On the use of side effects in logic: an investigation into proving completeness of classical logic

Hugo Herbelin

17 February 2012

Logic and Interaction 2012

Proofs and Programs

Outline

- On the relation between soundness, completeness and reducibility proofs
- An analysis of Henkin's proof of completeness for classical logic
- A logical system with monotonically updatable variables (work in progress)
- An enumeration-free proof of completeness for classical logic using effects

On the relation between soundness, completeness and reducibility proofs

Soundness $\circ$ (cut-free completeness) $=$ semantic normalisation

We consider the following:

- $\tilde{\vDash}$, a notion of derivability for an object language
- $\tilde{\vDash}_{cf}$, its restriction on cut-free proofs
- $\vdash_{ML}$, the derivability in some metalanguage (typically second-order arithmetic)

Several results show that, by composing soundness with completeness towards cut-free proofs, one obtains semantic normalisation (e.g. for Kripke semantics):

$$\vdash_{ML} \ [\, \Gamma \tilde{\vDash} A \quad \overset{\text{soundness}}{\longmapsto} \quad \forall \mathcal{M}\, \forall w\, (w \Vdash [\![\Gamma]\!]_{\mathcal{M}} \rightarrow w \Vdash [\![A]\!]_{\mathcal{M}}) \quad \overset{\text{completeness}}{\longmapsto} \quad \Gamma \tilde{\vDash}_{cf} A \,]$$

- C. Coquand [1993]: For Kripke semantics and implicative simply-typed λ-calculus
- Okada [1999]: For phase semantics and first-order linear logic
- ... (long list)

Soundness ∘ (cut-free completeness) = semantic normalisation

Note that by a simple step of cut-elimination in the metalanguage, it is actually enough to consider "the" syntactic model $\mathcal{M}_0$:

$$\vdash_{ML} \ [\, \Gamma \mathrel{\tilde{\models}} A \quad \overset{\text{soundness}}{\longmapsto} \quad \forall w \, (w \Vdash [\![\Gamma]\!]_{\mathcal{M}_0} \to w \Vdash [\![A]\!]_{\mathcal{M}_0}) \quad \overset{\text{completeness}}{\longmapsto} \quad \Gamma \mathrel{\tilde{\models}}_{cf} A \,]$$

The structure of a normalisation proof based on reducibility

Looking at the structure of normalisation proof by (typed) reducibility, one observes the following:

$$\vdash_{ML} [\, \Gamma \mathrel{\tilde{\models}} p : A \overset{\text{adequacy}}{\longmapsto} \forall \Delta \forall \vec{q} \in [\![\Delta \vdash \Gamma]\!]_{WN}\, p[\vec{q}] \in [\![\Delta \vdash A]\!]_{WN} \overset{\text{escape}}{\longmapsto} \exists p'\, p \rightarrow p' \wedge \Gamma \mathrel{\tilde{\models}}_{cf} p' : A$$

otherwise said, if we forget the relation between the original proof and its normal form:

$$\vdash_{ML} [\, \Gamma \mathrel{\tilde{\models}} A \overset{\text{adequacy}}{\longmapsto} (\forall \Delta [\![\Delta \vdash \Gamma]\!]_{WN} \rightarrow [\![\Delta \vdash A]\!]_{WN}) \overset{\text{escape}}{\longmapsto} \exists \Gamma\, \Gamma \mathrel{\tilde{\models}}_{cf} A\,]$$

where WN is the model of normalisability

$$p \in [\![\Delta \vdash X]\!]_{WN} \triangleq (\Delta \mathrel{\tilde{\models}} p : X \wedge p\, WN)$$
$$p \in [\![\Delta \vdash A \rightarrow B]\!]_{WN} \triangleq \forall q \in [\![\Delta \vdash A]\!]_{WN}\, (p\, q) \in [\![\Delta \vdash B]\!]_{WN}$$

Remark 1: the terminology "escape lemma" is taken from Schürmann and Sarnat

Remark 2: reducibility is also known as logical relations, or Tait's method

Adequacy vs soundness

It is obvious that both adequacy and soundness are maps from proofs of $\Gamma \vdash^{\sim} A$ to the reflection of this very same proof into a proof of the structurally similar formula $[\![\Gamma]\!] \to [\![A]\!]$ in the metalanguage (though some extra expressiveness is generally present in the metalanguage: realisers and contexts in the case of typed reducibility, worlds/contexts in the case of Kripke semantics).

Completeness vs escape lemma

Both completeness and the escape lemma have the same structure too:

Cut-free completeness actually proves the following by induction on A

$$\vdash_{ML} \quad [\ \Gamma \vDash_{ne} A \quad \stackrel{\text{reflection}}{\longmapsto} \quad \Gamma \Vdash [\![A]\!]_{\mathcal{M}_0} \quad \stackrel{\text{reification}}{\longmapsto} \quad \Gamma \vDash_{cf} A\ ]$$

where $\vDash_{ne}$ denotes the neutral proofs, i.e. the proofs of the form $E[a]$ for E a cut-free evaluation context

Similarly, the escape lemma proves

$$\vdash_{ML} \quad [\ \Gamma \vDash_{ne} p : A \quad \stackrel{\text{reflection}}{\longmapsto} \quad p \in [\![\Gamma \vdash A]\!]_{WN} \quad \stackrel{\text{reification}}{\longmapsto} \quad \Gamma \vDash_{cf} p : A\ ]$$

We are now going to investigate how far the standard completeness proof for classical logic with respect to two-valued models can follow this pattern...

An analysis of Henkin's proof of completeness for classical logic

Gödel's completeness

Let $\tilde{\vDash}$ be an object language for first-order logic and A be a first-order formula in $\tilde{\vDash}$

Gödel's completeness is the statement $\forall A\ (\models A) \rightarrow (\tilde{\vDash} A)$

We place ourself in an intuitionistic second-order arithmetical metalanguage $\vdash_{ML}$ (extended with computational Markov's principle). In this metalanguage, a classical proof of $\models A$ is exactly the same as a proof in $\vdash_{ML}$ of the second-order closure of A.

