π, $compl(sound(\pi))$ can be turned into some normal proof π', which is determined by how cut-elimination is implemented.

E.g., if the meta-theory is based on λ-calculus, the computational content is easy to observe:

- Soundness maps π to its interpretation in the model
- Strong completeness depends on the model:
  - E.g., for Tarskian semantics, the situation is delicate.
  - However, for Kripke/Beth (C. Coquand 1993) or phase/pretopology semantics (Okada 2002), completeness can be done quite syntactically so that π' is the β-normal η-long normal form of π.

An approach popular from the 90's: Normalisation-by-Evaluation

Normalisation-by-Evaluation relies on that the latter can itself be proved:

- If the meta-theory is type-theoretic, statements about proofs can be expressed and the fact that π' is related to π can be proved explicitly.
- Even the meta-theory is not type-theoretic but has enough higher-order functions, a proof of validity can be lifted at the level of a functional object which we can talk about.

Soundness in the case of Kripke semantics

Let $(\mathcal{K}, \geq, \Vdash_X, \mathsf{mon}_X)$ be a Kripke model.

We extend $\Vdash_X$ to all formulae:

$$\begin{array}{lcl} w \Vdash_{\mathcal{K}} X & \triangleq & w \Vdash_X \\ w \Vdash_{\mathcal{K}} A \dot{\rightarrow} B & \triangleq & \forall w' \geq w\, (w' \Vdash_{\mathcal{K}} A \Rightarrow w' \Vdash_{\mathcal{K}} B) \end{array}$$

We show that mon_X extends to to all formulae:

$$\mathsf{mon}_A(h) : w \Vdash_{\mathcal{K}} A \Rightarrow w' \Vdash_{\mathcal{K}} A$$

whenever $h : w \leq w'$.

We also write $\mathsf{refl}_w : w \geq w$.

Soundness in the case of Kripke semantics

For simplicity, we restrict ourselves to finite theories, written Γ.

We write $\Gamma \vDash_{\mathcal{K}} A \triangleq \forall w \, (w \Vdash_{\mathcal{K}} \Gamma \Rightarrow w \Vdash_{\mathcal{K}} A)$

We write $\Gamma \vDash A \triangleq \forall (\mathcal{K}, \geq, \Vdash_X, \mathsf{mon}_X) \, \Gamma \vDash_{\mathcal{K}} A$

We map syntax to semantics by induction on the derivation:

$$
\begin{array}{lcl}
[\![\ ]\!]_{\mathcal{K}} \ : \ \ \Gamma \vdash A & \Rightarrow & \Gamma \vDash_{\mathcal{K}} A \\
[\![\mathsf{Var}_n]\!]_{\mathcal{K}} & \triangleq & \lambda w.\lambda \vec{\alpha}.\alpha_n \\
[\![\mathsf{Lam}(\pi)]\!]_{\mathcal{K}} & \triangleq & \lambda w.\lambda \vec{\alpha}.\lambda w'.\lambda h.\lambda \alpha.[\![\pi]\!]_{\mathcal{K}} w' (\mathsf{mon}_{\Gamma}(h)(\vec{\alpha}), \alpha') \\
[\![\mathsf{App}(\pi, \pi')]\!]_{\mathcal{K}} & \triangleq & \lambda w.\lambda \vec{\alpha}.[\![\pi]\!]_{\mathcal{K}} w \vec{\alpha} \; w \; \mathsf{refl}_w \; ([\![\pi']\!]_{\mathcal{K}} w \vec{\alpha})
\end{array}
$$

$$
\begin{array}{lcl}
\mathit{sound} \ : \ \Gamma \vdash A & \Rightarrow & \Gamma \vDash A \\
\mathit{sound} \qquad \pi & \triangleq & \lambda (\mathcal{K}, \geq, \Vdash_X, \mathsf{mon}_X).\, [\![\pi]\!]_{\mathcal{K}}
\end{array}
$$

Completeness in the case of Kripke semantics

We use the universal model of contexts $\mathcal{U}$:

$$
\begin{array}{lll}
\mathcal{K} & \triangleq & \text{Contexts} \\
\Gamma \Vdash_X & \triangleq & \Gamma \vdash_{nf} X
\end{array}
\qquad
\begin{array}{lll}
\Gamma \leq \Gamma' & \triangleq & \Gamma \subset \Gamma' \\
\text{mon}_X & \triangleq & \text{Weak}
\end{array}
$$

We write $\text{shift}_{\Gamma,A} : \Gamma \subset (\Gamma, A)$.

We map semantics to syntax back and forth by induction on the type:

$$
\begin{array}{lll}
\uparrow^{\Gamma}_{A} & \Gamma \vdash_{neut} A & \Rightarrow \ \Gamma \Vdash_{\mathcal{U}} A \\
\uparrow^{\Gamma}_{X} & \pi & \triangleq \ \pi \\
\uparrow^{\Gamma}_{A \dot{\to} B} & \pi & \triangleq \ \lambda\Gamma'.\lambda h.\lambda\alpha.\uparrow^{\Gamma'}_{B} (\text{App}(\text{Weak}(h,\pi), \downarrow^{\Gamma'}_{A} \alpha))
\end{array}
$$

$$
\begin{array}{lll}
\downarrow^{\Gamma}_{A} & \Gamma \Vdash_{\mathcal{U}} A & \Rightarrow \ \Gamma \vdash_{nf} A \\
\downarrow^{\Gamma}_{X} & \alpha & \triangleq \ \alpha \\
\downarrow^{\Gamma}_{A \dot{\to} B} & \alpha & \triangleq \ \text{Lam}(\downarrow^{\Gamma,A}_{B} (\alpha\, (\Gamma, A)\, \text{shift}_{\Gamma,A}\, (\uparrow^{\Gamma,A}_{A} \text{Var}_0)))
\end{array}
$$

Hence: $compl\ v \triangleq \downarrow^{\Gamma}_{A} (v\, \mathcal{U}\, \Gamma\, (\uparrow^{\Gamma}_{\Gamma} \text{Var}_{\text{i}}))$

Normalisation-by-Evaluation with Kripke models (C. Coquand)

We shortcut the generalisation over all models and take instead

$$\begin{array}{lcl} sound & : & \Gamma \vdash A \Rightarrow \Gamma \vDash_{\mathcal{U}} A \\ compl & : & \Gamma \vDash_{\mathcal{U}} A \Rightarrow \Gamma \vdash_{nf} A \end{array}$$

We define $\alpha \sim_A^\Gamma \pi$ by induction on A with iteration on Γ.

We prove: $\forall \Gamma A \forall \pi : \Gamma \vdash A \; (sound(\pi) \sim_A^\Gamma \pi)$

We prove: $\forall \Gamma A \forall \pi : \Gamma \vdash A \; \forall \alpha : \Gamma \Vdash A \; (\alpha \sim_A^\Gamma \pi \Rightarrow compl(\alpha) =_{\beta\eta} \pi)$

Hence $compl(sound(\pi)) =_{\beta\eta} \pi$

Normalisation by reducibility

Let us first annotate derivations with proof-terms.

$$p, q \ ::= \ a \mid p\ q \mid \dot{\lambda} a.p$$

$$\Gamma \ ::= \ \emptyset \mid \Gamma, a : A$$

$$\frac{}{\Gamma, a : A, a_1 : A_1, ..., a_n : A_n \vdash a : A} \ \mathsf{Var}_n^a$$

$$\frac{\Gamma, a : A \vdash p : B}{\Gamma \vdash \dot{\lambda} a.p : A \dot{\rightarrow} B} \ \mathsf{Lam}^{\lambda a.p} \qquad \frac{\Gamma \vdash p : A \dot{\rightarrow} B \qquad \Gamma \vdash q : A}{\Gamma \vdash p\ q : B} \ \mathsf{App}^{pq}$$

Reduction is taken to be β-reduction and η-expansion

$$\frac{t \to_{\beta\eta} t'}{t\,u \to_{\beta\eta} t'\,u} \qquad \frac{u \to_{\beta\eta} u'}{t\,u \to_{\beta\eta} t\,u'} \qquad \frac{t \to_{\beta\eta} t'}{\lambda x.t \to_{\beta\eta} \lambda x.t'}$$

$$\frac{}{(\lambda x.t)u \to_{\beta\eta} t[x := u]} \qquad \frac{x \text{ fresh in } t}{t \to_{\beta\eta} \lambda x.(t\,x)}$$

Reducibility/realisability semantics

A set X is closed by anti-reduction if $p[x := q]\vec{r} \in X \Rightarrow (\lambda x.p)q\vec{r} \in X$.