We shall give an executable proof-term $\texttt{compl}_A$ in $\vdash_{ML}$ that maps a meta-proof of A (i.e. a proof of the second-order closure of A in $\vdash_{ML}$) into an object-proof of A (i.e. a proof of A in $\vdash_{ML}$):

$$\vdash_{ML}\ \texttt{compl}_A : (\models A) \rightarrow (\tilde{\vDash} A)$$

Gödel's completeness formulated using reify/reflect functions

In completeness proofs with respect to Kripke semantics (see e.g. C. Coquand [1993]), a reify function $\downarrow_A$ maps each inference rule used in the metalanguage to the corresponding inference rule in the object language, and a reflect function $\uparrow_A$ works the way back so as to deal with the contravariance of implication.

We show that Henkin's proof can be formulated in terms of reify and reflect functions with a computational content more elaborated than in the case of Kripke semantics.

The completeness of classical first-order logic wrt two-valued models

First proof by Gödel [1929]

- reasoning on the prenex form + induction on the number of alternation of quantifiers + reasoning by contradiction

Standard proof by Henkin [1949]

- reasoning by contradiction + construction of a model by enumeration of the formulae over a language extended with Henkin constants coming from the skolemisation of the drinkers' paradox ($\exists x\,(P(x) \rightarrow \forall y\,P(y))$).

Tableaux-based proofs by Beth [1955], Hintikka [1955], Schütte [1956], Kanger [1957]

- building a tableau + reasoning by contradiction to show it has no infinite branch

Alternative proofs

- algebraic proofs by Mostowski [1948], Rasiowa-Sikorski [1950], ...

Different formulations

Strong (standard) completeness is one of the following two classically equivalent statements

- "Formula A true in all models of theory T implies A provable from (a finite subset) of T"
- "Theory T consistent implies T has a model"

Weak completeness is one of the following two classically equivalent statements

- "Formula A true in all models implies A provable"
- "Context Γ consistent implies Γ has a model"

Logical strength of completeness

- Kreisel [1962], after Gödel [1957]: completeness is equivalent to Markov's principle over intuitionistic second-order arithmetic
- Simpson [1999]: strong completeness equivalent to weak König's lemma over RCA_0

Avoiding the need for Markov's principle

Krivine's proof [1996]

- restricted to minimal classical logic (no $\bot \to A$) so that negation disappears in the definition of the model and Friedman's A-translation [1978] is applicable to get rid of Markov's principle

- analyzed by Berardi and Valentini [2001]: Krivine adds one extra (degenerated) model, the always-true model (similar to Friedman's fallible models and Veldman's exploding nodes in intuitionistic logic semantics)

- the modified statement is classically equivalent to the original one but does not need Markov's principle

- formalised in the PhoX proof assistant; formalised again by Danko Iliḱ in Coq

- Markov's principle = an exception mechanism; no need to avoid it after all, we can reason in the second-order arithmetic extension of IQC_{MP}

The statement of completeness
(weak form, restricted to the negative fragment)

$$t \in \mathcal{T} ::= x \mid f t_1 \dots t_{a_f}$$
$$A, B \in \mathcal{F} ::= P\, t_1 \dots t_{a_P} \mid \tilde{\bot} \mid A \tilde{\to} B \mid \tilde{\forall} x\, A$$

A model is a triple $(\mathcal{M}_D,\, \mathcal{M}(f) \in \mathcal{M}_D^{a_f} \to \mathcal{M}_D,\, \mathcal{M}(P) \in \mathcal{P}(\mathcal{M}_D^{a_P}))$. Truth in $\mathcal{M}$ is defined recursively:

$$\begin{array}{lcl}
[\![x]\!]^\sigma_\mathcal{M} & \triangleq & \sigma(x) \\
[\![f t_1 \dots t_{a_f}]\!]^\sigma_\mathcal{M} & \triangleq & \mathcal{M}(f)[\![t_1]\!]^\sigma_\mathcal{M} \dots [\![t_{a_f}]\!]^\sigma_\mathcal{M} \\
[\![P t_1 \dots t_{a_f}]\!]^\sigma_\mathcal{M} & \triangleq & \mathcal{M}(P)[\![t_1]\!]^\sigma_\mathcal{M} \dots [\![t_{a_P}]\!]^\sigma_\mathcal{M} \\
[\![\tilde{\bot}]\!]^\sigma_\mathcal{M} & \triangleq & \bot \\
[\![A \tilde{\to} B]\!]^\sigma_\mathcal{M} & \triangleq & [\![A]\!]^\sigma_\mathcal{M} \to [\![B]\!]^\sigma_\mathcal{M} \\
[\![\tilde{\forall} x\, A]\!]^\sigma_\mathcal{M} & \triangleq & \forall t \in \mathcal{M}_D\, [\![A]\!]^{\sigma[x \leftarrow t]}_\mathcal{M}
\end{array}$$

A model is classical, written $Class(\mathcal{M})$ if for each A and σ, $[\![\tilde{\neg}\tilde{\neg} A]\!]^\sigma_\mathcal{M} \to [\![A]\!]^\sigma_\mathcal{M}$.

The completeness statement : $\forall A\, (\forall \mathcal{M} \forall \sigma\, Class(\mathcal{M}) \to [\![A]\!]^\sigma_\mathcal{M}) \to \; [\tilde{\vdash} A]$

Remarks on the formulation

We placed ourselves in intuitionistic second-order arithmetic with MP.

An alternative is to reason instead in the arithmetic of types of rank 2 and define $\mathcal{M}(P)$ as a boolean function in $\mathcal{M}_D^{a_P} \rightarrow \texttt{bool}$. The completeness proof then needs the axiom of unique choice (AC!) which makes the metatheory actually equivalent to second-order arithmetic (assuming classical reasoning). To avoid having to computationally interpret AC!, we prefer to directly reason in second-order arithmetic.