Let an untyped reducibility/realisability model be an assignment ρ of atoms to sets closed by anti-reduction.

We define:

$$
\begin{array}{lcl}
p \;\mathtt{r}\; X & \triangleq & p \in \rho(X) \\
p \;\mathtt{r}\; A \dot{\to} B & \triangleq & \forall q\,(q \;\mathtt{r}\; A \;\Rightarrow\; p\,q \;\mathtt{r}\; B) \\
\\
\overrightarrow{a : B} \vdash p \;\mathtt{r}\; A & \triangleq & \forall \vec{q}\,(\overrightarrow{q \;\mathtt{r}\; B} \Rightarrow p[\overrightarrow{a := q}] \;\mathtt{r}\; A)
\end{array}
$$

Anti-reduction scales to arbitrary type:

$$\text{anti} : p[x := q]\vec{r} \;\mathtt{r}\; A \Rightarrow (\lambda x.p)q\vec{r} \;\mathtt{r}\; A$$

Normalisation by reducibility: adequacy

Soundness/Adequacy: $\forall \Gamma A\ \forall p\ (\Gamma \vdash p : A \Rightarrow \Gamma \vdash p\ \mathtt{r}\ A)$

Proof by induction on the derivation:

$$\begin{array}{lcl}
[\![\ ]\!]^p \ : \ \Gamma \vdash p : A & \Rightarrow & \Gamma \vdash p\ \mathtt{r}\ A \\
[\![\mathsf{Var}_n]\!]^a & \triangleq & \lambda\vec{q}.\lambda\vec{\alpha}.\alpha_n \\
[\![\mathsf{Lam}(\pi)]\!]^{\lambda a.p} & \triangleq & \lambda\vec{q}.\lambda\vec{\alpha}.\lambda q'.\lambda\alpha'.\mathsf{anti}([\![\pi]\!]^p(\vec{q}q')(\vec{\alpha}\alpha')) \\
[\![\mathsf{App}(\pi,\pi')]\!]^{pp'} & \triangleq & \lambda\vec{q}.\lambda\vec{\alpha}.[\![\pi]\!]^p\vec{q}\vec{\alpha}\ ([\![\pi']\!]^{p'}\vec{q}\vec{\alpha})
\end{array}$$

Normalisation by reducibility: escape lemma

We define $p\ \mathtt{wn} \triangleq \exists p'\ (p'\ \mathtt{nf} \wedge p \rightarrow_{\beta\eta} p')$ and take the normalisation model defined by $p\ \mathtt{r}\ X \triangleq p\ \mathtt{wn}$ and which satisfies stability by anti-reduction.

Let App_{wn} proves $a\vec{q}\ \mathtt{wn} \wedge q'\ \mathtt{wn} \Rightarrow a\vec{q}q'\ \mathtt{wn}$, $\mathsf{Var}_{wn}\ a$ proves $a\ \mathtt{wn}$ and Lam^{a}_{wn} proves $t\ \mathtt{wn} \Rightarrow \lambda a.t\ \mathtt{wn}$.

Escape lemma: $\forall A \begin{cases} \forall\vec{q}\ (a\vec{q}\ \mathtt{wn}\ \Rightarrow a\vec{q}\ \mathtt{r}\ A) \\ \wedge\forall p\ (p\ \mathtt{r}\ A \Rightarrow p\ \mathtt{wn}) \end{cases}$

Proof mutually by induction on the type:

$$\begin{array}{llcl}
\uparrow_A & a\vec{q}\ \mathtt{wn} & \Rightarrow & a\vec{q}\ \mathtt{r}\ A \\
\uparrow_X & \pi & \triangleq & \pi \\
\uparrow_{A \dot\rightarrow B} & \pi & \triangleq & \lambda q'.\lambda\alpha.\ \uparrow_B (\mathsf{App}_{wn}(\pi, \downarrow_A \alpha)) \\
 & & & \\
\downarrow_A & p\ \mathtt{r}\ A & \Rightarrow & p\ \mathtt{wn} \\
\downarrow_X & \alpha & \triangleq & \alpha \\
\downarrow_{A \dot\rightarrow B} & \alpha & \triangleq & \mathsf{Lam}^{a}_{wn}(\downarrow_B (\alpha\, a\, (\uparrow_A (\mathsf{Var}_{wn}\, a))))
\end{array}$$

Normalisation by typed reducibility

It becomes obvious that we can generalise realisability and Kripke semantics into a typed notion of reducibility.

Notes:

- To emphasise the similarity of computational content, we produced η-long normal form. We could also reason avoiding η.
- The proof scales to first-order universal quantification and to positive connectives when interpreted negatively.
- Interpreting positive connectives positively raises problems (see Ilik 2010).
- This kind of normalisation proof is also related to Type-Directed Partial Evaluation.

Tarskian semantics: soundness

Let $\mathcal{M}$ be a Tarskian model, i.e. an interpretation $\rho_{\mathcal{M}}$ of object-language atoms X as meta-language atoms. Truth and validity are defined by:

$$
\begin{array}{lcl}
\models_{\mathcal{M}} X & \triangleq & \rho_{\mathcal{M}}(X) \\
\models_{\mathcal{M}} A \dot{\to} B & \triangleq & \models_{\mathcal{M}} A \;\Rightarrow\; \models_{\mathcal{M}} B \\
\\
\Gamma \models_{\mathcal{M}} A & \triangleq & \models_{\mathcal{M}} \Gamma \;\Rightarrow\; \models_{\mathcal{M}} A \\
\Gamma \models A & \triangleq & \forall \mathcal{M}\, \Gamma \models_{\mathcal{M}} A
\end{array}
$$

Soundness (for minimal logic) works as in the Kripke and reducibility cases:

$$
\begin{array}{lcl}
[\![\ ]\!]_{\mathcal{M}} \;:\; \Gamma \vdash A & \Rightarrow & \Gamma \models_{\mathcal{M}} A \\
[\![\mathsf{Var}_n]\!]_{\mathcal{M}} & \triangleq & \lambda\vec{\alpha}.\alpha_n \\
[\![\mathsf{Lam}(\pi)]\!]_{\mathcal{M}} & \triangleq & \lambda\vec{\alpha}.\lambda\alpha.[\![\pi]\!]_{\mathcal{M}}(\vec{\alpha}, \alpha') \\
[\![\mathsf{App}(\pi, \pi')]\!]_{\mathcal{M}} & \triangleq & \lambda\vec{\alpha}.[\![\pi]\!]_{\mathcal{M}}\vec{\alpha}\ ([\![\pi']\!]_{\mathcal{M}}\vec{\alpha}) \\
\\
\mathit{sound} \;:\; \Gamma \vdash A & \Rightarrow & \Gamma \models A \\
\mathit{sound} \quad\ \pi & \triangleq & \lambda\mathcal{M}.[\![\pi]\!]_{\mathcal{M}}
\end{array}
$$

Tarskian semantics: completeness

There are several proofs of completeness (for classical logic):

- Henkin's proof
- Beth-Hintikka-Kanger-Schütte's proofs of *strong* completeness
- Rasiowa-Sikorski's variant of Henkin's proof
- ...

They are constructive as soon as:

- We interpret $\perp$ by an arbitrary formula (stronger than all other formulae)
- We interpret positive formulae negatively
- We strictly consider Tarskian semantics rather than two-valued semantics which would require a functional reification axiom: $\forall x\, \exists b\, (b = true \Leftrightarrow P(x)) \Rightarrow \exists f\, \forall x\, (f(x) = true \Leftrightarrow P(x))$, which itself would require an instance of unique choice and classical logic.