It is also common to replace $\mathcal{M}_P$ by a set $\mathcal{M}_F$ of formulae enriched over $\mathcal{D}$ such that:

$$
\begin{array}{lcl}
\tilde{\bot} \in \mathcal{M}_F & \leftrightarrow & \bot \\
A \tilde{\rightarrow} B \in \mathcal{M}_F & \leftrightarrow & A \in \mathcal{M}_F \rightarrow B \in \mathcal{M}_F \\
\tilde{\forall} x\, A \in \mathcal{M}_F & \leftrightarrow & \forall t\, A[t/x] \in \mathcal{M}_F \\
A \in \mathcal{M}_F & \leftrightarrow & \neg\neg A \in \mathcal{M}_F
\end{array}
$$

Our approach has both the advantage of avoiding to consider formulae enriched over $\mathcal{D}$ and to make the connection with intuitionistic models (e.g. Kripke) closer.

The finitist content of the Henkin's axioms

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

Let x_n be a variable fresh in $\phi(0), \ldots, \phi(n)$.

The context of Henkin axioms known at step n is defined by taking

$$
\begin{array}{lcll}
\Theta_0 & \triangleq & \epsilon & \\
\Theta_{n+1} & \triangleq & \Theta_n, A[x_n/x] \tilde{\rightarrow} \tilde{\forall} x\, A & \text{if } \phi(n) = \tilde{\forall} x\, A \\
\Theta_{n+1} & \triangleq & \Theta_n & \text{otherwise}
\end{array}
$$

The finitist content of the model construction

Let A_0 be the formula we expect a proof of.

Let us write F_n to denote the model built at step n. We do not define F_n but instead define the property $\Gamma \subset F_n$ meaning that Γ is a subset of the model virtually built at level n:

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

$$\frac{\Gamma \subset F_n \qquad \phi(n) = A \tilde{\rightarrow} B \qquad \forall \Gamma' \subset F_n \;\; [\Theta_n, \Gamma', A \tilde{\rightarrow} B \tilde{\vdash} \tilde{\bot}] \rightarrow \bot}{\Gamma, A \tilde{\rightarrow} B \subset F_{n+1}} I_n$$

In particular, only formulae of the form $A \tilde{\rightarrow} B$ need to be added to F_n.

Note

For a formula $A\tilde{\rightarrow}B$ to be added to F_n henceforth means that we have a continuation which reduces any proof of $\Gamma', A\tilde{\rightarrow}B \tilde{\vdash} \tilde{\bot}$ to a proof of $\tilde{\neg}A_0 \vdash \tilde{\bot}$ (since Markov's principle amounts to identify $\bot$ with $\tilde{\neg}A_0 \vdash \tilde{\bot}$) for any extension Γ' bound by the list of formulas known at step n of the enumeration.

The syntactic model

Being provable at some stage of the construction of the model is defined by

$$A \in F_\omega \;\triangleq\; \exists n \exists \Gamma \subset F_n \, [\Theta_n, \Gamma \vdash A]$$

i.e. "A becomes provable after provability has been extended with enough *context*, together with a continuation that allows to get rid of this extra knowledge"

The (syntactic) model $\mathcal{M}_0$ is then defined from provability of atoms at some stage n:

$$\begin{array}{lcl} \mathcal{D} & \triangleq & \mathcal{T} \\ \mathcal{M}(f)(t_1, \ldots, t_{a_f}) & \triangleq & f(t_1, \ldots, t_{a_f}) \\ \mathcal{M}(P)(t_1, \ldots, t_{a_P}) & \triangleq & P(t_1, \ldots, t_{a_P}) \in F_\omega \end{array}$$

The object language

We assume given a (non-minimal) set of appropriate object language constructions:

$\tilde{\mathtt{Ax}}_i : [\Gamma, A, \Gamma' \tilde{\vDash} A]$ (for Γ' of length i)

$\tilde{\mathtt{Ax}}'_i : [\Gamma, A, \Gamma' \tilde{\vDash} A]$ (for Γ of length i)

$\tilde{\mathtt{Dn}} : [\Gamma \tilde{\vDash} \tilde{\neg}\tilde{\neg} A] \longrightarrow [\Gamma \tilde{\vDash} A]$

$\tilde{\mathtt{Abs}} : [\Gamma, A \tilde{\vDash} B] \longrightarrow [\Gamma \tilde{\vDash} A \tilde{\to} B]$

$\tilde{\mathtt{App}}^{\to} : [\Gamma \tilde{\vDash} A \tilde{\to} B] \longrightarrow [\Gamma' \tilde{\vDash} A] \longrightarrow [\Gamma \cup \Gamma' \tilde{\vDash} B]$

$\tilde{\mathtt{Drinker}}_n : [A[x_n/x] \tilde{\to} \tilde{\forall} x\, A, \Gamma \tilde{\vDash} \tilde{\bot}] \longrightarrow [\Gamma \tilde{\vDash} \tilde{\bot}]$ if $\phi(n+1) = \tilde{\forall} x A$ and x_n fresh

$\tilde{\mathtt{Drinker}}_n : [\Gamma \tilde{\vDash} \tilde{\bot}] \longrightarrow [\Gamma \tilde{\vDash} \tilde{\bot}]$ otherwise

$\tilde{\mathtt{App}}^{\forall} : [\Gamma \tilde{\vDash} \tilde{\forall} x\, A(x)] \longrightarrow \tilde{\forall} t \in \mathcal{T}\, [\Gamma \tilde{\vDash} A(t)]$

$\pi_1^{\tilde{\to}} : [\Gamma, A \tilde{\to} B \tilde{\vDash} \tilde{\bot}] \longrightarrow [\Gamma \tilde{\vDash} A]$

$\pi_2^{\tilde{\to}} : [\Gamma, A \tilde{\to} B \tilde{\vDash} \tilde{\bot}] \longrightarrow [\Gamma \tilde{\vDash} \tilde{\neg} B]$

$\tilde{\mathtt{Exfalso}} : [\Gamma \tilde{\vDash} \tilde{\bot}] \longrightarrow [\Gamma \tilde{\vDash} A]$