Henkin's proof: Assumptions on the object language

We assume to have a distinguished atom $\dot{\perp}$ and we reason in the theory $Class \triangleq \{\dot{\neg}\dot{\neg}A \dot{\rightarrow} A \mid A \text{ formula}\}$.

The following rules are then admissible:

$$\frac{Class, \Gamma, A \dot{\rightarrow} B \vdash \dot{\perp}}{Class, \Gamma \vdash A}\ \dot{\mathtt{Proj}}_1 \qquad \frac{Class, \Gamma, A \dot{\rightarrow} B \vdash \dot{\perp}}{Class, \Gamma \vdash \dot{\neg} B}\ \dot{\mathtt{Proj}}_2$$

$$\frac{Class, \Gamma \vdash \dot{\neg}\dot{\neg}A}{Class, \Gamma \vdash A}\ \dot{\mathtt{Dn}}$$

Tarskian semantics: A computational presentation of Henkin's proof

Let $\lceil A \rceil$ and ϕ form a Gödel's numbering of implicative formulae such that $\lceil \phi(n) \rceil = n$.

Let F_n be (informally) the countermodel built at step n. We write $F_\omega \vdash A$ for $\exists n \exists \Gamma \subset F_n\ (Class, \Gamma \vdash A)$ ("A gets provable at some step of the construction of a context equiconsistent to $\dot\neg A_0$") where $\Gamma \subset F_n$ is formally defined inductively:

$$\frac{}{\dot\neg A_0 \subset F_0}\ I_0 \qquad\qquad \frac{\Gamma \subset F_n}{\Gamma \subset F_{n+1}}\ I_S$$

$$\frac{\Gamma \subset F_n \qquad \forall \Gamma' \subset F_n\ (Class, \Gamma', \phi(n) \vdash \dot\bot \Rightarrow Class, \dot\neg A_0 \vdash \dot\bot)}{\Gamma, \phi(n) \subset F_{n+1}}\ I_n$$

The syntactic model $\mathcal{M}_0$ is defined by $\rho_{\mathcal{M}}(X) \triangleq F_\omega \vdash X$.

The core of the proof

$$
\begin{array}{llcl}
\uparrow_A & : F_\omega \vdash A & \longrightarrow & \vDash_{\mathcal{M}} A \\
\uparrow_{X)} & (n, \Gamma, f, \pi) & \triangleq & (n, \Gamma, f, \pi) \\
\uparrow_{A \dot{\to} B} & (n, \Gamma, f, \pi) & \triangleq & m \mapsto \begin{array}{l} \mathtt{dest}\ \downarrow_A m\ \mathsf{as}\ (n', \Gamma', f', \pi') \\ \mathtt{in}\ \uparrow_B (max(n, n'), \Gamma \cup \Gamma', \mathtt{join}^{\Gamma\Gamma'}_{nn'}(f, f'), \mathsf{App}(\pi, \pi')) \end{array}
\end{array}
$$

$$
\begin{array}{llcl}
\downarrow_A & : \vDash_{\mathcal{M}} A & \longrightarrow & F_\omega \vdash A \\
\downarrow_X & m & \triangleq & m \\
\downarrow_{A \dot{\to} B} & m & \triangleq & (n, (\dot{\neg} A_0, A \dot{\to} B),
\end{array}
$$

$$
I_n(\mathtt{inj}_n, (\Gamma, f, \pi) \mapsto \begin{array}{l} \mathtt{dest}\ \downarrow_B (m(\uparrow_A (n, \Gamma, f, \mathtt{Pr\dot{o}j}_1\, \pi)))\ \mathsf{as}\ (n', \Gamma', f', \pi') \\ \mathtt{in}\ \mathtt{flush}^{\Gamma \cup \Gamma'}_{max(n,n')}(\mathtt{join}^{\Gamma\Gamma'}_{nn'}(f, f'), \mathsf{App}(\mathtt{Pr\dot{o}j}_2\, \pi, \pi')) \end{array}),
$$