The core of the proof

$$\begin{array}{llll}
\downarrow_A & : A \in \mathcal{M} & \rightarrow & A \in F_\omega \\
\downarrow_{P(\vec{t})} & m & \triangleq & m \\
\downarrow_{\tilde{\bot}} & m & \triangleq & \mathtt{Ex\tilde{f}also}\, m \\
\downarrow_{A \tilde{\to} B} & m & \triangleq & (n, (\tilde{\neg} A_0, A \tilde{\to} B), \\
& & & \quad I_n(\mathtt{inj}_n, (\Gamma, f, p) \mapsto \begin{array}{l} \mathtt{dest}\ \downarrow_B (m(\uparrow_A (n, \Gamma, f, \pi_1^{\tilde{\to}}\, p)))\ \mathtt{as}\ (n', \Gamma', f', p') \\ \mathtt{in}\ \mathtt{flush}^{\Gamma \cup \Gamma'}_{max(n,n')}(\mathtt{join}^{\Gamma\Gamma'}_{nn'}(f, f'), \mathtt{A\tilde{p}p}^{\to}(\pi_2^{\tilde{\to}}\, p, p')) \end{array}), \\
& & & \quad \mathtt{\tilde{A}x}_1) \qquad \textit{where } n = \lceil A \tilde{\to} B \rceil \\
\downarrow_{\tilde{\forall} x\, A} & m & \triangleq & \mathtt{dest}\ \downarrow_{A[x_n/x]} (m\, x_n)\ \mathtt{as}\ (n', \Gamma', f', p') \\
& & & \mathtt{in}\ (max(n, n'), \Gamma', \mathtt{join}^{(\neg A_0)\Gamma'}_{nn'}(\mathtt{inj}_n, f'), \mathtt{A\tilde{p}p}^{\to}(\mathtt{\tilde{A}x}'_0, p')) \\
& & & \qquad \textit{where } n = \lceil \tilde{\forall} x\, A \rceil
\end{array}$$

$$\begin{array}{llll}
\uparrow_A & : A \in F_\omega & \rightarrow & A \in \mathcal{M} \\
\uparrow_{P(\vec{t})} & (n, \Gamma, f, p) & \triangleq & (n, \Gamma, f, p) \\
\uparrow_{\tilde{\bot}} & (n, \Gamma, f, p) & \triangleq & \mathtt{flush}^{\Gamma}_{n}(f, p) \\
\uparrow_{A \tilde{\to} B} & (n, \Gamma, f, p) & \triangleq & m \mapsto \begin{array}{l} \mathtt{dest}\ \downarrow_A m\ \mathtt{as}\ (n', \Gamma', f', p') \\ \mathtt{in}\ \uparrow_B (max(n, n'), \Gamma \cup \Gamma', \mathtt{join}^{\Gamma\Gamma'}_{nn'}(f, f'), \mathtt{A\tilde{p}p}^{\to}(p, p')) \end{array} \\
\uparrow_{\tilde{\forall} x\, A} & (n, \Gamma, f, p) & \triangleq & t \mapsto \uparrow_{A[t/x]} (n, \Gamma, f, \mathtt{A\tilde{p}p}^{\forall}(p, t))
\end{array}$$

Auxiliary lemmas

$$\begin{array}{llcl}
\mathtt{flush}^{\Gamma}_{n} & : \Gamma \subset F_n \wedge [\Theta_n, \Gamma \tilde{\vDash} \tilde{\bot}] & \longrightarrow & \bot \\
\mathtt{flush}^{\Gamma}_{0} & (\mathtt{I}_0, p) & \triangleq & \mathtt{raise}_{\hat{\alpha}_0}\, p \\
\mathtt{flush}^{\Gamma}_{n+1} & (\mathtt{I}_{\mathtt{S}} f, p) & \triangleq & \mathtt{flush}^{\Gamma}_{n}(f, \mathtt{Dri\tilde{n}ker}_n\, p) \\
\mathtt{flush}^{\Gamma A}_{n+1} & (\mathtt{I}_{\mathtt{n}}(f, k), p) & \triangleq & k\, \Gamma\, f\, p
\end{array}$$

$$\begin{array}{lllcll}
\mathtt{join}^{\Gamma_1\Gamma_2}_{n_1 n_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}^{\tilde{\neg} A_0 \tilde{\neg} A_0}_{00} & \mathtt{I}_0 & \mathtt{I}_0 & \triangleq & \mathtt{I}_0 & \\
\mathtt{join}^{(\Gamma_1 A)(\Gamma_2 A)}_{(n+1)(n+1)} & \mathtt{I}_{\mathtt{n}}(f_1, k_1) & \mathtt{I}_{\mathtt{n}}(f_2, k_2) & \triangleq & \mathtt{I}_{\mathtt{n}}(\mathtt{join}^{\Gamma_1'\Gamma_2'}_{nn} f_1 f_2, k_1) & \\
\mathtt{join}^{(\Gamma_1 A)\Gamma_2}_{(n+1)(n+1)} & \mathtt{I}_{\mathtt{n}}(f_1, k_1) & \mathtt{I}_{\mathtt{S}} f_2 & \triangleq & \mathtt{I}_{\mathtt{n}}(\mathtt{join}^{\Gamma_1'\Gamma_2}_{nn} f_1 f_2, k_1) & \\
\mathtt{join}^{\Gamma_1(\Gamma_2 A)}_{(n+1)(n+1)} & \mathtt{I}_{\mathtt{S}} f_1 & \mathtt{I}_{\mathtt{n}}(f_2, k_2) & \triangleq & \mathtt{I}_{\mathtt{n}}(\mathtt{join}^{\Gamma_1\Gamma_2'}_{nn} f_1 f_2, k_2) & \\
\mathtt{join}^{\Gamma_1\Gamma_2}_{(n+1)(n+1)} & \mathtt{I}_{\mathtt{S}} f_1 & \mathtt{I}_{\mathtt{S}} f_2 & \triangleq & \mathtt{I}_{\mathtt{S}}(\mathtt{join}^{\Gamma_1\Gamma_2}_{nn} f_1 f_2) & \\
\mathtt{join}^{\Gamma_1\Gamma_2}_{n_1 n_2} & \mathtt{I}_{\mathtt{S}} f_1 & f_2 & \triangleq & \mathtt{I}_{\mathtt{S}}(\mathtt{join}^{\Gamma_1\Gamma_2}_{n_1' n_2} f_1 f_2) & \text{if } n_1 = n_1' + 1 > n_2 \\
\mathtt{join}^{(\Gamma_1 A_1)\Gamma_2}_{n_1 n_2} & \mathtt{I}_{\mathtt{n}_1'}(f_1, k_1) & f_2 & \triangleq & \mathtt{I}_{\mathtt{n}_1'}(\mathtt{join}^{\Gamma_1\Gamma_2}_{n_1' n_2} f_1 f_2, k_1) & \text{if } n_1 = n_1' + 1 > n_2 \\
\mathtt{join}^{\Gamma_1\Gamma_2}_{n_1 n_2} & f_1 & \mathtt{I}_{\mathtt{S}} f_2 & \triangleq & \mathtt{I}_{\mathtt{S}}(\mathtt{join}^{\Gamma_1\Gamma_2}_{n_1 n_2'} f_1 f_2) & \text{if } n_1 < n_2' + 1 = n_2 \\
\mathtt{join}^{\Gamma_1(\Gamma_2 A_2)}_{n_1 n_2} & f_1 & \mathtt{I}_{\mathtt{n}_2'}(f_2, k_2) & \triangleq & \mathtt{I}_{\mathtt{n}_2'}(\mathtt{join}^{\Gamma_1\Gamma_2}_{n_1 n_2'} f_1 f_2, k_2) & \text{if } n_1 < n_2' + 1 = n_2
\end{array}$$