$$
\mathsf{Var}_1) \qquad \textit{where } n = \lceil A \dot{\to} B \rceil
$$

Auxiliary lemmas

$$
\begin{array}{llll}
\mathtt{flush}_n^{\Gamma} & : \Gamma \subset F_n \wedge \Gamma \vdash \bot & \longrightarrow & \dot\neg A_0 \vdash \bot \\
\mathtt{flush}_0^{\Gamma} & (\mathtt{I}_{\emptyset}, \pi) & \triangleq & \pi \\
\mathtt{flush}_{n+1}^{\Gamma} & (\mathtt{I}_{\mathtt{S}} f, \pi) & \triangleq & \mathtt{flush}_n^{\Gamma}(f, \pi) \\
\mathtt{flush}_{n+1}^{\Gamma A} & (\mathtt{I}_{\mathtt{n}}(f, H), \pi) & \triangleq & H\,\Gamma\, f\, \pi
\end{array}
$$

$$
\begin{array}{llllll}
\mathtt{join}_{n_1 n_2}^{\Gamma_1 \Gamma_2} & : \Gamma_1 \subset F_{n_1} & \wedge\ \Gamma_2 \subset F_{n_2} & \longrightarrow & \Gamma_1 \cup \Gamma_2 \subset F_{max(n_1, n_2)} & \\
\mathtt{join}_{00}^{\dot\neg A_0 \dot\neg A_0} & \mathtt{I}_{\emptyset} & \mathtt{I}_{\emptyset} & \triangleq & \mathtt{I}_{\emptyset} & \\
\mathtt{join}_{(n+1)(n+1)}^{(\Gamma_1 A)(\Gamma_2 A)} & \mathtt{I}_{\mathtt{n}}(f_1, H_1) & \mathtt{I}_{\mathtt{n}}(f_2, H_2) & \triangleq & \mathtt{I}_{\mathtt{n}}(\mathtt{join}_{nn}^{\Gamma_1' \Gamma_2'} f_1 f_2, H_1) & \\
\mathtt{join}_{(n+1)(n+1)}^{(\Gamma_1 A)\Gamma_2} & \mathtt{I}_{\mathtt{n}}(f_1, H_1) & \mathtt{I}_{\mathtt{S}} f_2 & \triangleq & \mathtt{I}_{\mathtt{n}}(\mathtt{join}_{nn}^{\Gamma_1' \Gamma_2} f_1 f_2, H_1) & \\
\mathtt{join}_{(n+1)(n+1)}^{\Gamma_1 (\Gamma_2 A)} & \mathtt{I}_{\mathtt{S}} f_1 & \mathtt{I}_{\mathtt{n}}(f_2, H_2) & \triangleq & \mathtt{I}_{\mathtt{n}}(\mathtt{join}_{nn}^{\Gamma_1 \Gamma_2'} f_1 f_2, H_2) & \\
\mathtt{join}_{(n+1)(n+1)}^{\Gamma_1 \Gamma_2} & \mathtt{I}_{\mathtt{S}} f_1 & \mathtt{I}_{\mathtt{S}} f_2 & \triangleq & \mathtt{I}_{\mathtt{S}}(\mathtt{join}_{nn}^{\Gamma_1 \Gamma_2} f_1 f_2) & \\
\mathtt{join}_{n_1 n_2}^{\Gamma_1 \Gamma_2} & \mathtt{I}_{\mathtt{S}} f_1 & f_2 & \triangleq & \mathtt{I}_{\mathtt{S}}(\mathtt{join}_{n_1' n_2}^{\Gamma_1 \Gamma_2} f_1 f_2) & \text{if } n_1 = n_1' + 1 > n_2 \\
\mathtt{join}_{n_1 n_2}^{(\Gamma_1 A_1)\Gamma_2} & \mathtt{I}_{\mathtt{n}_1'}(f_1, H_1) & f_2 & \triangleq & \mathtt{I}_{\mathtt{n}_1'}(\mathtt{join}_{n_1' n_2}^{\Gamma_1 \Gamma_2} f_1 f_2, H_1) & \text{if } n_1 = n_1' + 1 > n_2 \\
\mathtt{join}_{n_1 n_2}^{\Gamma_1 \Gamma_2} & f_1 & \mathtt{I}_{\mathtt{S}} f_2 & \triangleq & \mathtt{I}_{\mathtt{S}}(\mathtt{join}_{n_1 n_2'}^{\Gamma_1 \Gamma_2} f_1 f_2) & \text{if } n_1 < n_2' + 1 = n_2 \\
\mathtt{join}_{n_1 n_2}^{\Gamma_1 (\Gamma_2 A_2)} & f_1 & \mathtt{I}_{\mathtt{n}_2'}(f_2, H_2) & \triangleq & \mathtt{I}_{\mathtt{n}_2'}(\mathtt{join}_{n_1 n_2'}^{\Gamma_1 \Gamma_2} f_1 f_2, H_2) & \text{if } n_1 < n_2' + 1 = n_2
\end{array}
$$