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

Final weak completeness result

$$\texttt{class}_0 \;:\; (\tilde{\neg}\tilde{\neg}A) \in \mathcal{M}_0 \;\longrightarrow\; A \in \mathcal{M}_0$$
$$\texttt{class}_0 \quad m \;\triangleq\; \uparrow_A (\texttt{dest} \downarrow_{\tilde{\neg}\tilde{\neg}A} m\,\texttt{as}\,(n,\Gamma,f,p)\,\texttt{in}\,(n,\Gamma,f,\tilde{\texttt{dn}}p))$$

$$\texttt{compl}_{\texttt{A}_0} \;:\; (\forall\mathcal{M}\forall\sigma\, Class(\mathcal{M}) \to [\![A_0]\!]^\sigma_\mathcal{M}) \;\longrightarrow\; \tilde{\vdash} A_0$$
$$\texttt{compl}_{\texttt{A}_0} \quad \psi \;\triangleq\; \tilde{\texttt{Dn}}(\tilde{\texttt{Abs}}(\texttt{try}_{\texttt{new}\,\hat{\alpha}_0} \begin{array}{l}\texttt{dest} \downarrow_{A_0} (\psi\,\mathcal{M}_0\,\texttt{id}\,\texttt{class}_0)\,\texttt{as}\,(n,\Gamma,f,p)\\ \texttt{in}\,\texttt{Ex}\tilde{\texttt{fa}}\texttt{lso}\,\texttt{flush}^\Gamma_n(f,\tilde{\texttt{App}}^{\to}(\tilde{\texttt{Ax}}_{|\Gamma|-1},p))\end{array}))$$

Remarks about the computational content

If A_0 is provable, then the model (virtually) built is actually the degenerated model that contains all formulae, including $\tilde{\bot}$.

Along the computational interpretation of Markov's principle, reasoning classically by assuming $[\tilde{\neg} A_0 \tilde{\vdash} \tilde{\bot}] \to \bot$ is the same as providing an exception which returns a derivation of $[\tilde{\neg} A_0 \tilde{\vdash} \tilde{\bot}]$ as soon as a contradiction is obtained. Along Friedman's A-translation, this amounts to reinterpret $\bot$ as the formula $[\tilde{\neg} A_0 \tilde{\vdash} \tilde{\bot}]$.

Computationally, the proof of a negation can be seen as a continuation. Combined with the computational interpretation of Markov's principle, this is the same as a continuation that eventually returns a derivation of $[\tilde{\neg} A_0 \tilde{\vdash} \tilde{\bot}]$.

In particular, a proof that some finite section $\tilde{\neg} A_0, \Gamma$ of the model is consistent is the same as a continuation that transforms a derivation of $[\tilde{\neg} A_0, \Gamma \tilde{\vdash} \tilde{\bot}]$ into a derivation of $[\tilde{\neg} A_0 \tilde{\vdash} \tilde{\bot}]$.

More remarks about the computational content

The ordering of formulae has an effect on the order of application of continuations: continuations are applied in the decreasing order of the Gödel number of the formulae.

In case of branching, i.e. in the case of modus ponens, if two continuations are available at level n, the `join` function arbitrary chooses one of them. In particular, some subproofs of the initial meta-proof might be lost and replaced by another proof of the same formula in the same original meta-proof.

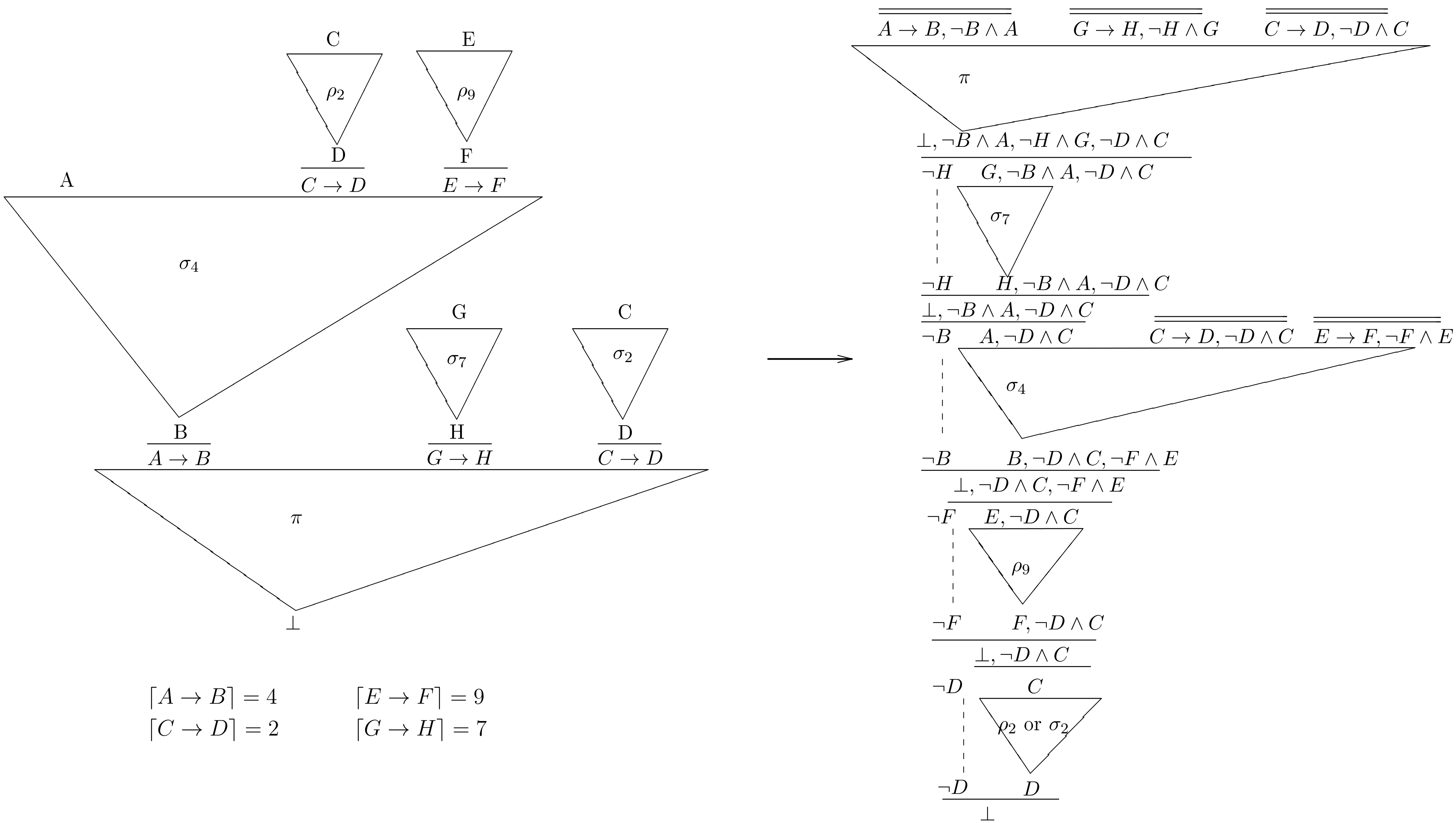
Compared to the completeness proof with respect to Kripke semantics where the world is locally extended with the knowledge of A to show that $A \tilde{\to} B$ is provable, here, in the completeness for two-valued semantics, one extends the knowledge Γ with $A \tilde{\to} B$ but altogether with a proof that contradicting $\Gamma, A \tilde{\to} B$ (in the sense of a derivation of $\Gamma, A \tilde{\to} B \tilde{\vdash} \tilde{\bot}$) eventually reduces to a derivation of $\tilde{\neg} A_0 \tilde{\vdash} \tilde{\bot}$.

It is worth noticing that the definition of $\Gamma \subset F_n$ is of high logical complexity. Due to the contravariance in the clause I_n, the definition of $\Gamma \subset F_{n+1}$ involves implications nested at level n. Henceforth, the definition of $A \in F_\omega$ is a formula whose implication nesting depth is arbitrary large.

Remarks about the Henkin axioms

Henkin axiom $A[x_n/x]\tilde{\rightarrow}\tilde{\forall}xA$ can be seen as a way to delegate the ensurance of the (moral) freshness of x_n and the ability to go from $A[x_n/x]$ to $\tilde{\forall}xA$ even in contexts where other occurrences of x_n might occur (namely the context $\Gamma' \subset F_{n'}$ in the $\tilde{\forall}$ clause of $\downarrow$). Henkin axioms are eliminated in `flush` following the order of dependencies between fresh variables. At stage n, x_n in $A[x_n/x]\tilde{\rightarrow}\tilde{\forall}xA$ is ensured to be fresh in any $\Gamma \subset F_n$. It then reduces to the drinkers' paradox $\tilde{\exists}y(A[y/x]\tilde{\rightarrow}\tilde{\forall}xA)$ which is provable.

Intuition about the computational content



β-normal η-expanded meta-proof

resulting object language classical prof (not cut-free!)

The strong statement of completeness

Let T be a theory. The standard statement of completeness can be obtained by modifying condition I_n of F_n as follows:

$$\frac{\Gamma \subset F_n \qquad \forall \Gamma' \subset F_n \, \forall \Delta \subset T \;\; [\Delta, \Theta_n, \Gamma', \{A\}_n \mathrel{\tilde{\vdash}} \tilde{\bot}] \to \bot}{\Gamma, \{A\}_n \subset F_{n+1}} \, I_n$$

and defining $A \in F_\omega$ as $\exists n \exists \Gamma \subset F_n \exists \Delta \subset T \, [\Delta, \Theta_n, \Gamma \mathrel{\tilde{\vdash}} A]$.

Then, every time the union of Γ and Γ' is taken, the union of Δ and Δ' is taken also.

In $\mathtt{try}_{\mathtt{new}\,\hat{\alpha}_0}$, the variable $\hat{\alpha}_0$ is taken to be of type $\exists \Delta \subset T \;\; [\Delta, \tilde{\neg} A_0 \mathrel{\tilde{\vdash}} \tilde{\bot}]$ and in the first case of $\mathtt{flush} : \Gamma \subset F_n \wedge [\Delta, \Theta_n, \Gamma \mathrel{\tilde{\vdash}} \tilde{\bot}] \to \bot$, the extra parameter $\Delta \subset T$ is thrown together with the proof of $[\Delta, \tilde{\neg} A_0 \mathrel{\tilde{\vdash}} \tilde{\bot}]$.