$$
\begin{array}{lll}
\mathtt{inj}_n & : & \dot\neg A_0 \subset F_n \\
\mathtt{inj}_0 & \triangleq & \mathtt{I}_{\emptyset} \\
\mathtt{inj}_{n+1} & \triangleq & \mathtt{I}_{\mathtt{S}}(\mathtt{inj}_n)
\end{array}
$$

Final completeness result

$$\texttt{class}_0 \;:\; \forall A\ (Class \vDash_{\mathcal{M}_0} \dot\neg\dot\neg A \dot\rightarrow A)$$

$$\texttt{class}_0 \;\triangleq\; \lambda A.\lambda m.\uparrow_A (\texttt{dest}\ \downarrow_{\dot\neg\dot\neg A}\ m\ \texttt{as}\ (n, \Gamma, f, \pi)\ \texttt{in}\ (n, \Gamma, f, \dot{\mathrm{Dn}}\pi))$$

$$\texttt{compl}_{A_0} \;:\; \forall \mathcal{M}\ Class \vDash_{\mathcal{M}} A_0 \longrightarrow Class \vdash A_0$$

$$\texttt{compl}_{A_0}\quad \psi \quad \triangleq \dot{\mathsf{Dn}}(\mathsf{Lam}(\begin{array}{l}\texttt{dest}\ \downarrow_{A_0} (\psi\ \mathcal{M}_0\ \texttt{class}_0)\ \texttt{as}\ (n, \Gamma, f, \pi)\\ \texttt{in flush}_n^{\Gamma}(f, \mathsf{App}(\mathsf{Var}_{|\Gamma|}, \pi))\end{array}))$$

Henkin's proof: comments

In Kripke semantics, knowledge can be extended whenever a new assumption is known.

In Tarskian semantics instead, knowledge cannot be extended. However, we can consider that an assumption holds whenever we know how to get rid of it.

The model construction is computationally a type of continuation: any formula can be added to the model as soon as its addition comes with a continuation showing that it preserves consistency

No concrete "model" is built, even though the ordering matters on what the resulting proof is.

Henkin's proof: further comments

When composing Henkin's completeness with soundness:

- the resulting proof is not necessary normal
- even if the structure of the initial proof is used, it is "damaged" in the resulting proof
- can we twist Henkin's proof so as to return normal proofs?
- use effects to mimic the semantic and get a normal form?

A few references

NbE for λ-calculus: Berger-Schwichtenberg 1991

Computational analysis of reducibility proofs: Berger 1993

NbE by Kripke semantics: C. Coquand 1993, 2002; NbE by phase semantics: Okada 2002

NbE for disjunction: Altenkirch, Altenkirch-Dybjer-Hofmann-Scott 2001

Semantic normalisation for classical NbE: Herbelin-Gyesik-Lee 2010

Constructiveness of Tarskian semantics: Kreisel 1958, Kreisel 1962, Krivine 1996, Berardi 1999, Berardi-Valentini 2004, McCarty 2008,

Effects in completeness proofs: Ilik 2010.

...