Open questions

The proof is mainly a linear proof made of cuts between clauses. Is there a connection to see with resolution proofs?

Can we modify the proof such that it produces normal forms?

Can we get rid of the total ordering of formulae?

A logical system with monotonically updatable variables

(work in progress)

Monadic-style vs direct-style

It turns out that some form of memory assignment is used in logic: both of

- Kripke semantics based translations
- Cohen's forcing translation

correspond to providing *monotonically* memory assignment in *monadic style*.

Kripke semantics based translations (i.e. $\mathit{Kripke}_x\, A(x) \triangleq \forall x' \geq x\, A(x')$ for x ranging over $\mathcal{W}$) is a dependent form of the environment monad ($\mathit{Env}\,(A) \triangleq \mathcal{W} \to A$) known to provide Lisp-style dynamic bindings.

Cohen's forcing translation ($\mathit{Forcing}_x\, A(x) \triangleq \forall x' \geq x \exists x'' \geq x'\, A(x'')$ for x ranging over some domain S) is a dependent form of the state monad ($\mathit{State}\, A \triangleq S \to S \times A$).

Towards a logic with side effects

We shall now sketch an extension of second-order intuitionistic arithmetic with the following two effects:

- exceptions (i.e. a weak form of classical logic)

- monotonically updatable memory

Note: Soundness is expected to come by embedding into second-order intuitionistic arithmetic using a combination of Kripke-style (dependent) environment monad, Friedman-style exception monad, and Coquand-Hofmann's translation [1999].

Logical rules providing delimited *direct-style* exceptions and monotone memory updates

A rule to simultaneously declare a memory x with initial value t and monotonically updatable along a preorder $\geq$ and an handler of exceptions of name $\hat{\alpha}$ in type T for T a Σ-formula (this delimits the *direct-style* use of the memory update and exception effects):

$$\frac{\Gamma, \hat{\alpha} : T(x)^{\perp\!\!\perp}, b : x \geq t \vdash q : T(x) \qquad \Gamma \vdash r : \texttt{preorder}\ \geq}{\Gamma \vdash \texttt{set}\, x := t\, \texttt{as}\, b\, \texttt{using}\, r\, \texttt{in}\, \texttt{try}_{\hat{\alpha}}\, q : T(t)}\,\text{SETEFF}$$

A rule to monotonically update the value of the memory x (this provides in *direct-style* what Kripke-style translation $Kripke$ provides):

$$\frac{\Gamma, b : x \geq u(x') \vdash q : T(x) \qquad \Gamma \vdash r : u(x) \geq x \qquad (\hat{\alpha} : T(x)^{\perp\!\!\perp}) \in \Gamma \qquad x' \text{ fresh}}{\Gamma \vdash \texttt{update}\, x := u(x)\, \texttt{as}\, (x', b)\, \texttt{by}\, r\, \texttt{in}\, \texttt{try}_{\hat{\alpha}}\, q : T(u(x))}\,\text{UPDATE}$$

A rule to raise an exception of name $\hat{\alpha}$ in type T (this provides in *direct-style* what the exception monad Exc provides in monadic style):

$$\frac{\Gamma \vdash p : T(x) \qquad (\hat{\alpha} : T(x)^{\perp\!\!\perp}) \in \Gamma}{\Gamma \vdash \texttt{raise}_{\hat{\alpha}}\, p : B}\,\text{RAISE}$$

Condition for instanciation of second-order predicates

$$\frac{\Gamma \vdash \forall X\, A(X) \qquad \Gamma \vdash \texttt{monotone}_\Gamma\, B}{\Gamma \vdash A(X)[B(\vec{y})/X(\vec{y})]}\forall^2_E$$

where $\texttt{monotone}_\Gamma\, B$ means that B should be monotone with respect to all updatable variables declared in Γ

An enumeration-free proof of Gödel's completeness

An enumeration-free proof of Gödel's completeness

Weak completeness: if A is true in all models $\mathcal{M}$, then A is provable.

We took $\lambda\mu$-calculus equipped with $\tilde{\to}$, $\tilde{\forall}$ and $\tilde{\bot}$ as object language. Inference rules are:

$$
\begin{array}{lll}
\tilde{\texttt{Dn}} & : & (\Gamma, A^{\tilde{\perp\!\!\!\perp}} \tilde{\vdash} \tilde{\perp\!\!\!\perp}) \to (\Gamma \tilde{\vdash} A) \\
\tilde{\texttt{Throw}} & : & (\Gamma', A^{\tilde{\perp\!\!\!\perp}} \subset \Gamma) \to (\Gamma \tilde{\vdash} A) \to (\Gamma \tilde{\vdash} B) \\
\tilde{\texttt{App}}_{\to} & : & (\Gamma \tilde{\vdash} A \tilde{\to} B) \to (\Gamma \tilde{\vdash} A) \to (\Gamma \tilde{\vdash} B) \\
\tilde{\texttt{Abs}}_{\to} & : & (\Gamma, A \tilde{\vdash} B) \to (\Gamma \tilde{\vdash} A \tilde{\to} B) \\
\tilde{\texttt{App}}_{\forall} & : & (\Gamma \tilde{\vdash} \tilde{\forall} x\, A) \to (\Gamma \tilde{\vdash} A[t/x]) \\
\tilde{\texttt{Abs}}_{\forall} & : & (\tilde{y} \text{ fresh in } \Gamma) \to (\Gamma \tilde{\vdash} A[\tilde{y}/x]) \to (\Gamma \tilde{\vdash} \tilde{\forall} x\, A) \\
\tilde{\texttt{Ax}} & : & (\Gamma', A \subset \Gamma) \to (\Gamma \tilde{\vdash} A)
\end{array}
$$

An enumeration-free proof of Gödel's completeness

Let A_0 be a formula and let us prove $A_0^{\tilde{\perp\!\!\perp}} \tilde{\vDash} \tilde{\perp\!\!\perp}$. The idea is to consider an updatable variable Γ initialised to $A_0^{\perp\!\!\perp}$ with $\Gamma \tilde{\vDash} \tilde{\perp\!\!\perp}$ as objective (i.e. as formula "$T(\Gamma)$") and to take the syntactic model $\mathcal{M}_0$ defined by $A \in \mathcal{M}_0$ iff $\Gamma \tilde{\vDash} A$.

Hence, if $H : \forall \mathcal{M} \forall \sigma\, Class(\mathcal{M}) \to [\![A_0]\!]^{\sigma}_{\mathcal{M}}$, then the proof of $\vdash A_0$ is

$$\mathtt{compl}_{\mathtt{A_0}}\, H \triangleq \tilde{\mathtt{Dn}}(\mathtt{set}\, \Gamma := A_0^{\perp\!\!\perp}\, \mathtt{as}\, b\, \mathtt{in}\, \mathtt{try}_{\hat{\alpha}}\, \mathtt{Thr\tilde{o}w}(b, \downarrow_{A_0} (H\, \mathcal{M}_0\, \mathtt{id}\, s_0)))$$

where

$\downarrow_A$	proves	$A \in \mathcal{M}_0 \;\to\; \Gamma \vdash A$
s_0	proves	$\tilde{\neg}\tilde{\neg} A_0 \in \mathcal{M}_0 \to A_0 \in \mathcal{M}_0$
$\mathtt{id}$	is the identity substitution	
$\hat{\alpha}$	:	$A_0^{\tilde{\perp\!\!\perp}} \tilde{\vDash} \tilde{\perp\!\!\perp}$
b	:	$A_0^{\tilde{\perp\!\!\perp}} \subset \Gamma$

An enumeration-free proof of completeness for classical logic using effects

We proceed as in a completeness proof w.r.t. Kripke models except that the world, a context, is now an updatable variable

$$
\begin{array}{llll}
\uparrow_A & : \Gamma \vdash A & \longrightarrow & A \in \mathcal{M}_0 \\
\uparrow_{P(\vec{t})} & g & \triangleq & g \\
\uparrow_{\tilde{\bot}} & g & \triangleq & \texttt{raise}_{\hat{\alpha}}\, g \\
\uparrow_{A \tilde{\to} B} & g & \triangleq & m \mapsto \uparrow_B \tilde{\texttt{App}}_{\to}(g, \downarrow_A m) \\
\uparrow_{\tilde{\forall} x\, A} & g & \triangleq & t \mapsto \tilde{\texttt{App}}_{\forall}(\uparrow_{A[t/x]}\, g)
\end{array}
$$

$$
\begin{array}{llll}
\downarrow_A & : A \in \mathcal{M}_0 & \longrightarrow & \Gamma \vdash A \\
\downarrow_{P(\vec{t})} & m & \triangleq & m \\
\downarrow_{\tilde{\bot}} & m & \triangleq & \texttt{exfalso}\, m \\
\downarrow_{A \tilde{\to} B} & m & \triangleq & \tilde{\texttt{Abs}}_{\to}(\tilde{\texttt{Dn}}(\texttt{update}\, \Gamma := (\Gamma, A)\, \texttt{as}\, b_1\, \texttt{by}\, r_1\, \texttt{in}\, \texttt{try}_{\hat{\alpha}} \\
& & & \quad \texttt{update}\, \Gamma := (\Gamma, B^{\perp\!\!\perp})\, \texttt{as}\, b_2\, \texttt{by}\, r_2\, \texttt{in}\, \texttt{try}_{\hat{\alpha}} \\
& & & \quad \tilde{\texttt{Throw}}(b_2, \downarrow_B\, (m\, (\uparrow_A \tilde{\texttt{Ax}}(b_1)))))) \\
\downarrow_{\tilde{\forall} x\, A} & m & \triangleq & \texttt{let}\ \tilde{y} = \texttt{fresh}\, \Gamma\ \texttt{in}\ \tilde{\texttt{Abs}}_{\forall}(s, \downarrow_{A[\tilde{y}/x]}\, (m\, \tilde{y}))
\end{array}
$$

where r_1 and r_2 prove $\Gamma \subset (\Gamma, A)$ and $\Gamma \subset (\Gamma, B^{\tilde{\perp\!\!\perp}})$ while s proves freshness of $\texttt{fresh}\,\Gamma$ in Γ for $\texttt{fresh}\ \Gamma$ an appropriate generator of a name fresh in Γ.

An enumeration-free proof of Gödel's completeness

This proof is constructive... we can compute with it using exceptions and assignment!

The proof is extensible to strong completeness: if A is true in all models $\mathcal{M}$ satisfying theory $\mathcal{T}$, then A is provable in some finite subset of $\mathcal{T}$.

But, before all: this proof *reproduces* the proof of validity in the metalanguage, up to η-expansions, as a proof of completeness w.r.t. to Kripke semantics (w/o disjunction and existential) would do.

Alternatively, the previous proof expresses in direct-style that the completeness theorem holds for a model defined from its value on atoms by

$$
\begin{array}{lcl}
\Gamma \vDash_{\mathcal{M}} \tilde{\bot} & \triangleq & \bot \\
\Gamma \vDash_{\mathcal{M}} A \tilde{\rightarrow} B & \triangleq & \forall \Gamma' \supset \Gamma \ \Gamma' \vDash_{\mathcal{M}} A \rightarrow \Gamma' \vDash_{\mathcal{M}} B \\
\Gamma \vDash_{\mathcal{M}} \forall x \, A & \triangleq & \forall t \in \mathcal{M}_D \ \Gamma \vDash_{\mathcal{M}} A[t/x]
\end{array}
$$

and such that $\Gamma \vDash_{\mathcal{M}} \neg\neg A$ implies $\Gamma \vDash_{\mathcal{M}} A$. We recognise here that the proof with effects is a *direct-style* formulation of the completeness wrt Kripke semantics relativised to those models that satisfy $Class(\mathcal{M})$.